Kernel for a Semantic Learning Platform with adapted suggestions

Ioan SZILAGYI, Radu BALOG-CRIŞAN, Ioan ROXIN, Ana ROXIN
LASELDI, University of Franche Comté
Multimedia Department
Montbéliard, France
{ioan.szilagyi; radu.balog-crisan; ioan.roxin}@univ-fcomte.fr

Abstract — **In the context of personalized learning, an important aspect is providing adapted suggestions to the learner. These suggestions consist in resources that are either specially designed for learning purposes or not, notably in the case of blog articles, forum discussions, etc. Anyway, the goal is to help the learner in better understanding a concept, with any kind of additional resources. The offered suggestions are selected in close relation with the learner's profile. The Learner Profile consists in the sum of specific learner characteristics that are used to describe the Learning Style, Learning Path, Learning goals, Knowledge Base, etc. The decisions about suggested resources are taken at the Kernel level, which represents the central part of the Semantic Learning Content Management System.**

Keywords: SLCMS; semantic learning, adapted suggestions; ontologies; RDF; OWL; LOM

I. INTRODUCTION

Personalized learning represents an active research domain, being discussed even in government's reports [1]. In most web services and mainly in e-commerce services there are already systems that offer personalized automatic recommendations. However, today most e-learning systems are still delivering the same educational content to learners with different needs. There can be different approaches of adaptation. A taxonomy of adaptive hypermedia technologies is presented in [2] which categorizes them in two main approaches: Adaptive Presentation and Adaptive Navigation Support.

In our presentation, we implement Adaptive Navigation Support. The learning objects and learning resources are not created at running time, but only presented to the learner in concordance with his/her needs at given time. Adapted recommendation implies that the user profile is created and maintained dynamically in order to improve the suggestions.

Along with the construction of the learner profile, we think that the competency-based approach suits well for the learning suggestion in order to associate learning resources, contents and suggestions for the learner. In this way, the student obtains fast and reliable learning performances for specific tasks.

II. LEARNER PROFILE

In order to be able to personalize the learning process, we need to build the learner profile. A learner profile is a collection of information related to an individual learner. Its purpose is to provide a view of the current development and future potential in terms related to self-access learning [3]. The learner profile is the student's representation in the learning system.

The Learner Profile is constructed from two different information perspectives, regarding the learner. First, we take in consideration the learning style of the learner, which contains psychological characteristics about the learner and preferences regarding the learning approach. The second source of information is related to the learning objectives, competences and knowledge that the student has or wants to achieve.

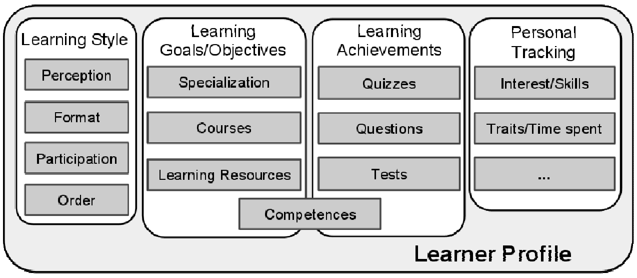

Figure 1. Learner Profile.

The profile has to be dynamic, enriched and improved through every interaction the student makes with the learning platform. The learning profile must correspond as much as possible to the real pedagogical characteristics of the student, notably his level of knowledge and acquired competences.

A. *Learning Style of the student*

Several approaches exist for the implementation of Learning Styles: The Myers-Briggs Type Indicator, Kolb's Experiential Learning Model, The Felder-Silverman Model [4], Dunn and Dunn learning style model [5], Honey and Mumford model based on [6]. In our approach, we implement the method presented by Felder-Silverman [7]. According to their model, the student's learning style can be defined by the answers given to four questions related to: the student's perception (sensory/intuitive), the format of content (visual/auditory), the student's participation (active/reflective) and the order chosen to present the content (sequential/global).

At the start point, when the student starts the process of registering to the system, the system asks him/her to answer four questions, in order to establish the learning style that matches him/her best. The learning style of each learner is characterized by four values, one for each dimension (question?) on a large scale. These scales facilitate and allow describing the learning style preferences in more detail.

During the learning process, the system repeats some of these tests in order to update learner preferences and to stay as close as possible to the real learning style. The tests can consist in small quizzes with multiple choice questions, short questions, visual matches or simple games.

The defined learning style will be checked against the learner's feedback obtained from the different actions that he/she makes in the learning system: types of resources that he chose to study (visual/auditory, interactivity level), good response rates to different types of questions, etc.

We are interested in knowing what kind of suggestions the student chooses first or the amount of time that the learner spends with different types of resources (interactivity level, text resources, visual resources etc.). This way, we can measure the interest of students for different types of learning resources (and the effectiveness of different resources).

B. *Learning objectives, user knowledge, competences*

The second source of information needed to construct the learner profile consists in data related to his/her learning objectives and progress, learning goals, competences that he wants to achieve or he already masters, the knowledge base and learning background of the student. We characterize the student related to his learning objectives, knowledge and competences. The platform can suggest learning resources according to these characteristics, learner's learning objectives and knowledge level.

When the student is registered within the system, he/she completes the education profile that he/she has at that time: acquired diplomas, implicitly gained competences and knowledge, learning objectives, known languages etc. The more the learner interacts with the system and completes quizzes and questions, the more information is acquired about his knowledge level, resulting in an enriched personal learning profile.

When the student selects the specialization he/she wants to pursue, the related courses are selected in order to be covered by the student. Every course has a list of learning objects and different learning resources (articles, quizzes, etc.), in a certain order that should be followed. The Specialization, Learning Path and Course concept are defined in LMD Ontology [8] using Web Ontology Language (OWL). However the data are expressed and processed at Resource Description Framework (RDF) level in the Kernel and application.

In the LMD Ontology, the Learning Path concept represents a list of specific learning objects, learning resources, quizzes and tests that every student should follow according to the chosen course. The Learning Path from the LMD Ontology is generated according to specializations and courses selected by the student. That is not a personalized Learning Path. The proposed learning resources and their order are defined by the responsible professor according to the selected specialization, course, etc.

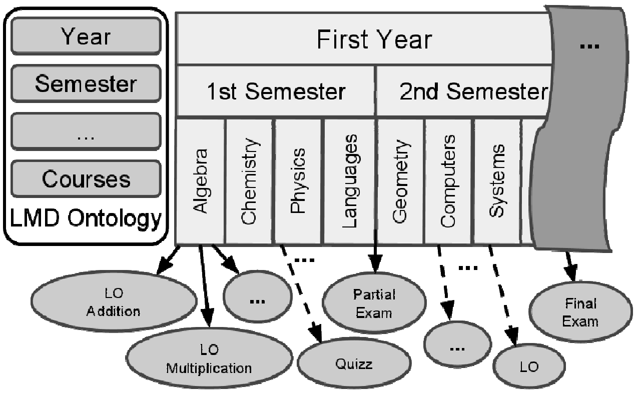

Figure 2. Learning Path in the context of LMD Ontology.

In the context of a Learner Profile, we define a Learning Path which is personalized according to every student's needs and personal learning profile. The personalized Learner Path defined in Learner Profile Ontology does not overwrite the Learning Path defined in LMD Ontology. The student is not forced to cover all the learning resources existing in the Personalized Learning Path, but he/she has to go through all resources listed in the Learning Path defined in LMD Ontology.

The Learning Path from the Learner's Profile can be personalized either automatically or by professors. When this is done automatically, the system can remove or add learning objects, quizzes and different learning resources from the student's learning path after the student has interacted with the platform and his learning profile was changed, for example when the student failed to pass an exam or a quiz.

Starting from the learning path that is defined in the LMD Ontology, the professor can personalize and enrich the leaning path of a student by adding the suggested learning resources to the personal Learning Path of the student.

The personalized learning path can be dynamically changed and recomposed as the result of learner's actions and feedback within the platform (Fig. 3).

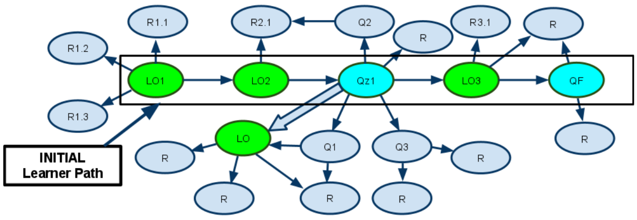

Figure 3. Learning Path - 2.

The paths made by students in the system and the quizzes' responses represent the main source of information regarding the automated change of the learner path.

The system records inside the learner profile the educational profiles of the student's most chosen learning resources, the time spent with different types of resources, the effectiveness of resources with the different educational profiles.

In addition, apart from the learner profiles and LOM's metadata, we consider implementing a rating system for all learning resources used in the platform. The students and professors have to be able to classify and appreciate the effectiveness and the pedagogical value of a learning resource. A resource that had a good impact to a learner with a specific learning profile will be suggested to other users with the same profile in order to obtain fast and better achieving of their learning objectives.

In order to better relate the learning objectives, specializations, learning resources, student's learning goals and other learning concepts within the platform, we consider a competency approach.

C. *Competency approach*

Nowadays competence-based education is gaining more interest from the learning organization or even from the governmental learning system because it provides a clear and precise purpose in the learning process [9]. The competence approach gives us a better and consistent correlation between the learning resources. Also the Competence concept helps making easily the transition to professional environments (human resource managements), where this concept is mostly used. .

In our approach, we build a Competence Ontology. The classes defined in this ontology are in close resemblance with the data elements described in Reusable Competency Definitions [10]. In addition, we add new concepts that help us linking the learning resources to other resources. There is also a hierarchy of competences, with the idea that the access of a competence can be restricted for a learner that doesn't have a prerequisites list of competences. Taking into account that acquiring learning objectives is in close relation with achieving competences, we can build links between learning resources and benefit from those links in order to provide the well-adapted suggestions.

III. Adapted suggestions

A suggestion is an idea or a plan put forward for consideration, a slight trace or indication of something. In this case, the suggestions will consist in pedagogical and learning resources proposed to the learner for taking in consideration in their exploring for better knowledge and ways that makes the learning concepts more easily to be understood.

The suggestion is provided in order to improve the learning process, learning time and learning quality of the student. In our approach, each suggestion is actually a learning resource that explains a concept or helps the student to better understand notions, concepts and learning objectives. The aim is to make those suggestions or indications, more sensitive to user needs at a specific time on the completion of his/her learning path. The learner's needs are deducted from his/her profile.

These suggestions will be more efficiently chosen for the student if we propose learning resources that are in concordance with his/her learning profile. For example, if the student responds well at a highly interactive content, than the system will order accordingly the proposed suggestions and the learning resources (highly interactive content is provided first).

The system computes a list of suggestions that match learner's needs and will order that list according to predefined rules. One rule can be for example: When two suggested learning resources have the same learning objectives and one has more interactive content (which match my learning preferences) than the other, but it is lower rated by students (that approximately have the same learning style), than the best rated resources is suggested first.

The suggestions are continuously improved and at some point, the suggestions can be included in the learner's path. As an example, we may cite the case when the student doesn't succeed to a test and one can conclude that he/she didn't understand a concept. So the system decides that the learner needs to redo some sections from the learning course and review some specific learning objects.

IV. Semantic Kernel

The Semantic Kernel is the central part of the SLCMS [11]. The architecture of the SLCMS is presented in Figure 4.

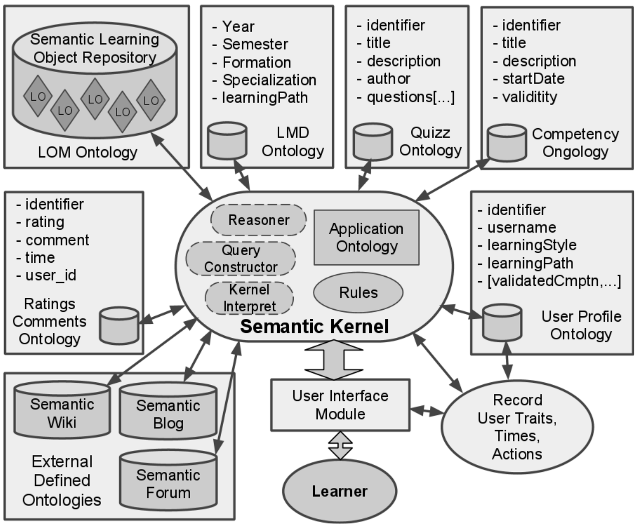

Figure 4. SLCMS Arhitecture.

In SLCMS all data that interacts must be represented using semantic technologies. Behind every involved module there is a defined ontology which is an abstract model that describes data from that module. These are the Content Ontologies. In this paper, we do not present the details regarding the defined content ontologies, as some of those ontologies have been presented or anticipated in other paper works: LMD Ontology and Quiz Ontology in [12], LOM Ontology in [RDF4LOM] which is in close relation with descriptions from [13]. Other content ontologies are mentioned here with the intention to give a better overview of the entire system.

Two approaches exist for kernel data acquisition. One approach is when the kernel asks and makes queries in each module connected to the platform; in this case, the kernel builds different queries for the different terminologies defined in the Content Ontologies. The second approach consists in the construction of two more ontologies at the kernel level that are linked to the content ontologies, in order to allow building queries at kernel level and retrieving data from all other modules. We take in consideration the second approach because it gives us more flexibility and control on the data representation. We define two other ontologies that are: the Application Ontology and the Bridge Ontology.

The Application Ontology is an abstract model of the data that we want to handle at application level. For example, when we search for learning resources, we want the application to know that the learning objects, articles from wikis, images, videos or other documents are learning resources. We want to retrieve data from all modules at the kernel level, without having to build specified queries for each module.

To connect the Content Ontologies to the Application Ontology, we create the Bridge Ontology. The Bridge Ontology allows us making queries to the Application Ontology and get data from all defined Content Ontologies. The Bridge Ontology will merge and map the Content Ontologies to Application Ontology. In that way we can also integrate and use more easily external defined ontologies.

[…creating ontologies(reuse, merge, extend)]

Inside the Application Ontology, we define the LearningResource class that represents the concept of learning resources at a higher level. A Learning Object or an Article are specific concepts in the context of learning resources and are declared in separated ontologies. At Application Ontology level we declare:

```
LearnRes rdf:type owl:Class . (1)
```

Inside the Bridge Ontology, we specify that the classes LearningObject and Article are subclasses of the LearningResource class:

```
LearnObj rdfs:subClassOf LearnRes . (2)
 Article rdfs:subClassOf LearnRes . (3)
```

When applying the First Order Logic writing to statements (2) and (3) we obtain the following:

$$\forall o\,[LearningObject(o) \nabla Article(o) \rightarrow LearningResource(o)] \quad (4)$$

At the Bridge Ontology level, we define equivalences between the concepts represented inside the Application Ontology and concepts represented inside the Concept Ontologies. The OWL properties used for defining these equivalences are:

- owl:equivalentClass
- owl:equivalentProperty
- owl:sameAs
- rdfs:subClassOf
- rdfs:subPropertyOf

After an equivalence is defined, the Reasoner will inference and will know that Learning Objects and Articles are also LearningResources, and a request that asks for Learning Resources will return Learning Objects and Articles also.

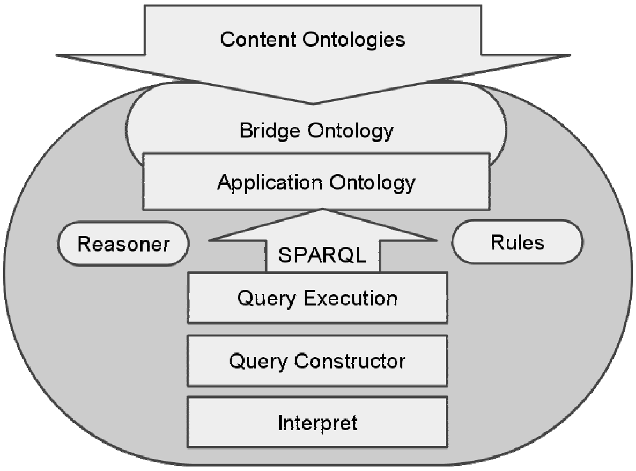

Figure 5. Arhitecture and Technologies used at Kernel level.

After the ontologies are linked and the connection to the data graph is established, we load the Reasoner in order to infer about the data, so we can get milliseconds time for asked queries. The Reasoner will infer and generate new relations between the resources based on the linkage that we made between the Application and Bridge Ontology. There are two ways in which we can run the Reasoner. First approach is to make reasoning at the query time (*backward chaining*); this way the Reasoner will do the minimum reasoning in order to satisfy the query. The second approach is to load the Reasoner engine and to infer everything that is possible (*forward chaining*). The Reasoner will produce new inferred triples deducted from the existing data and generate new relations between the objects defined in the RDF-database.

Several different Reasoner engines can be used (Pellet, Bossam, Hoolet) and each of them supports different rule languages and has different capabilities. For this project, we use the OWLMicroReasoner because it comes with Jena Framework and provides minimum reasoning requirements.

The java code for loading a reasoner is:

```
Reasoner             myReasoner             =
ReasonerRegistry.getOWLMicroReasoner();
model = ModelFactory.createInfModel(reasoner,
model);
```

Some examples of how the OWLMicroReasoner functions are given in Table I.

TABLE I. SOME INFERENCE RESULTS

IF	THEN
LearnObj rdfs:subClassOf LearnRes . myObj rdf:type LearnObj .	myObj rdf:type LearnRes .
Images rdfs:subClassOf FileType . FileType rdfs:subClassOf ContentFormat .	Images rdfs:subClassOf ContentFormat .
Student hasPersonalizedLearnPath LearningPath .	Student hasLearnPath LearningPath .

hasPersonalizedLearnPath rdfs:subPropertyOf hasLearnPath .	
myObj hasInteractivityLevel “very high” . hasInteractivityLevel rdfs:domain LearnRes .	myObj rdf:type LearnRes .

In order to make queries over the RDF-database, we use SPARQL. The queries must return suggestions consisting in identifiers for learning resources.

The Query Constructor will construct the queries and sends them to the Query Execution in order to be executed against the RDF-database.

At Query Constructor level, several queries are saved and can be mixed and combined dynamically in order to obtain more specific responses. The queries are created with respect to the terminology defined in Application Ontology.

The query for retrieving learning resources related to my learning style and has “myCompetence” as associated competence looks like:

```
SELECT ?learnRes
WHERE {
?learnRes appOnt:type appOnt:myLrnStyle .
?me appOnt:hasLrnStyle appOnt:?myLrnStyle
.
?learnRes appOnt:assocCompetence
“myCompetence”
…
}
```

The syntax of our SPARQL queries is simple because we ask for suggestions consisting in learning resources, while the data is enriched all the time, with every interaction of the learner with the system.

It is possible to implement rules with different inference engines. When we specify rules, we obtain more accurate and consistent data.

The learner should be able to ask complex and specific queries. An example of a request that the kernel can also answer is: *Give me all the learning resources that match my learning style in order to acquire the competence “Multiplying Numbers”.*

The kernel has to identify the concepts used in the request:

- *me* – the student concept, which has a representation on the LMD Ontology, and the object that represent me
- *learning resources* – all the instances of which the types are subclasses of LearningResource Class.
- *match* – a property which defines the limits within which two objects can be considered that resembling
- *myLearningStyle* – a property defined by the four dimension of the learning style preferences.
- *acquire* – a property defined for the competences
- *competence* – the Competence concept defined in the Competence Ontology
- *multiplying numbers* – an instance of a Competence class, a defined competence

Other possible questions could be:

Give me all learning resources:

- that have a lot of interactivity
- that are images
- that I need in order to pass at Partial Quiz
- that are rated with four or five stars with the students that have the same learning style like mine
- that I need in order to achieve that competency
- made by professor Y

The kernel is capable of answering different SPARQL queries from the data that it has in the system and make decisions for providing the best answer possible.

Another feature of the platform is that the user can explore the content by itself but with recommendations for every resource based on the relation between the browsed resources.

The Semantic Kernel must provide also other essential functions for the platform. It has to be able to interpret data from the connected modules and dynamically construct queries.

The Kernel Interpret is the part which handles the data exchange between the kernel and User Interface Module, it interpret the requested queries and format it in order to send it to the Query Construction part.

V. Conclusion and Further Work

The adapted suggestion is just a step from the personalized learning idea.

This part of the Semantic Learning Content Management System helps students gain time and provide consistent resources adapted to their needs. The application is still in developing phase. The used ontologies must be enriched and refined in order to express data at a high granularity. The Kernel Interpret and User Interface Module needs to be developed in order to understand complex queries from the learner. The Query Constructor has to able to construct more complex and intelligent queries.

The interface of the platform has to be easy to use, in natural language and powerful in order to provide access at complex queries.

We consider using a more complex reasoning engine and implement rules.

References

[1] Teaching and Learning in 2020 Review: Call for Evidence, July 2006, http://publications.teachernet.gov.uk/default.aspx?PageFunction=productdetails&PageMode=publications&ProductId=NEWS-5556-2006

[2] Brusilovsky, P., “Adaptive Hypermedia”, User Modeling and User-Adapted Interaction 11, 2001, pp87-110

[3] Gardener D. and Miller L. “Establishing Self-Access: From Theory to Practice” Cambridge Language Teaching Library, Cambridge University Press (March, 1999), pp 84.

[4] Felder, M, R, and Brent, R, “Understanding Student Differences”, Journal of Engineering Education, 94, 1, 2005, pp 57-72.

[5] Dunn, R. and Dunn, K. Teaching students through their individual learning styles: A practical approach. Reston, VA: Reston Publishing, 1978

[6] Honey, P. and Mumford A. The Manual of Learning Styles, Peter Honey, Maidenhead, 1992.

[7] Felder, R. M., and Silverman, L. K. (1988). Learning and Teaching Styles in Engineering Education. Engineering Education

[8] Balog-Crişan, R., Roxin, I., Szilagyi, I. “Ontologies for a Semantic Quiz Architecture”…

[9] Sampson, D.G. “Competence-related Metadata for Educational Resources that Support Lifelong Competence Development Programmes”. Educational Technology & Society, 12(4), pp.149-159.

[10] IEEE RCD (2004). *IEEE P1484.20.1/Draft Standard for Learning Technology – Data Model for Reusable Competency Definitions*, Retrived January 20, 2010, from *http://.......*

[11] Balog-Crişan, R., Roxin I., “Semantic Learning Content Management System”, Proceedings of e-Learning vol.II, IADIS Multi Conference on Computer Science and Information Systems, Amsterdam, The Netherlands, 2008, pp.85-89.

[12] Balog-Crişan, R., Roxin I., Szilagyi I., “Ontologies for a Semantic Quizz Architecture”, IEEE ….2009, pp.

[13] IEEE P1484.12.4/D1 Draft Recommended Practice for Expressing IEEE Learning Object Metadata Instances Using the Dublin Core Abstract Model