Supporting Interoperability between Web-based Educational Systems

Ig Ibert Bittencourt, Evandro Costa, Lucas Braz and Henrique Pacheco
Federal University of Alagoas, Alagoas Brazil, ibert@dsc.ufcg.edu.br, ebc@fapeal.com

Darina Dicheva
Winston-Salem State University, USA, dichevad@wssu.edu

Abstract - Current opportunities for students to improve their knowledge in a certain domain (e.g. Computer Science) through interaction with Computer-based Educational Environments are increasing. The large number of students using such systems and their different knowledge and goals motivate the use of technologies to provide personalized activities, resources, and content to the students. Such environments are called Adaptive Educational Systems. However, current efforts in this direction fail on providing external resources, content, or even interaction with peers. Therefore, systems are needed that interoperate, collaborate and exchange content or reuse functionality, in order to support a richer set of educational functions and increase their effectiveness. The semantic web technologies have been proposed to ensure interoperability between web applications to support such requirements. In this paper is proposed a semantic web-based approach to provide interoperability between (concept/ontology-based) educational environments. It is also described a case study that illustrates the interoperation of educational environments through the use of shared ontologies/knowledge.


INTRODUCTION

Computer-based Educational Environments have been used to provide opportunities to students to improve their knowledge in a certain context through interactions with the system. For this reason, there is an increasing interest in online learning support systems that are aimed at providing resources and functional components for various educational goals and tasks. The number of students using such systems and their different knowledge and goals motivates the use of technologies to provide personalized activities, resources, and content to students. These systems are called Adaptive (Educational) Systems, having to alter aspects of their structure, functionality or interface in order to accommodate the different needs of individuals or groups of users and the changing needs of users over time. However, it is common in such environments to fail to provide some resources, content, or even interaction with peers. As a result, such systems often need to inter-operate, collaborate and exchange content or re-use functionality, in order to support a richer set of educational functions and increase their effectiveness. Indeed, one of the ways to provide a system with the necessary content, functionalities, and knowledge is through its communication with other educational environments.

In other words, the way point out is to ensure interoperability between educational environments in order for them to provide content, functionalities, and knowledge one to another thus improving the quality of the system-student-system interaction. Several problems are related to providing interoperability between educational environments, such as, different knowledge representation formalisms, different languages, automation, scalability, and so on. Some researchers pointed out that the decomposition of a system into modules can be used to provide steps towards the solution of the problem. Another important direction for providing interoperability between applications is the use of semantic web technologies. The Semantic Web in general is becoming popular with its support for semantic descriptions of the content and machine processing of the web resources [1].

Several proposals concerning interoperability between educational environments have been proposed by the AIED community. However, these approaches are lacking in scalability and automation. They do not provide architectures that ensure the interoperation between several educational environments and the interoperability is achieved manually. As a consequence, to provide interoperation with a new environment, it is necessary to specify, implement, and configure the interaction. On the other hand, general problems are identified in ensuring interaction between educational environments, such as application discovery, service discovery and match, problems with concepts (synonyms, different names, different meanings, so on), and unshared ontologies/knowledge.

This paper proposes an approach to provide interoperability between (concept/ontology-based) educational environments. The proposed architecture emphasizes two aspects. First, it assures the integration of several educational environments and the interoperability is achieved manually. As a consequence, to provide interoperation with a new environment, it is necessary to specify, implement, and configure the interaction. On the other hand, general problems are identified in ensuring interaction between educational environments, such as application discovery, service discovery and match, problems with concepts (synonyms, different names, different meanings, so on), and unshared ontologies/knowledge.

This paper proposes an approach to provide interoperability between (concept/ontology-based) educational environments. The proposed architecture emphasizes two aspects. First, it assures the integration of several educational environments and the interoperability is achieved manually. As a consequence, to provide interoperation with a new environment, it is necessary to specify, implement, and configure the interaction. On the other hand, general problems are identified in ensuring interaction between educational environments, such as application discovery, service discovery and match, problems with concepts (synonyms, different names, different meanings, so on), and unshared ontologies/knowledge.
a result, the architecture provides facilities to deal with the challenges (development complexity, time development cost, unshared ontologies/knowledge, and others) in order to provide interoperability functionalities.

To demonstrate the feasibility of the proposed architecture, a case study illustrating the interoperation of educational environments through the use of shared ontologies/knowledge is presented. The case study also describes how to automate the interoperability process through the use of an agent-based approach.

**PROPOSED ARCHITECTURE**

This section describes the proposed architecture aimed at ensuring interoperability between Adaptive Web-based Educational Environments (AWBESs). In order to allow semantic interoperability, these systems have to be concept-based or ontology-based. The implementation of such an architecture and AWBESs that provide interoperability is a hard and time consuming task. For this reason, as already mentioned, we decouple the complexity to allow the interaction between environments, emphasizing two aspects – the use of semantic web technologies to assure the integration of several educational environments and the use of semantic web services to extend interoperability functionalities. Figure 1 shows the proposed architecture and its components.

![Figure 1. Architecture Supporting interoperability between AWBES.](image)

This architecture represents an extension of the generic architecture proposed in [2]. The idea was to provide an implementation of the architecture proposing and describing the technologies. A description of each component of the architecture is provided below.

**I. Ontology Repository**

In order to guarantee the discovering and interoperability between educational environments, there is a need for a repository where the environments would provide an instance of the ontology about the environment. In other words, each environment is described through an application ontology and then each environment has an instance of the application ontology in a repository.

The use of ontologies is motivated due to their capability of providing a vocabulary for knowledge representation in a fashion that the share and reuse of knowledge is guaranteed. In other words, they are attempts to more carefully define parts of the data world and to allow interaction between data held in different formats [1]. Additionally, ontologies have been addressed by the community as an important requirement to assure the interoperability between educational environments. In fact, the repository has an application ontology with two kinds of information:

**Application Information:** The application ontology defines the necessary information for the environments to get registered and be discovered. This ontology is presented using OWL-DL; some of its classes are described below (See Figure 2):

- **Environment:** it represents general information about the registered educational environment (for instance, e-Mathema, TM4L). This is the main class because it defines the relation with the entities and services available in the environment;
- **Category:** represents the context of the related environment, for instance, Medicine, Math, and others;
- **Entity:** entities which are responsible for performing the service provided by the environment. Possible values of the entityType slot are for instance, agent-based, component-based, service-oriented, or object-oriented;
- **Service:** the functionalities provided by the environments are implemented by services. It is through services that the environments are able to discover whether other environments are available for collaboration.

![Figure 2. Application Ontology.](image)
Alignment Information: This represents an excerpt of the application ontology which has information about the aligned ontologies. The goal is to identify whether the ontology of an environment is aligned with others ontologies. As defined in Figure 3, the class Environment has general information about the educational environment and it is also defined the ontologies present in it. However, these ontologies can be aligned with others. For this reason, the class Education describes the ontologies of the environment and the class Alignment has the URI of the other ontology. In fact, there is no tool available that can ensure the automatic mapping between educational ontologies. For this reason, this is a manual step. Figure 3 describes the information about the mapped ontologies. Basically, it is an extension of the application ontology on which the Environment class has defined their educational ontologies and which ontologies are mapped with them.

II. Interoperability Server

The Interoperability Server is responsible for providing the semantic web services that ensure the registration of educational environments as well as the discovery of environments capable of communicating with each other. Semantic Web Services (SWS) support a number of different activities transforming a static collection of information into a distributed way on the basis of semantic web technology making the content within the WWW machine processable and machine-interpretable [3]. As a result, it is possible to automatically discover, compose, and invoke them. In order for the server to ensure the discovery of environments capable to communicate, a number of services are available as described below.

- Register Service: it is responsible for registering educational environments in the ontologies repository. In order for the registry to be done, an instance of the application ontology is needed. As a result, the output of the service is an identifier (ID) to the educational environment;
- Connect Service: it allows the environments to connect to the repository (through their ID). As a result, the output of the service is a Boolean type informing whether the registration succeed or not;
- Request Service: this service is used by the environments to request external help from other education environments. In other words, this service indicates whom the environments are able to request resources from. This service receives the name of a desired service as an entry (e.g. Recommend) and returns the tuple < entity, hostEntity, service, inputService, outputService>;
- Discovery Service: this service is the most complex service because it is responsible for composing the semantic web services with regard to the verification concepts in order to discover a new functionality available in other educational environment.

III. Educational Environments

Educational Environments represent the group of systems registered in a repository. They have an indirect relation with the ontology repositories. Although the ontology repositories keep instances about the description of the environments and access to their educational ontologies and alignments other ontologies, the communication with the environments happens through the semantic web services. Moreover, the adaptive web-based educational environments could be configured and implemented to allow automatic interoperability with other environments. This is possible through the components provided on the architecture and the use of semantic web technologies. However, it is also important to implement the AWBES with semantic web ontologies, otherwise the automatic interaction will be hard to be provided.

Furthermore, the educational environments invoke semantic web services in three situations: 1) First access: when the system will be available by the first time, it must register the environment in the repository; 2) New access: it happens when the system was off-line and on-line afterwards; then the environment must invoke the connect service, and 3) Solicit Help: it is used when the system identifies a situation which requires it to interoperate with another environment.

The next section provides more detail about these processes.

CASE STUDY

This case study describes a framework for building Semantic Web-based Educational Systems (SWBES), called e-Mathema [4]. The main goal of e-Mathema is to increase the efficiency and re-use in the development of such
applications. It is aimed at: (1) assuring low time and cost for building such systems, with a minimal amount of code modification; (2) producing applications adaptable to user needs; (3) involving autonomous tutoring agents, knowledge, and inference capabilities [5]. The technologies used in the development of the framework include Tomcat, Jade, Protégé, OWLSEditor, Mindswap, and OWL-DL. Figure 4 shows the framework architecture.

Figure 4. e-Mathema Architecture.

Details about the architecture components are described below:

- **Persistence Services**: implement persistence mechanisms by using Jena and Hibernate;
- **Agents**: assure the adaptiveness of the learning process. They include Controller Agent, Mediator Agent and heterogeneous agents forming an Agent Society: MA (Mediator Agent): this agent has three functionalities: (i) to recommend actors (students, experts, agents) according to the solicitor agent necessities; (ii) to coordinate the complex problem solving process, and (iii) to intermediate the communication with other educational environments in order to assure the interoperation between them; CA (Controller Agent): it has three fundamental skills: (i) start agents, (ii) add, remove, and update society agents, and (iii) add, remove and update agents’ services; Agent Society: it is composed of (i) support agents, which provide support to the others agents in order for them to ensure the accomplishment of their activities (examples of these agents are support agents for mining the web) and (ii) autonomous tutoring agents (ATA) that have information about educational aspects (cognitive, motivational and affective)[6];
- **Semantic Web Services**: they are responsible for automatic discovery, compositions and invocation of services by the agents.

Accordingly, the mediator agent is responsible for providing interoperability between e-Mathema and other learning environments. Details about the support for interoperability are given below.

**Identification**: This phase identifies the need for interoperability. In e-Mathema this needs arises in seven different situations to fall into two main categories:

- **Recommendations**: they are used to provide additional help to students according to different situations. These include recommendations of students, experts, and autonomous tutoring agents. Scenarios for each one these cases are described below: Scenario 1 (Student Recommendation): A student uses e-Mathema as a support tool in his coursework. When trying to solve a specific task, however, he doesn’t succeed, even with the hints provided by the tutoring agents. Consequently, the student announces this and e-Mathema seeks for other students who could help him to figure out the answer; Scenario 2 (Expert Recommendation): This is a continuation of the previous scenario. When the student tries to solve the problem and doesn’t succeed again, he announces this and e-Mathema seeks for an expert to help him. This situation can also occur when the system seeks for a student and cannot find one; Scenario 3 (ATA Recommendation): A student configures a new problem for the system to solve. When the system decomposes the problem, there can be some tasks which the ATA is not able to solve. In this case, the system might seek for ATAs capable of solving those tasks;

- **Educational Resources**: they represent the learning objects in a specific educational system, including examples, problems, counter-examples, units of activities, etc. Four scenarios were identified to provide interoperability as follows: Scenario 4 (More detailed/relevant information): A student is using e-Mathema and trying to solve a specific task, however, the student is not satisfied by the informational support provided by e-Mathema since it is not enough for him to achieve his learning goal. The student announces this and the system seeks for more relevant information externally; Scenario 5 (New content): When the student would like to receive information about requirements to understand contents he can ask the system for additional information from other environments; Scenario 6 (New problems/exercises): When a student is not satisfied with the exercises/problems provided by the system, he can announce that he would like new problems/exercises; Scenario 7 (New examples): The student is studying and he does not understand a concept. Consequently, he asks the system for new examples to help him understand the concept. As a result, the system tries
to retrieve new examples: if not successful, it seeks external help.

Request: After the identification of an interoperability scenario, e-Mathema should react appropriately. First of all, it is necessary to inform the first actions of the environments. In fact, the first action of an e-Mathema environment when it is deployed is to register or connect to the Interoperability Server. When the environment is started, the ControllerAgent is automatically created. After that, the MediatorAgent is created and this agent verifies if it is the first execution of the environment in order to register it, otherwise the MediatorAgent simply connects the environment. The semantic web services used are described as follows (see Figure 5):

1. **RegisterService**: this service is used to register a new environment to provide interoperability feature;
2. **ConnectService**: it is responsible for enabling the environment to interoperate with others.

![Figure 5. Sequence Diagram describing e-Mathema Registration and e-Mathema Connection.](image)

Afterward, the MediatorAgent has a sensor to perceive the scenarios and a set of rules to decide if external help is necessary (see Figure 6).

**Recommendation Rule**

```plaintext
if student not succeed to solve a task then
    recommend student
if new student to help not exists
    then recommend expert
else
    if not succeed to solve a task
    then recommend expert
else
    if not succeed to solve a task
    then recommend expert
```

**Educational Resource Rule**

```plaintext
if student requires educational resources then
    seek local educational resource
if not exists local educational resource
    then require external educational resource
```

![Figure 6. Example rules.](image)

As shown in Figure 7, a specific agent (in this case, called AgentX) requires help from the Mediator Agent that performs the rules to verify which kind of help the solicitor is requiring. If external help is needed, the MediatorAgent invokes the semantic web services to seek for an agent able to help AgentX.

![Figure 7. Sequence Diagram describing when MediatorAgent requires external help.](image)

Discovery: In order to present the discovery step, it is necessary to describe details about the instance of the application ontology used to register an e-Mathema environment.

In addition, some semantic web services are used to ensure the discovery process of new agents able to interoperate. However, this process must discover, compose, and invokes the semantic web services. As a result, a matchmaking algorithm [7] is used to provide such functionality. The following services are provided:

1. **RequestHelpService**: it is used by the environments to request help from the interoperability server;
2. **DiscoverService**: this service is responsible for discovering entities inside the environments that are able to interact with the environments seeking help. This service gets the environments provided by the DiscoverEnvironmentsService and tries to find if any entity inside them is able to interact. This service uses verification services (e.g. IndexVerificationService) to provide a more precise answer. It uses the instances of the Application Ontology (presented in Figure 8), more specifically:
   a. **Entity and Service classes**: which represent the entities (components) available to interact inside an environment and their respective services;
   b. **entityType** parameters: which are related to the interaction between two entities (e.g. if they have the same infrastructure the interaction is faster and easier than in case of different infrastructures);
   c. All the parameters of the **Service** class: used to identify if the required functionality can be accomplished by the services of the entities.
3. **IndexVerificationService**: this service finds the similarity between two terms (e.g. the similarity between the terms recommendation and recommender is 100% since they stem out from recommend).

Communication: This step is quite simple in case of systems with the same infrastructure because no additional service is needed to ensure the communication (it occurs directly). In other words, when the MediatorAgent gets the
information of an agent able to provide the requested functionality, e.g. AgentY (see Figure 7), the communication between AgentX and AgentY occurs directly, as shown in Figure 9.

Figure 9. Sequence Diagram describing the interaction between agents from different environments.

**RELATED WORK**

Several works have approached the interoperability problem through the use of Semantic Web’s emerging technologies. Some of these works are described below.

In [8], a model for interoperability assurance of Semantic Web distributed applications is presented. The solution makes use of an ontology alignment based approach, which is enabled through the use of POAF and SWEDER. The positive aspect of such an approach is that POAF has an alignment associated methodology. However, the approach assures interoperability through ontology levels only, which causes some difficulties to the assurance of ontologies/environments evolution along with the insertion of new distributed environments.

[9] proposes a framework for user models interoperability through the use of ontologies and a service-oriented approach. However, the search and discovery mechanisms are defined manually, making the model update a very slow and complex task on the ontologies connection.

[10] proposes an approach for student models alignment inside adaptive and distributed e-learning environments. Besides, the integration of two education environments was performed, the environments are Java-Probllets and QuizJET. However, such ontology based approach demands the insertion of new distributed e-Learning environments to be made manually which is very time consuming.

Moreover, the interoperability approaches described above are not scalable and not automatic. For this reason, every time a need to interoperate with a new environment, it is necessary to specify, implement, and configure the interaction. Aiming to explore the limitations of the works described above, an architecture for providing interoperability through educational environments is proposed in this paper.

**CONCLUSIONS**

In this paper, an architecture supporting interoperability between adaptive web-based educational system (AWBES) is proposed. The idea was to provide a computational architecture to provide interoperability between AWBES, giving support to communicate several universities in different places and with different courses. Additionally, several challenges to provide such facilities had to be dealt such as development complexity, different ontologies, semantics, and others. The approach used is based on semantic web technologies including ontologies, semantic web services, and ontology alignment.

As a future work, new case studies will be developed as well as new semantic web services to improve the quality of the discovering process. Furthermore, new case studies with different environments (e.g. TM4L) will be developed.

**REFERENCES**


