Ontology–based Learner Modeling System for Web-based Education

DANIJELA MILOSEVIC[1], RAMO SENDELJ[2], MIRJANA BRKOVIC[1]
[1]Technical faculty Cacak, University of Kragujevac
Svetog Save 65, 32000 Cacak,
SERBIA
[2]Faculty of information technologies,
Kralja Nikole 114, 81000 Podgorica
MONTENEGRO

Abstract: - The aim of this article is to present an ontology-based learner modeling system for deployment in personalized Web-based education, named LeMONT. In order to obtain student-centered learning we emphasize the need to understand that students come with different cognitive and affective characteristics, demanding personalized instruction. The proposed learner modeling system relies on a learner ontology, using Semantic Web technologies, based on the IMS LIP specifications and extended with new *Psychological* concept. The LeMONT system architecture with its components *Learner Profile Editor* and *Learner Modeler*, suited for implicit and explicit ontology-based user modeling, is presented. Implementation issues are also discussed and explained.

Key-Words: - Learner model, Ontologies, Web-based education, Learner modeling systems, Student-centered learning, Learner profiles, Personalized learning.

1 Introduction

In various application domains, user-adaptive software systems have already proved to be more effective and usable than non-adaptive systems. As pointed out in [1], adaptive systems with clear user benefits are user-adaptive (personalized) tutoring systems which significantly improve the overall learning progress.

Adaptivity of Web-based educational systems assumes collecting information about the student working with the system and creating the appropriate student model, which can then be further used to adapt the presentation of the learning material, navigation through it, and sequencing and annotation to the student. From the learners' perspective it supports pro-active learning that at the same time adapts itself to the potentials and needs of the individual learner.

Trends towards adaptive and intelligent Web-based educational systems (AIWBES) require the building of a model of the goals, preferences and knowledge of each individual student in order to adapt to their needs, and attempt to be more intelligent by incorporating activities like coaching students or diagnosing their misconceptions [2]. Depending on available information about the learner and AIWBES features, a certain degree of adaptivity can be achieved.

The necessity of ontological support for Web-based educational systems is emphased in [3], and states that the next-generation of Web-based educational applications should exhibit more theory and content-oriented intelligence and adaptivity and pay more attention to interoperability, reusability, and knowledge sharing issues. Also, according to [4], ontologies open doors to AI in Education (AIED) by introducing the possibility of reusable knowledge to intelligent tutoring systems. Ontological engineering plays a critical role in the advancement of knowledge-rich research like AIED which has diverse research fields related to it: Artificial intelligence, computer science, cognitive science, learning science, educational science, instructional science, etc.

In order to provide personalization to Web-based educational systems, we have created an ontology-based learner modeling system named LeMONT. Our system relies on a learner model ontology which provides the necessary armature around which student model knowledge base will be built. It contains common vocabulary for representing different learner information, placed on Web-server and accessible for various web-based applications.

The benefits such design are numerous, providing a standardized way of representing learner information and its usage in various educational

systems, as well as tracking information on long life learning.

2 Building the learner ontology

Building a learner model ontology gives the learner's information a common representation and semantic, providing a set of knowledge terms, vocabulary and semantic interconnections, together with simple rules of inference and logic. The ontologies are used to ensure that the teacher's description of the student's abilities and the machine-held index of skills and abilities use the same vocabulary. Such shareable and reusable learner model ontology is placed on a Web server and can be reused by a number of Web-based educational systems.

2.1 Design and implementation issues

In order to create comprehensive learner ontology and to include the majority of relevant concepts that will represent either explicit or implicit part of learner model we conducted a research on concepts that learner model ontology should consist of, taking into consideration the purpose of ontology.

There are many recommendations on which student characteristics should be kept in student models and a few attempts have been made in the recent past to model user cognitive and affective attributes in order to achieve a system's adaptivity according to the needs of individual user. And while researchers agree on the importance of adaptation towards user cognitive and affective characteristics, there is *"little agreement on which features can and should be used and how to use them"* [5].

Our intentions are primarily concerned with modeling learner's cognitive and affective characteristics in order to provide effective personalization e.g. the possibility of tailoring instructions to the individual student needs. Since cognitive and affective modeling is gaining in popularity we tried to include the most important concepts, so that different educational systems can rely on the same learner ontology, using more or less necessary concepts. The more in-depth approach on learner's cognitive and affective characteristics that we included in our ontology can be found in [6], [7] and [8]. In [8] we presented guidelines for designing lesson content tailored to individual users, taking into consideration specific *learning style* (Kolb learning style), as a cognitive learner's characteristics and subject matter *learning motivation*, as an affective learner's characteristics.

Learner model is structured according to the IMS Learner Information Package (IMS LIP) specification [9]. The IMS LIP specification defines a common format for describing learner information, enabling exchange and interoperability among systems. IMS LIP is structured in eleven groupings and includes: *Identification, Goal, QCL (Qualifications, Certifications and Licenses), Accessibility, Activity, Competence, Interest, Affiliation, Security Key and Relationship*. Each grouping contains relevant concepts for describing learner characteristics. In order to include additional information about learner's cognitive and affective characteristics we extended IMS LIP specification with a new concept named *Psychological preferences*, Fig.1.

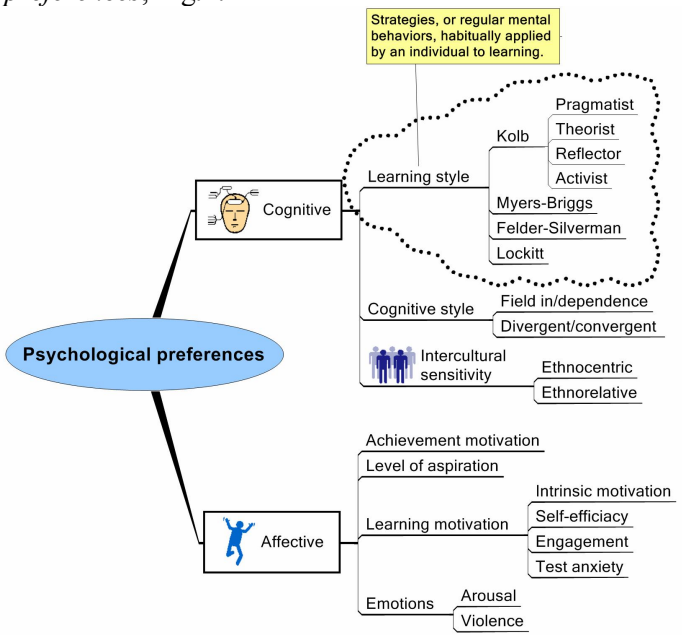

Fig.1 *Psychological preferences* concept

Proposed learner model ontology is implemented with Protégé-Ver. 3.0 ontology development environment [10]. The Protégé tool enables creating, editing, browsing and codifying ontologies. Since ontology is modeled on conceptual level it is independent on final representation language. In order to create sharable and reusable learner model ontology we converted it into RDF/RDF Schema format.

The Protégé plug-in named OntoViz is intended for ontology visualization. Fig. 2 shows the *Activity* concept with its properties for capturing learner's activities and achievement during teaching process. The presented activity has two instances *Course* and *Lecture*, where *Lecture* is a sub-activity of *Course*.

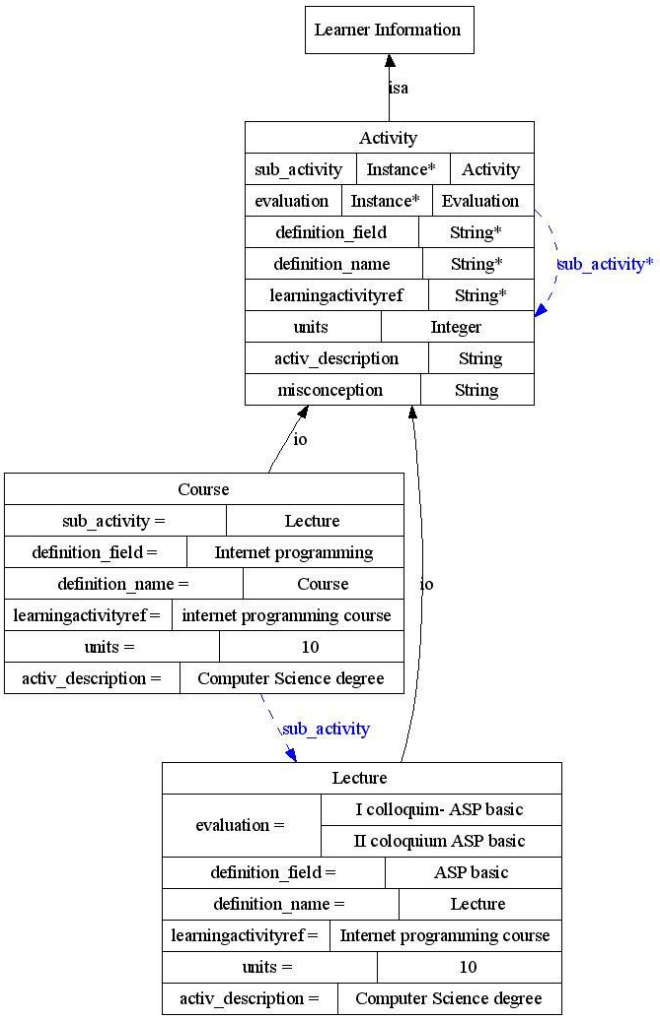

Fig. 2 *Activity* concept with instances by OntoViz

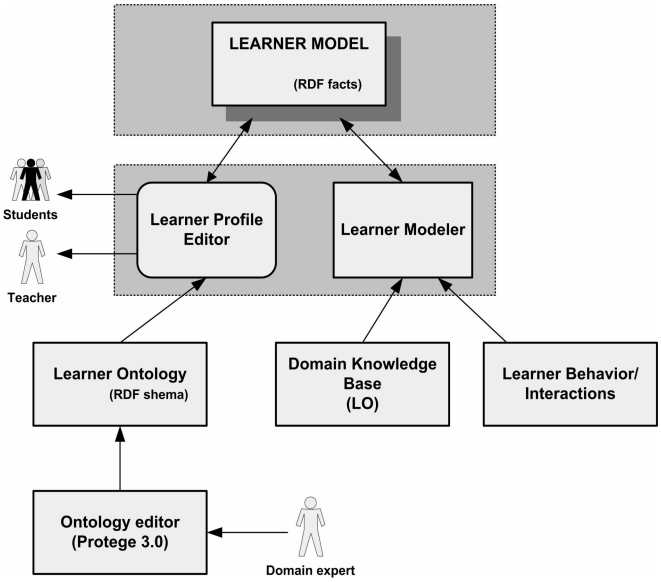

Fig. 3 The LeMONT system architecture

3 The LeMONT architecture

The LeMONT system is an ontology-based learner modeling system. Its architecture comprises several components, as shown on Fig. 3. The ontology editor Protégé enables domain expert to create and modify learner ontology according to the required concepts, properties and their relationships. The core of the LeMONT architecture is learner ontology, a vocabulary for representing the learner model structured according to IMS LIP specification in order to provide reusability among various applications.

The LeMONT system stores ontology as an RDF Schema and the instances of learner ontology concepts as RDF facts, generating Learner model. The learner model data are obtained on two ways: through *Learner Profile Editor* (LPE) and *Learner Modeler* modules.

The *Learner Profile Editor* is ontology-based editor for acquiring explicit learner data according to proposed ontology. LPE has a role of instantiating learner ontology. It enables learners or teachers to edit, browse, update and visualize the specific learner profile.

The *Learner modeler* module is in charge of providing implicit learner model data. It analyzes learner's tracking data stored in *Learner Behavior/Interactions* part of specific Web-based educational system and makes the necessary changes in the *Learner model,* concerning student's knowledge and preferences.

The *Learner modeler* can be realized differently according to the needs of specific educational system. We envisage it will be used for modeling learner knowledge level in relation to a specific concept, using overlay learner model and modeling knowledge status based on fuzzy set theories. Also, meta-data of specific Learning object (LO), like minimal and maximal learning time and semantic density, are used for computation of knowledge status and for tailoring teaching process to individual student needs.

4 The LeMONT implementation

The system has been implemented as a web application using RDF API for PHP (RAP) [11], PHP technology, MySQL server and Apache web server. LeMONT server is a learner modeling ontology-based system based on IMS LIP specification. RDF API for PHP is a Semantic Web toolkit, recommended by W3C as developer tool

[12] for PHP developers with built-in programming interfaces for manipulating ontologies using RDF-Schema specific methods. It includes a web server, in-memory or database model storage and provides extra capabilities for handling ontology data sources such as inference engine supporting RDF-Schema reasoning.

PHP technology is used for front-end application – Learner Profile Editor (LPE). LeMONT stores learner profile data in RDF/RDFS format. MySQL is used for persistent database model storage.

Users are accessing the LeMONT system through *Learner Profile Editor* - LPE interface for explicitly updating learner model. At run-time, the system works with several models: *learner ontology model* (stored as .RDFS file) used for generating editor interface, *database model* for loading existing learner model data and *memory model* of the retrieved learner data containing changes during editing profile.

On the other hand, learner model is implicitly updated by *Learner Modeler*. *Learner modeler* is interacting directly with learner model stored in database.

Specific learner information is calculated and updated based on the learner's tracking data stored in learner behavior/interactions module and meta-data of learning objects in domain knowledge base. These changes, made implicitly by the system, are also stored in the same database and viewable and explicitly editable through the LPE. When users enter the system through a login process they are accessing learner profile from learner profiles database store and new users can be registered by providing username, password and *learner id* property that will be associated with the profile.

The learner can view or edit its own profile or if the user is a teacher, he can list existing profiles or create a new one through the interface shown in Fig.4.

Learner model ontology is used for generating LPE Web interface that enables viewing, adding, deleting and replacing learner model properties in a property hierarchy, as well as, adding entire existing models exported from other IMS LIP conformant educational systems. The LPE interface is generated *automatically* from underlying learner model ontology stored in RDFS format.

The core data structures containing the learner information are placed on the left hand side of the LPE: *Accessibility*, *Activity*, *Affiliation*, *Competency*, *Goal*, *Identification*, *Interest*, *Psychological*, *QCL* and *Security key*.

These learner information elements are based on IMS LIP, extended with the *Psychological preferences* concept. RAP methods are used for extracting in-depth concept data structures.

When a user selects a concept, form fields are generated from schema model that provide ontology specific methods. Using those methods, information about the property is retrieved: its cardinality, subclasses and sub properties. If a property has multiple maximal cardinality defined in the ontology model, LPE allows the user to add more than one value/instances to the property, by pressing the *Add* button. The user can also edit properties by pressing the *Edit* button.

When the user updates learner model, the model is saved in database and also in learner model RDF file. The RDF file containing models of all learners is saved after each learner model update.

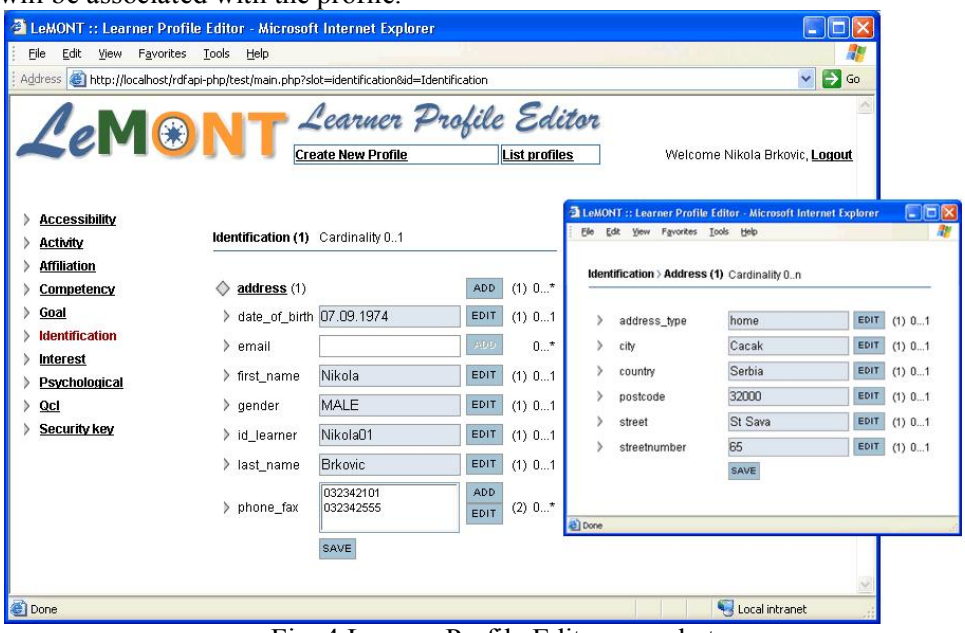

Fig. 4 Learner Profile Editor snapshot

4 Conclusion

Development and implementation issues of ontology-based learner modeling system, named LeMONT, are presented. LeMONT has a generic architecture for modeling learners in personalized Web-based educational systems, based on ontologies. The underlying learner ontology is structured according to IMS LIP specification extended with *Psychological preferences* concept, capturing the most relevant learner's characteristics for tailoring teaching process to the individual student needs. The LeMONT system has been implemented as a learner modeling server using RDF API for PHP (RAP) developer tool for RDF.

Our future work will go towards developing two applications that will use the proposed learner ontology: Skills management system and Adaptive educational system tailored to student's learning style and subject matter learning motivation. An evaluation study on using ontologies in those two systems will be provided.

References:

[1] Fink, J., Kobsa, A., A Review and Analysis of Commercial User Modeling Servers for Personalization on the World Wide Web. User Modeling and User-Adapted Interaction, UMUAI, Kluwer Academic Publichers, Vol. 10, No. 2-3, 2000, pp. 209-249.

[2] Brusilovsky, P., Peylo, C., Adaptive and Intelligent Web-based Educational Systems. International Journal of Artificial Intelligence in Education, IOS Press, Vol. 13, 2003, pp. 156-169.

[3] Devedžić, V., Key Issues in Next-Generation Web-Based Education. IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews, Vol. 33, No. 3, 2003, pp. 339-349.

[4] Mizoguchi, R., The Role of Ontological Engineering for AIED Research. ComSIS, Computer Science and Information Systems, Volume 2, Number 1, 2005.

[5] Brusilovsky, P., Peylo, C., Adaptive Hypermedia. User Modeling and User-Adapted Interaction, Kluwer Academic Publishers, 11, 2001, pp. 87-110.

[6] Milošević, D., A Cognitive Approach to Learner Modeling for Web-based Education. Proceedings of The 28th German Conference on Artificial Intelligence (KI-2005), Workshop "Knowledge Engineering and Software Engineering (KESE)", Koblenz, Germany, 2005, September 11-14.

[7] Milošević, D., Brković, M., Learner model ontology in adaptive web based educational systems. Proceedings of The 8th International Conference on Interactive Computer aided Learning - ICL2005, CD edition (ISBN 3-89958-136-9), Kassel University Press, Villach/Austria, 2005, September 28 – 30.

[8] Milošević, D., Brković, M., Designing lesson content in adaptive learning environments. Proceedings of he international conference on Interactive Mobile and Computer Aided Leaning, IMCL 2006, Amman, Jordan, April, 2006.

[9] IMS Learner Information Package, Information Model Specification, Final specification, version 1.0. [Online] Available: http://www.imsglobal.org/specifications.html

[10] Protégé-Ver. 3.0 ontology development environment [Online] Available: http://protege.stanford.edu/

[11] RDF API for PHP V0.9.3, [Online] Available: http://sourceforge.net/projects/rdfapi-php/

[12] Developer Tools for RDF, Developer Resources for PHP, [Online] Available: http://www.w3.org/rdf/.