Ontologies for a Semantic Quiz Architecture

Radu Balog-Crisan, Ioan Roxin, Ioan Szilagyi
LASELDI, University of Franche-Comte, Montbéliard, France, Multimedia Department
{firstname.lastname}@univ-fcomte.fr

Abstract

In this paper we propose an architecture for a Semantic Quiz. In the context of a Semantic LCMS we introduce two ontologies: the Quiz ontology and the LM ontology. The Quiz ontology is associated with the LM ontology that is related with the field of higher education in Europe. We present an application that uses these two ontologies. With the Quiz ontology we build semantics metadata that express the question; thus each question to be identifiable and reusable. This solution may be a supplementary one to the classical pedagogical methods, in order to constitute the student profile and learning path, knowing that each question can contribute to the validation of the student competences.

Keywords: Semantic LCMS, RDF, OWL, Quiz

1. Introduction

In the context of the Semantic Learning Content Management System (SLCMS) platform introduced in [1], we have presented the different modules associated to the SLCMS. Considered from a broader angle, the goal of this research project is to develop a prototype of a semantic knowledge model for learning platforms. The SLCMS includes the Semantic Course module, the Semantic Wiki, Semantic Blog, Semantic Forum and the Semantic Lightweight Directory Access Protocol (LDAP) and other modules. The Semantic Course module incorporates the Semantic Quiz tool.

We agree with the idea that course management systems need to be supplemented by external resources, available on the web [10]. This has to be done in the context of a semantic data web [3] [4] [7] [8]. Indeed, there are cases when, the system doesn’t support innovative instruction sequences, such as creation and modeling: therefore course designers must use external tools to implement the task. There are also cases when several resources are not accessible directly into course management systems. In the context of our domain ontology we use competences, which are assigned to the questions; these are playing a central role in finding similar resources on the web or in the limit of the SLCMS. To achieve this, into the SLCMS, our approach was to use the Semantic Kernel module. The role of this component is to provide more specific resources, at the use of learner, that are related to his course domain. For that we use the semantic data descriptions of the related modules [1]. We see this as a way of enriching the learner’s support and knowledge database with new data, in a benefit of a better understanding and faster learning.

In this paper we focus on presenting the architecture of the Semantic Quiz, which will express with ontologies the questions, question variants, competences, student answers and courses with Learning Object Metadata (LOM) which is linked with the university LM ontology.

2. The domain ontologies

In our approach we use two main ontologies:

- The Quiz ontology defines the question concept, along with its configuration and variants. The competence concept can be validated, by the learner, depending on his answer to the question, which is modeled in ontology;
- The LM (Licence –Bachelor–, Master) ontology is a complementary ontology related to the field of higher education in Europe (“The Bologna Declaration on the European space for higher education”, Bologna, 1999). This ontology is enclosed in the LMD model (LMD is the name given in France to the harmonization of academic degrees throughout the European Union). We consider this ontology complementary because it simply defines a field of application for our Quiz ontology.

The Semantic Quiz application processes these domain ontologies at Resource Description Framework (RDF) level. Each question has semantic metadata related to it, which gives independence, in a sense that it can be reusable and enables the teacher to easily manipulate and elaborate quizzes, knowing that each question has a relation, to course domain and to competences.

The semantic layer allows the teacher or the student to gather more accurate information on questions and quizzes. This can be helpful in creating the student’s path through the learning objects, knowing that each question validates, in part, the student’s skills and competences.
2.1. The Quiz ontology

Generally speaking a quiz is defined, as a set of questions designed in order to evaluate knowledge, in to a given domain [9]. In most cases, questions are elaborated by the teacher, in the perspective of verifying one or more competences of the learner. The competences can be: a. General competences (notably the ability to measure evidence, the application of known principles to new situations, the power of deduction, etc.); b. Specific competences (notably encapsulation, modularity, polymorphism, inheritance, etc.).

The Quiz ontology is characterized by four main classes defining the concepts of: Question, Quiz, Configuration and Competence. In the structure of the ontology (Figure 1) we can note two central concepts: Quiz and Question.

The Question competence is associated to the LOM RDF binding [2], which is also linked to our Course class, by the object property hasLOM. Every Question has at least one Competence; if a group of questions that verify the same competence are answered correctly, this Competence will be assigned to the student and normally he masters this competence.

The hasQuestion property allows to be associated several questions for a given Quiz. The hasType property allows defining the following types of question True/False, Drag and Drop, Multiple Choice, Numerical, etc., to be set up.

Each Question has a Configuration that includes the question text but also the linked multimedia resources. The Configuration and the Variant class are linked by the hasVariant property. The Variant concept defines variant text, media resources and a boolean property, isValid, that is used to validate the actually variant of the question.

The Student, Course and Author classes are in relation to the Quiz class. Through data property hasQuiz the Course class is in relation with LM ontology.

The teacher can associate to a Student quizzes, and those become pending quizzes. As the student answers the system proposes for the validation to teacher the associated competences for the student.

With the LM ontology (Figure 1) we emphasize, in a practical way the Quiz ontology.

The link with Quiz ontology is made through the Course class and Quiz class, but also by linking the learning object to the Course class. We will not present here this ontology, because is beyond the subject.

3. Semantic Quiz application

Now that we have introduced the techniques and ontologies used for our application, we can take a closer look at the proposed application.

The main goals of the application are:

- Building specific RDF metadata around each concept used by the application;
- Managing interaction between users, questions, quizzes and other concepts defined in ontologies.

In the following section we will discuss the architecture for this application.

3.1. General architecture and technologies used

We used a typical three-tier architecture (presentation, logic and data tier), as in Figure 2.

![Figure 2: Application Architecture](image)

At presentation tier, we have chosen Flex because it allows the creation of rich interfaces for users and therefore taking advantage of the Rich Internet Application (RIA) associated technologies and features. Moreover, Flex uses Action Script 3 (AS3) as a programming language, and this enables the creation of
client logics and an easier mapping between Flex and Java classes.

At server side we use Tomcat. For creation and manipulation of RDF triplets we use Jena, which is a Java API and it have object classes to represent graphs, resources, properties and literals [5]. At the top of the Tomcat server, we have installed LifeCycle Data Services from Adobe, so we can use Flex Framework for presentation. Communications with clients is done through the Real Time Message Protocol (RTMP), which allows sending messages from the server to client without the need for the client to call the server.

The JManager class manages Java events and logic. In a similar manner, the FManager class manages Flex events and logic. The JManager and the FManager modules also ensure the communication, using RTMP and coordination between Flex and Java entities.

The functioning of the whole application is based on class events. The Data tier manages resources depending on their type. MySQL server is used to store RDF triplets.

3.2. Application Functionalities

The client has access to the application through a Web browser with Flash Player enabled. RTMP communication protocol allows displaying the interface without having the browser reload the entire page.

At the Logic tier, located on the Tomcat server we find the core of the application: the RDF graph engine, which constructs RDF graphs from metadata received from users (students, teachers, and administrators) and our previously defined ontology. The RDF graph engine is encapsulated in the RDF Construction module. Generated graphs are stored in the application’s database.

For every concept defined in the above presented ontologies, we have created a structure Java class and an alias structure AS3 class, which is bound to the corresponding Java class. Data received from the client are attributed to an instance of AS3 corresponding class and then sent to Java side. In Java these data are sent to the RDF Construction module, which assigns the data received from the instantiated class to the concepts, and then constructs the associated RDF graph. In form of triplets, the concepts are saved in the database with all of the corresponding properties.

In order to dynamically retrieve resources that are stored into database, the interrogation queries are used, relative to our given application domain. On the Java side RDF SPARQL module creates these queries, depending on what kind of information is required.

Data is divided into two main categories and then saved to the Data-tier depending on their type:

- The RDF triplets for each concept used in our ontology models, the metadata about concepts (Question, Quiz, Users, etc.) used in the application, are saved along with the RDF types models on MySQL server.
- Multimedia assets (jpeg, mp3, mpeg), which are stored in a specially created folder.

In the future we will consider saving the metadata information, MPEG-7 and MPEG-21, along with the files, as the standards evolve and integrate the RDF syntax [6].

4. Conclusion

We have proposed an architecture for Semantic Quiz, that include two ontologies: Quiz and LM. The Quiz ontology builds the metadata around the question, using OWL. Therefore we have explicit formal semantics that enables reasoning and inference of global knowledge.

We presented an application that uses this architecture in the context of the Semantic LCMS.

As a further work, in short term, we plan to emphasize the student learning path for identifying the advantages that underpin their semantic construction. In long term we plan to take into consideration the building of the Semantic Kernel of a SLCMS, which will the ontological resources. The benefit of this approach is that the learner’s support and knowledge database will be enriched with new data and information, which will result in a better understanding and faster learning.

5. References