E-Learning Model Based On Semantic Web Technology
Fayed Ghaleb, Sameh Daoud, Ahmad Hasna, Jihad Jaam, Samir A. El-Seoud, and Hosam El-Sofany
Pages 63 - 71

E-Learning Model Based On Semantic Web Technology

Fayed Ghaleb[1], Sameh Daoud[1], Ahmad Hasna[2], Jihad M. ALJa'am[2], Samir A. El-Seoud[3], and Hosam El-Sofany[2]
[1]Department of Mathematics, Faculty of Science, Ain Shams University, Egypt
[2]Department of Engineering and Computer Science, Faculty of Engineering, Qatar University, Qatar
[3]Computer Science Department, Princess Sumaya University for Technology, Amman, Jordan
[hasnah, jaam, helsofany]@qu.edu.qa and selseoud@psut.edu.jo

Abstract: *Research works in the field of E-Learning are represented by a broad spectrum of applications, ranged from virtual classrooms to remote courses or distance learning. Web-based courses offer obvious advantages for learners by making access to educational resource very fast, just-in-time and relevance, at any time or place. In this paper, based on our previous work, we present the Semantic Web-Based model for our e-learning system. In addition we present an approach for developing a Semantic Web-based e-learning system, which focus on the RDF data model and OWL ontology language. We demonstrate the effectiveness of this approach through several experiments using different type of courses taught in Qatar university. The feedbacks of both teachers and students were highly promising.*

Keywords: *E-learning, Semantic Web, RDF, Ontology, OWL.*

Received: May 15, 2006 | ***Revised:*** July 10, 2005 | ***Accepted:*** August 25, 2006

1. Introduction

E-learning is not just concerned with providing easy access to learning resources, anytime, anywhere, via a repository of learning resources, but is also concerned with supporting such features as the personal definition of learning goals, and the synchronous and asynchronous communication, and collaboration, between learners and between learners and instructors [18,19].

One of the hottest topics in recent years in the AI community, as well as in the Internet community, is the *Semantic Web*. It is about making the Web more understandable by machines. It is also about building an appropriate infrastructure for intelligent agents to run around the Web performing complex actions for their users [12]. Furthermore, Semantic Web is about explicitly declaring the knowledge embedded in many web-based applications, integrating information in an intelligent way, providing semantic-based access to the Internet, and extracting information from texts [11]. Ultimately, Semantic Web is about how to implement reliable, large-scale interoperation of Web services, to make such services computer interpretable, *i.e.*, to create a Web of machine-understandable and interoperable services that intelligent agents can discover, execute, and compose automatically [15].

Unfortunately, the Web was built for human consumption, not for machine consumption, although everything on the Web is *machine-readable*, it is not *machine-understandable* [14]. We need the Semantic Web to express information in a precise, machine-interpretable form, ready for software agents to process, share, and reuse it, as well as to understand what the terms describing the data mean. That would enable web-based applications to interoperate both on the syntactic and semantic level.

Note that it is Tim Berners-Lee (inventor of the WWW, URIs, HTTP, and HTML) himself that pushes the idea of the Semantic Web forward. The father of the Web first envisioned a Semantic Web that provides automated information access based on machine-processable semantics of data and heuristics that use these metadata [8,9]. The explicit representation of the semantics of data, accompanied with domain theories (ontologies), will enable a Web that provides a qualitatively new level of service, such as: intelligent search engines, information brokers, and information filters [10].

Researchers from the World Wide Web Consortium (W3C) already developed new technologies for web-friendly data description [7]. Moreover, AI researchers have already developed some useful applications and tools for the Semantic Web [17].

We will introduce the implementation of Semantic Web concept on the e-Learning environment offered by our web-based e-learning system [6], which is used by Qatar University students. The facilities that the application will provide include allowing e-learning content to be created, annotated, shared and discussed, together with supplying resources such as lecture notes, course description, documents, announcements, student papers, useful URL links, exercises and quizzes for evaluation of the student knowledge.

The paper is organized as follows: in Section (2) we present some related works. In section (3) we give a brief overview about the Semantic Web and discuss a number of important issues. In section (4) we introduce the Semantic Web model for our web-based e-learning system. In section (5) we describe the implementation of the system, while in section (6) we evaluate the system. The paper is finally concluded in section (7).

2. Related Works

Recently, several researchers studied the issue of Web-based application. F. P. Rokou *et al*. [28] distinguished three basic levels in every web-based application: *the Web character of the program*, *the pedagogical background*, and *the personalized management of the learning material*. They defined a web-based program as an information system that contains a Web server, a network, a communication protocol like HTTP, and a browser in which data supplied by users act on the system's status and cause changes. The pedagogical background means the educational model that is used in combination with pedagogical goals set by the instructor. The personalized management of the learning materials means the set of rules and mechanisms that are used to select learning materials based on the student's characteristics, the educational objectives, the teaching model, and the available media.

Many works have combined and integrated these three factors in e-learning systems, leading to several standardization projects. Some projects have focused on determining the standard architecture and format for learning environments, such as *IEEE Learning Technology Systems Architecture* (LTSC), *Instructional Management Systems* (IMS), and *Sharable Content Object Reference Model* (SCORM). IMS and SCORM define and deliver XML-based interoperable specifications for exchanging and sequencing learning contents, i.e., learning objects, among many heterogeneous e-learning systems. They mainly focus on the standardization of learning and teaching methods as well as on the modeling of how the systems manage interoperating educational data relevant to the educational process [29].

IMS and SCORM have announced their content packaging model and sequencing model, respectively. The key technologies behind these models are the content package, activity tree, learning activities, sequencing rules, and navigation model. Their sequencing models define a method for representing the intended behavior of an authored learning experience, and their navigation models describe how the learner and system initiated navigation events can be triggered and processed.

Juan Quemada and Bernd Simon have also presented a model for educational activities and educational materials [30]. Their model for educational activities denotes educational events that identify the instructor(s) involved and take place in a virtual meeting according to a specific schedule. F. P. Rokou *et al*. [31] described the introduction of stereotypes to the pedagogical design of educational systems and appropriate modifications of the existing package diagrams of UML (Unified Modeling Language).

The IMS and SCORM models describe well the educational activities and system implementation, but not the educational contents knowledge in educational activities. Juan Quemada's and F. P. Rokou's models add more pedagogical background by emphasizing educational contents and sequences using the taxonomy of learning resources and stereotypes of teaching models. But the educational contents and their sequencing in these models are dependent on the system and lack standardization and reusability. Thus, we believe that if an educational contents frame of learning resources can be introduced into an e-learning system, including ontology-based properties and hierarchical semantic associations, then this e-learning system will have the capabilities of providing adaptable and intelligent learning to learners.

The hierarchical contents structure is able to show the entire educational contents, the available sequence of learning, and the structure of the educational concepts, such as the related super- or sub- concepts in the learning contents. Furthermore, some of semantic relationships among the educational contents, such as 'equivalent', 'inverse', 'similar', 'aggregate' and 'classified', can provide important and useful information for the intelligent e-learning system.

For this purpose, an ontology is introduced in our model. It can play a crucial role in enabling the representation, processing, sharing and reuse of knowledge among applications in modern web-based e-learning systems because it specifies the conceptualization of a specific domain in terms of concepts, attributes, and relationships. Moreover, the number of ontology-centered researches has increased dramatically because popular ontological languages are based on Web technology standards, such as XML and RDF(S), so as to share and reuse it in any web-based knowledge system [23,33]. Thus, we have devised a model that provides the contents structure using an ontology for an adaptive and intelligent e-learning system.

3. Semantic Web Overview

There is a number of important issues related to the Semantic Web. Roughly speaking, they belong to four categories: *Semantic Web languages*, *ontologies*, *semantic markup* of Web pages, and *Semantic Web services*.

Semantic Web Languages: In order to represent information on the Semantic Web and simultaneously make that information both syntactically and semantically interoperable across applications, it is necessary to use specific languages. It is important for Semantic Web developers to agree on the data's syntax and semantics before hard-coding them into their applications, since changes to syntax and semantics necessitate expensive application modifications [20].

There are a lot of such languages around, and most of them are based on *XML* (eXtensible Markup Language), *XML Schemas*, *RDF* (Resource Definition Framework), and *RDF Schemas*, all four developed under the auspices of W3C and using XML syntax [21].

An *XML document* consists of three parts: an XML declaration, a DTD or XML Schema, and an XML instance (XML document data). An XML declaration and schemas are not mandatory for an XML document. An *XML declaration* specifies the version and the encoding of XML being used. A *DTD* or *XML Schema* is a schema that constrains the structure of XML instances, and corresponds to an extended context-free grammar. An *XML instance* is a tagged document.

An XML instance is a hierarchy of elements, the boundaries of which are either delimited by start-tags and end-tags, or, for empty elements, by empty-element tags. Character data between start-tags and end-tags are the content of the element. Figure 1(a) shows an example of an XML instance. A start-tag is the token that encloses an element type with < and >, and an end-tag is the token that encloses an element type with </ and >. Elements can nest properly within each other, and the nesting represents logical structure. Within start-tags, attribute names and attribute values can be specified. Figure 1(b) shows an example of XML Schema.

XML documents have two levels of conformance: *valid* and *well-formed*. A well-formed XML document follows tagging rules prescribed in XML. An XML document is valid if it is well-formed and if the document complies with the constraints expressed in an associated schema.

Definition: *An XML document can be viewed as a tree, where leaf nodes correspond to data values (text) and internal nodes correspond to XML elements.*

RDF is a framework to represent data about data (*metadata*), and a model for representing data about "things on the Web" (*resources*). It comprises a set of triples (*O*, *A*, *V*) that may be used to describe any possible relationship existing between the data – *Object*, *Attribute* and *Value* [7]. Alternatively, each RDF model can be represented as a directed labelled graph, as Figure 2(b), or in an XML-based encoding.

```
<?xml version="1.0"?>
<books>
   <book>
      <author> H.M. Deitel and P.J.Deitel </author>
      <title>  C++ How To Program </title>
      <publisher> Prentice Hall Publishing Co. </ publisher>
   </book>
   <book>
       <author>Jack Herrington</author>
       <title>PHP Hacks</title>
       <publisher>O'Reilly</publisher>
   </book>
</books>
```
(a)

```
<xsd:schema
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<xsd:element name="BOOK" type="BOOKTYPE"/>
<xsd:complexType name="BOOK_TYPE" >
<xsd:element name="AUTHOR" type="xsd:string"
      minOccurs="1" maxOccurs="unbounded"/>
<xsd:element name="TITLE" type="xsd:string"/>
. . .
<xsd:attribute name="isbn" type="xsd:string"/>
</xsd:complexType>
</xsd:schema>
```
(b)

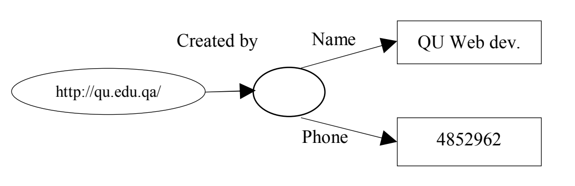
Figure 1. (a) An XML instance, and (b) An example of XML Schema.

Regardless of the representation syntax, RDF models use traditional knowledge representation techniques order to provide better semantic interoperability (traditionally, *O-A-V* triplets are natural semantic units for representing a domain). Still, an RDF model just provides a domain-neutral mechanism to describe metadata, but does not define the semantics of any application domain. Figure 2(a, b) shows that each statement is essentially a relation between an object (*a resource*), an attribute (*a property*), and a value (a resource or free text).

RDF Schema (RDFS) defines the vocabulary of an RDF model. It provides a mechanism to define domain-specific properties and classes of resources to which those properties can be applied, using a set of basic modeling primitives (*class*, *subclass-of*, *property*, *subproperty-of*, *domain*, *range, type*). An RDFS can be specified using RDF encoding, Figure 2(c) shows an example. However, RDFS is rather simple and it still doesn't provide exact semantics of a domain.

Object	*Attribute*	*Value*
http://qu.edu.qa/	created_by	#anonymous_resource1
#anonymous_resource1	name	"QU Web dev."
#anonymous_resource1	Phone	"4851238"

(a)



(b)

```
rdfs:Class rdf:ID="book">
<rdfs:subClassOf rdf:resource="#publication"/>
<rdfs:subClassOf>
...
</rdfs:subClassOf>
</rdfs:Class>
```

(c)

Figure 2. (a) A simple RDF model, (b) the equivalent directed labelled graph, and (c) An example of RDF Schema code.

Ontologies: An *ontology* comprises a set of knowledge terms, including the vocabulary, the semantic interconnections, and some simple rules of inference and logic for some particular topic [13]. Ontologies applied to the Web are creating the Semantic Web [16]. Ontologies provide the necessary armature around which knowledge bases should be built [22], and set grounds for developing reusable Web-contents, Web-services, and applications [23]. Ontologies facilitate knowledge sharing and reuse, *i.e.* a common understanding of various contents that reaches across people and applications.

Technically, an ontology is a text-based piece of reference-knowledge, put somewhere on the Web for agents to consult it when necessary, and represented using the syntax of an *ontology representation language*. There are several such languages around for representing ontologies, see [11] for an overview and comparison of them. It is important to understand that most of them are built on top of XML and RDF.

By 2004, the most popular higher-level ontology-representation languages were *OIL* (Ontology Inference Layer) and *DAML+OIL* [24,25]. An ontology developed in any such language is usually converted into an RDF/XML-like form and can be partially parsed even by common RDF/XML parsers [7]. Of course, language-specific parsers are necessary for full-scale parsing. There is a methodology for converting an ontology developed in a higher-level language into RDF or RDFS [10].

In early 2004, W3C has officially released *OWL* (Web Ontology Language) as W3C Recommendation for representing ontologies [7]. OWL is developed starting from description logic and DAML+OIL. The increasing popularity of OWL might lead to its widest adoption as the standard ontology representation language on the Semantic Web in the future. Essentially, OWL is a set of XML elements and attributes, with well-defined meaning, that are used to define terms and their relationships (e.g., *Class*, *equivalentProperty*, *intersectionOf*, *unionOf*, etc.). OWL elements extend the set of RDF and RDFS elements, and the *owl* namespace is used to denote OWL encoding. Figure 3 shows a piece of a simple ontology developed using the OWL language.

In practice, ontologies are often developed using integrated, graphical, *ontology-authoring tools*, such as Protégé-2000, OILed, and OntoEdit [26]. They are used to develop new ontologies and modify existing ones. They let the author edit and develop ontologies concentrating on the domain's concepts and relationships, without worrying much about ontology-representation languages. The author can choose ontologies from a list, choose attributes and relations from another list, edit, add, remove, and merge ontologies. The output is usually produced in a specific high-level ontology-representation language such as OWL, RDF/RDFS, HTML, or in plain text.

```
<owl:Class rdf:ID="Description">
   <rdfs:subClassOf rdf:resource="#Course"/>
   <owl:disjointWith rdf:resource="#Documents"/>
   <rdfs:seeAlso rdf:resource="#Useful_links_7"/>
</owl:Class>
```

Figure 3. A simple ontology defined in OWL

Semantic Markup: Ontologies merely serve to standardize and provide interpretations for Web content, but are not enough to build the Semantic Web. To make Web content machine-understandable, Web pages and documents themselves must contain semantic markup, i.e. *annotations* which use the terminology that one or more ontologies define and contain pointers to the network of ontologies. Semantic markup persists with the document or the page published on the Web, and is saved as part of the file representing the document/page. Services also must be properly marked-up, to make them computer-interpretable, use-apparent, and agent-ready. They must contain pointers to the corresponding service ontologies.

Semantic markup of a Web page, document, or service might state that a particular entity is a member of a class, an entity has a particular property, two entities have some relationship between them, and that descriptions from different people refer to the same entity. Typically, semantic markup is published using an XML encoding for a high-level ontology-representation language syntax [12,27].

The annotation is done by using appropriate tools. These tools can be part-of or integrated with ontology-authoring tools, such as OIL tools [16]. They can also be standalone tools, such as the Knowledge Annotator tool [12]. Furthermore, they can be integrated with specific Semantic Web applications. An example of this last approach is ITtalks, a fielded application that facilitates user and agent interaction for locating talks on information technology [17], which automatically generates

DAML+OIL descriptions of user profiles when they register.

Semantic Web Services: Intelligent, high-level services like information brokers, search agents, information filters, intelligent information integration, and knowledge management, are what the users want from the Semantic Web. They are possible only if a number of ontologies populate the Web, enabling semantic interoperation between the agents and the applications on the Semantic Web, *i.e.* semantic mappings between terms within the data, which requires content analysis.

One specific kind of ontology is necessary to enable high-level Semantic Web services - ontologies of services themselves [15]. These ontologies should include a machine-readable description of services (as to how they run), the consequences of using the service (*e.g*., the fee), and an explicit representation of the service logic (*e.g.*, automatic invocation of another service). Services have their properties, capabilities, interfaces, and effects, all of which must be encoded in an unambiguous, machine understandable form, to enable agents to recognize the services and invoke them automatically.

4. E-Learning Model Based On Semantic Web

In the following subsections, based on the Semantic Web technology and e-learning standards we describe our proposed e-learning model, illustrated in Figure 4.

The Web-based Services: Our model in Figure 4, provides the student with two kinds of contents, *Learning content* and *Assessment content*. Each content has different types of services such as:

- *Learning services*: provide registration, online course, interactive tutorial, course documents (is a repository for files that the instructor have made available to the student as a part of your course), announcements (displays information to the students that the instructors of the course want him to know), links (displays a list of useful URL links that have been identified by the course instructors), student papers (students can post/upload requests files to the instructor), and Semantic search (helps the student to search for resources).
- *Assessment services*: provide exercises and quizzes for evaluation of the student knowledge.

During the learning process, a dynamic selection presentation of both contents will be accomplished.

On other hand, our e-learning system allows instructors to create his course websites through a browser, and monitoring the students performance. they have many services and tools such as: publish documents in any format (Word, PDF, Video, ...) to the students, manage a list of useful links, compose exercises/quizzes, make announcements, and have students submit papers.

To illustrate the services architecture, we will go through an e-learning scenario. A student first searches for an online course: the broker handles the request and returns a set of choices satisfying the query. If no course is found, the user can register with a notification service. Otherwise, the user may find a suitable course among the offerings and then makes a final decision about registering for the course.

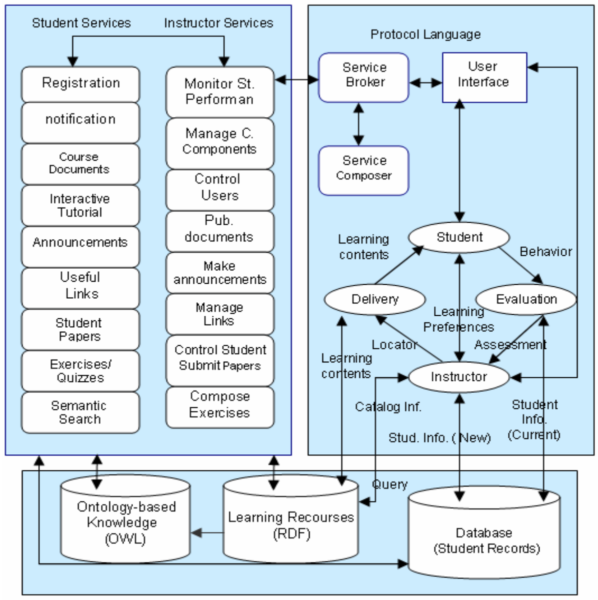


Figure 4. Proposed model for web-based e-learning system

Processing the registration can be seen as a complex service involving registering with the system, creating a confirmation notification, creating a student account (authentication/ authorization), and providing learning materials. Once all these in place, the student can start the course. As part of the course, a student will be logging on and checking his learning agenda (e.g. next assignment due). This request is answered by combining several sources of information, such as course schedule, current date and student progress to date (e.g. completed units).

The Ontology-based Model: Before describing our ontology-based model, we will discuss learning environments illustrated in Figure 4. Course sequencing generally starts with the *student entity* component that receives the learning contents, while the student's behavior is being observed. The *instructor* sends queries to the *learning resources* to search for learning content that is appropriate for the student entity component. The *ontological knowledge* is added to the learning resources as a resource for contextual learning, and it may be searched by means of queries. The student's performance is measured by the *evaluation* component, and the result is stored in the student records *database*. The data in the database can be used by the *instructor* component to locate a new content.

Searching learning resources and sequencing a course can be done using a knowledge base of learning resources and a *delivery* component. To implement the knowledge base, first of all, the leaning resources have to be described by means of metadata. The metadata consists of the contextual knowledge of the learning resources*, i.e.*, an ontology in our model. It contains the general representation of the structural knowledge on specific domains, such as computer science, mathematics, biology, and so on.

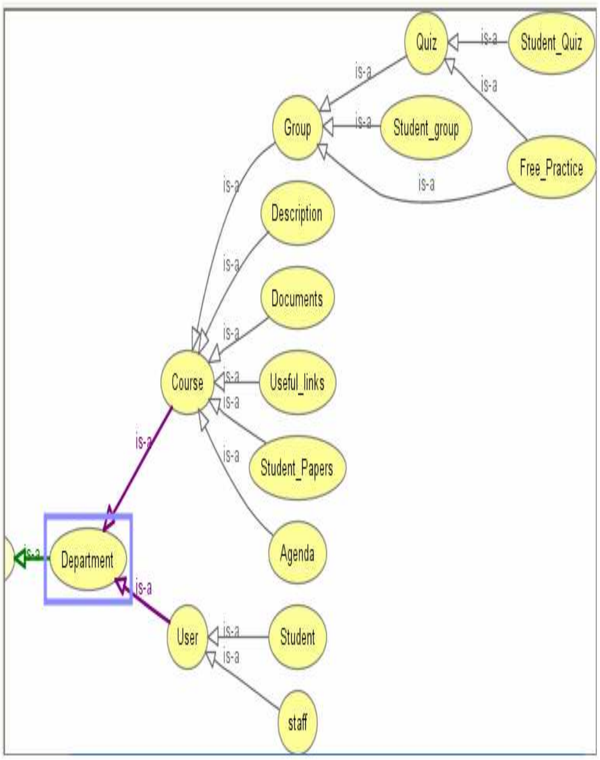


Figure 5. A snapshot of the proposed ontology using Protégé 2000.

5. Implementation

The ontology can be used for adaptive learning to retrieve the context of a course and to structure the contents. Also the metadata actually consists of the framing description of each learning object of a subject, *i.e.*, the modularized content, which is linked to the concept of the ontology. For instructors to be able to sequence courses and create exercises adaptively, the suitability of different approaches has to be analyzed based on the relationships between the resources and their descriptions. Figure 5 shows a snapshot of our e-learning ontology with the classes and properties in the Protégé 2000 ontology editor.

The main agents used in our system are: *Student* and *Instructor*, both of them are implemented as PHP classes, as illustrated in Figure 4. Users are served by the appropriate agents, which parse the metadata and tailor the user interface to satisfy the user's needs, whether student or instructor. The agents interact and communicate between each other by means of PHP, MySQL database, and using the Apache Web Server. Figure 6, show a snapshot of our proposed system.

Users will add any metadata to a document referenced via the RDF learning resources repository through dynamic PHP web pages. For the end-user, this process of *annotation* is identical to the action of filling out fields in a Web form. After the user submits the form, the application automatically converts this additional information to a set of RDF statements using the RAP API, and then adds them to the existing RDF statements for this document in the repository. Because the RDF specifications provide an XML syntax for writing down and exchanging RDF statements (called RDF/XML), the repository is implemented as a set of RDF/XML files. However, the RDF/XML syntax is quite complex and developing an RDF parser is not a trivial task.

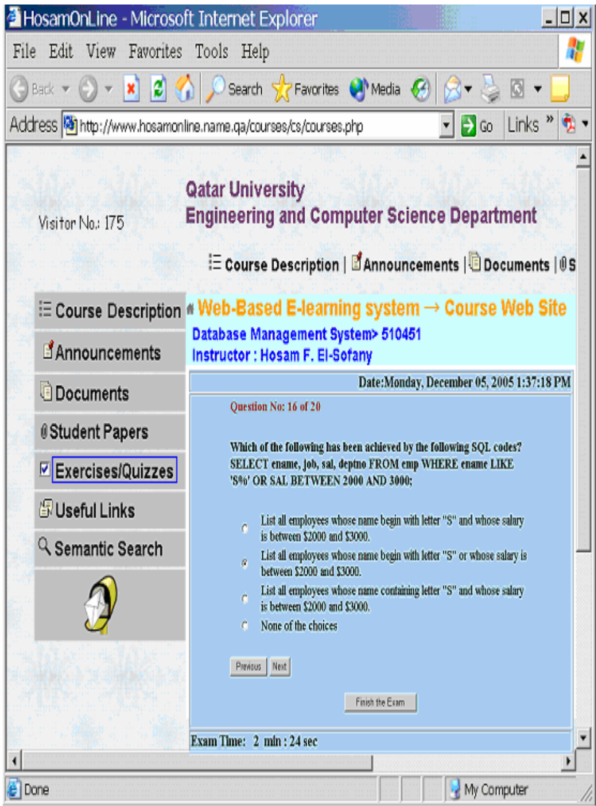


Figure 6. A snapshot of the proposed system.

Motivated by the need for an RDF parser, we are using a Semantic Web toolkit called RAP for developing our application. In the following sub-sections, we will illustrate some features of the RAP API.

RAP (RDF API for PHP): RAP is a Semantic Web toolkit for PHP developers. It offers features for parsing, manipulating, storing, querying, serving, and serializing RDF graphs. RAP was started as an open source project by the Freie Universität Berlin in 2002 and has been extended with code contributions from the Semantic Web community.

The core of RAP are two implementations of statement storages which hold RDF graphs either *in-memory* or in *a relational database*. Around these storages RAP provides rich programming interfaces for manipulating RDF graphs on different abstraction layers. Furthermore, RAP supports RDFS inference as well as some OWL entailments, allowing programmers to work with implicit (virtual) statements. Various tools complement the RAP package: an up-to-date RDF/XML parser, an integrated RDF server, and a graphical user-interface for managing database-backed RDF models as well as an implementation of the RDQL query language.

6. System Evaluation

To obtain some feedback about our Semantic Web-based e-learning system, we demonstrate the effectiveness of our model through several experiments using different type of courses taught in Qatar university. In this section we selected the Exercises/Quizzes service provided by the system and present the feedbacks of the students, see Figure (7). We have prepared a questionnaire and distribute it to the potential students registered in the system.

Two groups of 30 students each, have been randomly selected to use the system as a testing pool. The first group of 30 students used the hard paper quiz and the second group used our Semantic Web-based e-learning system. Table 1, shows that both groups obtained approximately similar results. Students had to obtain a grade of 60% to pass the quiz.

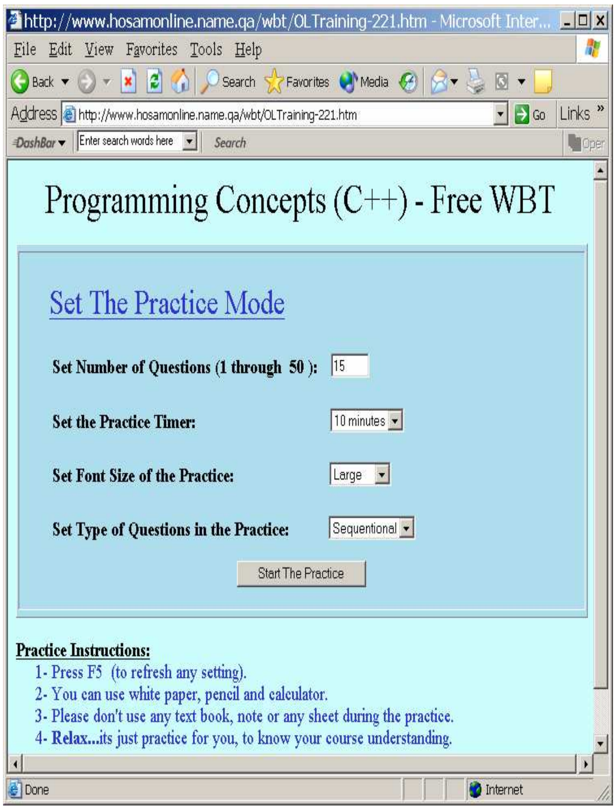


Figure 7. A snapshot of the Exercises/Quizzes service

After taking the quiz, all the students had to fill in a questionnaire with general and specific questions related to the method of testing. The most significant questions reflecting student opinion are set out in Table 2, with the relevant group responses.

Table 1. Quiz results of the two groups students

Student Groups	Passed	Percentage
A – Paper Quiz	22	% 73.33
B- Computer Quiz	25	% 83.33

Table 2. Results of the questionnaires of both groups

	Group A (%)	Group B(%)
(1) *How do you feel your result would have been, if the quiz had been by the on-line quiz system* (*i.e., computer-based*)?		
Worse	40	-
Same	30	-
Better	20	-
Do not know	10	-
(2) *How do you feel your result would have been, if the Quiz had been on paper as the traditional way*?		
Worse	-	25
Same	-	40
Better	-	5
Do not know	-	30
(3) *How would you describe the process of entering answers in the computer*?		
Easy	-	75
Acceptable	-	22
Difficult	-	3
Very difficult	-	-
(4) *The quiz questions were presented with the all-in-one format. How many question do you prefer to work with at the same time.*		
Less than three	-	70
Three	-	20
More than three	-	10
(5) *When do you prefer to be told your quiz score*?		
At the end	At the end	At the end
Later	Later	Later
(6) *Do you trust the on-line evaluation system*?		
Yes	Yes	Yes
No	No	No

The results of question 1 indicate that, a large percentage of student who did not use the computer-based system felt that their marks would have been worse if a computer had been used. Similarly, question 2 shows that a percentage of student using computers felt that they might have done better using traditional methods These students continuously use computers and they had no problems introducing the answers. However, in some cases they commented that a time allowance should be given understanding the instructions.

The results of question 4 indicate that, three questions on the same web page it too many, and that student prefer fewer questions so that using the scroll bar is unnecessary. This is optional and easily reconfigurable.

Student who used the Semantic web-based e-learning system obtained their marks immediately after submitting their responses to the server. Student who did the exam on paper received the results the day after the exam. The response shows that independently of the method used, a vast majority of students want to know their grade as soon as possible as well as the correct answers..

Finally, the majority of the students trust more a computer-based evaluation system than classical methods. Some students did comment on the absence of printed copies of their answers and the fact that they could not compare their answers with the correct results after the quiz.

7. Conclusions

The main contribution of this paper is our new model for e-learning system, using the Semantic Web technology. Our model including various services and tools in the context of a semantic portal, such as: course registration, uploading course documents and student assignments, interactive tutorial, announcements, useful links, assessment, and simple semantic search. A metadata-based ontology is introduced for this purpose and added to our model. The OWL language is used to develop our ontologies. In these ontologies, the actual resources and properties specified in the RDF models are defined. The Protégé 2000 ontology editor is used to create the e-learning ontology classes and properties.

A list of the technologies used in the implementation of our web-based e-learning system includes PHP Platform, Apache Web Server, MySQL database, and RAP Semantic Web Toolkit.

We believe that there are two primary advantages of our Semantic web-based model. One is that the proposed model, which contains a hierarchical contents structure and semantic relationships between concepts, can provide related useful information for searching and sequencing learning resources in web-based e-learning systems. The other is that it can help a developer or an instructor to develop a *learning sequence plan* by helping the instructor understand the why and how of the learning process.

References

[1] Berners-Lee T. "What the semantic web can represent",http://www.w3.org/DesignIssues/RDF/not.html , 2000.

[2] Downes, S. "Learning Objects: Resources For Distance Education Worldwide". *International Review of Research in Open and Distance Learning*. 2001.

[3] Maurer, H, M. Sapper. "E-Learning Has to be Seen as Part of General Knowledge Management". *ED-MEDIA Conference*, 2001.

[4] Naeve, A, M. Nilsson, M. Palmer. "The Conceptual Web – Our Research Vision". *The 1st Semantic Web Working Symposium*, 2001.

[5] J.Davies, D.Fensel, and F. Harmelen, "Towards The Semantic Web" ,*Willy* 2002.

[6] Hosam F. El-Sofany, Ahmad M. Hasnah, Jihad M. Jaam and Fayed F. M. Ghaleb., "A Web-Based E-Learning System Experiment". *Proc. of the Intel. Conf. on E-Business and E-learning*, PSUT, Amman-Jordan, 112-119, 2005.

[7] W3C site: http://www.w3c.org. (see www. w3 .org/XML,www.w3.org/RDF, www. w3.org /TR/2004/REC-owl-features-2004 0210/), and http://www.w3.org/2000/10/swap/ Primer.html.

[8] Berners-Lee, T., Hendler, J., & Lassila, O. The Semantic Web. *Scientific American* 284, 34–43, May 2001.

[9] Berners-Lee, T., Fischetti, M., & Dertouzos, T. M. Weaving the Web: *The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. San Francisco: Harper, 1999.

[10] Decker, S., Melnik, S., van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., & Horrocks, I. The Semantic Web: The Roles of XML and RDF. *IEEE Internet Computing* 4, 63-74, Sep./Oct. 2000.

[11] Gómez-Pérez, A., & Corcho, O. Ontology Languages for the Semantic Web. *IEEE Intelligent Systems* 17(1), 54-60, 2002.

[12] Heflin, J., & Hendler, J. A Portrait of The Semantic Web in Action. *IEEE Intelligent Systems* 16(2), 54-59, 2001.

[13] Hendler, J. Agents and the Semantic Web. *IEEE Intelligent Systems* 16(2), 30-37, 2001.

[14] Lassila, O. Web Metadata: A Matter of Semantics. *IEEE Internet Computing* 2(4), 30-37, 1998.

[15] McIlraith, S.A., Son, T.C., & Zeng, H. Semantic Web Services. *IEEE Intelligent Systems* 16(2), 46-53, 2001.

[16] Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D.L., & Patel-Schneider, P.F.

OIL: An Ontology Infrastructure for the Semantic Web. *IEEE Intelligent Systems* 16(2), 38-45, 2001.

[17] Scott Cost, R., Finin, T., Joshi, A., Peng, Y., Nicholas, C., Soboroff, I., et al. ITtalks: A Case Study in the Semantic Web and DAML+OIL. *IEEE Intelligent Systems* 17(1), 40-47, 2002.

[18] Barker, P. Developing Teaching Webs: "Advantages, Problems and Pitfalls". *Educational Multimedia, Hypermedia & Telecommunication (AACE) Conference*, 2000

[19] Drucker, P. Need to Know – "Integrating e-Learning with High Velocity Value Chains". Delphi Group White Paper. www.delphigroup .com, 2000.

[20] Wuwongse, V., Anutariya, C., Akama, K., & Nantajeewarawat,E. XML Declarative Description: *A Language for the Semantic Web. IEEE Intelligent Systems* 17(1), 54-65, 2002.

[21] Klein, M. Tutorial: The Semantic Web - XML, RDF, and Relatives. *IEEE Intelligent Systems* 16(2), 26-28, 2001.

[22] Swartout, W., & Tate, A. Ontologies, Guest Editors' Introduction. *IEEE Intelligent Systems* 14(1), 18-19, 1999.

[23] Devedzic, V. Knowledge Modeling - State of the Art. *Integrated Computer-Aided Engineering* 8, 257-281, 2001.

[24] Horrocks, I., Fensel, D., Broekstra, J., Decker, S., Erdmann, M., Goble, C., et al. (2002). The Ontology Inference Layer OIL, Tech. Report, *Vrije Universiteit, Amsterdam. Retrieved March* 19, 2002, from http://www.ontokn- owledge.org /oil/TR /oil.long.html.

[25] Horrocks, I. & van Harmelen, F. Reference Description of the DAML+OIL Ontology Markup Language. Retrieved March 19, 2002, *from http://www.daml.org/2000/ 12/ reference. html* , 2002.

[26] Protégé-2000: http://protege. stanford.edu/, OILed:http://img.cs.man.ac.uk/oil, and OntoEdit: http://ontoserver.aifb.uni-karlsruhe.de/ontoedit.

[27] Tallis, M., Goldman, N.M., & Balzer, R.M. The Briefing Associate: Easing Authors into the Semantic Web. *IEEE Intelligent Systems* 17(1), 26-32, 2002.

[28] F. P. Rokou et al., "Modeling web-based educational systems: process design teaching model," *Educational Technology and Society*, Vol. 7, pp. 42-50, 2004.

[29] H. Adelsberger et al., "The Essen model: a step towards a standard learning process," *http://citeseer.ist.psu.edu/515384.html*, 2003.

[30] J. Quemanda and B. Simon, "A use-case based model for learning resources in educational mediators," *Educational Technology and Society,* Vol. 6, pp. 149-163, 2003.

[31] M. D. Merrill, "Knowledge objects and mental-models," *http://reusability.org/read*, 2003.

[32] Y. Sure et al., "Methodology for development and employment of ontology based knowledge management applications," *ACM SIGMOD Record*, Vol. 31, pp.18-23, 2002.

[33] C. Brewster et al., "Knowledge representation with ontologies: the present and future," *IEEE Intelligent Systems,* Vol. 19, pp. 72-81, 2004.

Fayed Ghaleb received a Ph.D. degree in Mathematics from Moscow State University, Moscow USSR, June, 1978. He is currently a Professor E. at the Mathematics Department, Faculty of Science, Ain Shams Univ., Egypt. His research interests include Applied Functional Analysis and Spectral Theory, Graph Theory, Data Mining, and E-Learning. Dr Fayed is a board member at the Egyptian Mathematical Science Research Centre, he is also the Treasurer of the Egyptian Mathematical Society (ETMS).

Hosam F. El-Sofany received his M.Sc. degree in Computer Science from Ain Shams University, Cairo, Egypt. He is currently an Instructor at the Department of Engineering and Computer Science at Qatar University. Since 2004, he is working towards his Ph.D. in Computer Science from Faculty of Science, Ain Shams University. His research is focused on E-Learning, XML Databases, Relational Databases Systems, and Semantic Web Applications.