An approach to interoperability of ontology-based educational repositories

Darina Dicheva*
Department of Computer Science, Winston-Salem State University, Martin Luther King Jr. Drive, Winston Salem, NC 27110, USA
E-mail: dichevad@wssu.edu
*Corresponding author

Lora Aroyo
Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 315, 5600 MD Eindhoven, The Netherlands
E-mail: l.m.aroyo@tue.nl

Abstract: Web-based educational repositories aim at providing the learner with immediate, online access to a broad range of structured information and contextualised help to support efficient educational task performance. A significant effort is currently focused on bringing such systems to work together to provide better support within the context of web-based education. Our goal is to approach the problem from a rather practical perspective and propose a framework for supporting communication between existing ontology-based educational repositories to better utilise their resources. The framework allows two independent systems to share and interchange information solely through ontology-based communication without direct access to each other data stores. As a basis of the framework we define a communication ontology and propose a Concept-Based EIS Interaction Protocol (CBEIS IP) built over SAAJ-enabled Simple Object Access Protocol (SOAP) transport layer. We show how through communication protocols, we can realise a general modular architecture comprising components that can be shared and interchanged.

Keywords: interoperability of educational information systems; web-based education; ontologies; metadata; ontology-based educational systems.


Biographical notes: Darina Dicheva is the Paul Fulton/Delta Sigma Theta Distinguished Professor of Computer Science from the Winston-Salem State University, NC, USA. Her research interests include application of AI methods and tools in education, knowledge networks and management, Semantic Web, adaptive information retrieval and filtering and user modelling. She is Associate Editor of the International Journal of Continuing Engineering Education and Lifelong Learning (IJCEELL) and of the International Journal...
1 Introduction

With the growing presence of web-based education the role of educational information systems, including educational repositories, becomes rather central in achieving efficiency, effectiveness and satisfaction of both students and instructors. The main goal of these systems is to support more efficient performance of various educational objectives and tasks by providing learners and instructors with immediate, online access to a broad range of structured information as well as with domain-related help in the context of their work. These systems often need to interoperate, collaborate and exchange content or reuse functionality, to support a richer set of educational functions and increase their effectiveness. Consecutively, considerable research efforts are currently focused on designing powerful frameworks and architectures to tackle the issues of systems integration and interoperability. Clearly, such frameworks are fundamental for the development of the web-based educational systems of the future but are we ready for them now? What is the current situation with regard to the development and use of educational information systems?

Within the class of web-based educational systems, a major role in various instructional contexts play the Educational Information Systems (EIS), also referred to as educational repositories. There are a variety of such systems differing in the extent to which they provide intelligent, task-centred information support for solving problems and performing learning tasks. On the one hand, we have the monolith Learning Management Systems (LMS), such as Blackboard and WebCT, which, on more or less superficial level, cover various teaching, learning and administrative activities and as a result provide web-enhanced courses. On the other hand, we see multiple examples of specialised and effective educational systems and content providers, which support only a single task/function within the entire educational process. Representatives of such systems are adaptive textbooks constructed with AHA! (De Bra et al., 2003), InterBook (Brusilovsky et al., 1998) and NetCoach (Weber et al., 2001), or adaptive courses within ELM-ART (Brusilovsky et al., 1996),
An approach to interoperability

PAT Online (Ritter, 1997) and AIMS (Aroyo and Dicheva, 2001). There are also more global but still highly specialised efforts, such as ARIADNE\textsuperscript{1} and EdNa\textsuperscript{2} courseware-reusability frameworks that provide repositories of reusable educational objects.

Brusilovsky (2004) introduces Adaptive Web-Based Educational Systems (AWBES) as systems using techniques from Intelligent Tutoring Systems (ITS) and Adaptive Hypermedia (AH) to support courseware authors by combining different systems with different purposes to provide for different aspects of the instructional process and achieve an integrated solution. He claims that the ‘university has a clear need in a single integrated system that can support all critical functions in one package’. While we basically agree with him, we believe that it is not feasible to expect reaching the one-integrated-system goal in a near future and our claim here is that instead of a single complete integrated system, a good standardised way is currently needed to allow different specialised component-based systems to talk to each other. Such an approach from one side proposes a solution for Brusilovsky’s concern that ‘modern AWBES are designed to be used as a whole, not component by component’ and from another fits nicely to his advocacy of ‘distributed component-based architectures for building adaptive systems’.

In the current efforts targeting integration of various educational systems and content providers, Devedzic (2003) proposes educational servers (INES), which are based on using standards, ontologies and pedagogical agents to support interaction between clients (authors and students) and servers that host educational content and services. He claims that the interaction in the future educational systems will be between learners and services through educational service directories. We believe that WBES interactions in the near to medium future will be between educational systems’ components – before achieving the large-scale service-based integration, we need to explore and exploit possible direct communication between systems/components.

Another service-oriented perspective on the integration is given by the Elena project (Simon et al., 2003), which defines a smart learning space of educational service providers based on the Edutella peer-to-peer framework for interoperability and resource exchange between heterogeneous educational applications and different types of learning resource repositories (Nejdl et al., 2002). In the same context, we also see specific efforts trying to fill in the gap between adaptive educational systems and dynamic learning repository networks, by proposing service-based architectures for personalised e-learning. An example is the Personal Learning Assistant (Dolog et al., 2004), which uses Semantic Web technologies for realising personalised learning support in distributed learning environments.

In this paper, we try to approach the problem of systems integration from a rather practical perspective by proposing a framework for supporting communication between ontology-based EIS aimed at utilising systems’ resources and components. Our approach employs some features of web services and agent-based frameworks, but is intended to be simpler. Actually, we propose two architectures for systems integration: a general one and a minimalist one. While the general architecture, by providing a powerful service-oriented support for efficient communication between component-based educational information systems, fits well in the ambitious effort to define conceptually the shared and interoperable Educational Semantic Web, the minimalist one presents an efficient, currently realistic solution for supporting shareability and exchangeability of system resources. Our implementation efforts are
directed to the minimalist architecture, since we believe that it will help bridging the gap between current situation and the promises of the Educational Semantic Web of the future.

This paper is organised as follows. After identifying the common characteristics of ontology-driven educational information systems (Section 2), we outline the current needs and requirements for them to interact and share knowledge and resources (Section 3). In Section 4, we depict a general service-oriented framework to support the interoperability of various ontology-based EIS. In Section 5, we propose a minimalist approach for implementing the communication between two existing ontology-based EIS by defining a communication ontology (Section 6) and proposing a Concept-Based EIS Interaction Protocol (CBEIS IP) built over SAAJ-enabled Simple Object Access Protocol (SOAP) transport layer (Section 7). We conclude with a short summary.

2 Ontology-driven EIS

The main goal of web-based EIS is to help the learners perform their learning tasks efficiently by providing them, on one hand, with immediate, online access to a broad range of structured information and on the other, with domain-related help in the context of their work. A number of concept-based EIS have been already developed (Aroyo and Dicheva, 2001; Brusilovsky et al., 1996,1998; Dicheva et al., 2004; Dolog et al., 2004; Weber et al., 2001). These applications typically include:

- ontology-based representation of the subject domain
- repository of learning resources (digital library)
- course (learning) task presentation
- adaptation and personalisation.

The fundamental feature of these systems is the subject domain conceptualisation. It supports not only efficient implementation of the required functionality but also standardisation: the conceptual structure can be built to represent a domain ontology, which provides an agreed vocabulary for a domain knowledge representation. Thus, the ontology specifies the concepts to be included and how they are interrelated.

The repository contains learning resources (objects) related to the subject domain concepts. We can think of the resources as being attached to the domain concepts they describe, clarify or use. One of the most prominent themes in the ontology research is the construction of reusable components. If the attached objects have also a standards-based representation as opposed to a proprietary representation, this will ensure that the application’s content is reusable, interchangeable and interoperable. Course/learning tasks, when represented, are typically described in terms of subject domain concepts and some instructional relationships (such as ‘prerequisite’, ‘uses’, etc) between the involved concepts. The domain concepts are also used as a basis for implementing systems’ adaptive behaviour. The latter involves constructing learner models in terms of domain concepts, performed tasks and user characteristics.

An EIS user is typically involved in exploring the subject domain ontology and searching the repository for information related to a specific task. Good examples of such systems are OntoAIMS3 (an ontology-based successor of the AIMS system) (Denaux et al., 2005a,b) and TM4L4 (Topic Maps for Learning) (Dicheva and Dichev, 2004;
An approach to interoperability

Dicheva et al., 2004), which we use as a basis of our discussion. OntoAIMS and TM4L both focus on providing contextual support that enables learners to identify information necessary for performing a specific task (e.g. course assignment). They can be used standalone (e.g. as an extension to a traditional or online distance course) or integrated in a larger electronic learning environment that allows the users to perform open learning tasks in a specific subject domain. Since both focus on efficient information provision and support for task-oriented problem solving, these systems are quite similar but they can be also seen as complementary in the way they support learning tasks. While OntoAIMS includes course representation and sequencing, TM4L is a kind of digital library, which does not include direct course representation.

3 Need for EIS communication

Integration and interoperability are very important for EIS systems. If interoperable, two systems can benefit of additional functionality supplied by the other and especially of sharing resources and common components, for example, user models. In our example of TM4L and OntoAIMS, TM4L can use OntoAIMS course sequencing model, graphical viewer and resource metadata, while in turn OntoAIMS can use TM4L external and internal resources, domain and resources merging capability, text and external search.

We start our discussion on EIS system communication with presenting two use case scenarios illustrating the communication between the two considered systems.

Scenario I: a learner uses OntoAIMS as a support tool in her coursework. When trying to solve a specific task, however, she is not satisfied by the informational support provided by OntoAIMS since it is not enough for her to achieve her learning goal. She announces this and OntoAIMS seeks external help (more relevant information). It sends a request to TM4L for learning resources on the topic at hand. TM4L responds to OntoAIMS by sending back the information that it has. OntoAIMS feeds this information back to the learner. Figure 1 illustrates this scenario by showing the internal organisation of the two systems and the interface, which combines the task-based search from OntoAIMS with the resources from TM4L.

Figure 1 AIMS – TM4L communication
As we mentioned already, ontology-driven EIS have concept-based representation of the specific subject domain and learning resources indexed by domain concepts. Consequently, such a system would understand requests for information expressed in terms of domain concepts and relationships between them. Concerning a specific concept, possible requests of OntoAIMS to TM4L would be:

- give me the direct parents (children) of concept X (i.e. the concepts in a direct 'is-a' relation with it)
- give me all parents (children) of X (i.e. all concepts on the same path as X)
- give me all relations in which concept X plays a role (i.e. concept X is involved)
- give me all concepts related (in a relation of any kind) to a given concept
- give me all available resources connected with this concept
- give me learning resources of kind Y (e.g. using the LOM standard) connected to concept X
- give me all available information related to concept X (everything listed above).

Scenario II: an author uses OntoAIMS to construct a course. When preparing it, in addition to the learning resources she creates, she would like to reuse some available resources relevant to the course topics. She announces this to OntoAIMS. OntoAIMS sends a request for information on the specific topic(s) to TM4L. TM4L responds to OntoAIMS providing the information. OntoAIMS feeds this information back to the author to be approved for storing in the OntoAIMS domain model or disregarded.

Possible requests in this scenario include:

- give me all available resources connected to a concept X
- give me learning resources of kind Y (e.g. using the LOM standard) connected to concept X
- give me all your available information related to concept X
- import your entire domain ontology.

The communication between the two systems can be realised either at a system level or at a component level (in the second case between components with either identical or different functionality, e.g. domain models, user models, etc.). The latter imposes additional requirements to the architecture of the involved systems – they should have a clear component-based architecture.

The main research questions related to implementing communication between the systems include:

1. Level of granularity of information exchange: what should be the information contained in one communication transaction? For example, in the second scenario, should the author be allowed to ask about importing the entire TM4L domain model in one transaction?
An approach to interoperability

2 Request semantics: what kind of questions the requesting system should be able to ask?

3 Request syntax: in what form the questions should be expressed?

4 Domain or user model awareness: should the requesting system send any indication about what it ‘knows’ or its user already ‘knows’, so that the responding system does not send information already known? If so, what kind of information and in what form?

We attempted to answer these questions at two levels – a general one and a minimalist one – providing guidelines to the design of two corresponding frameworks for ontology-based EIS communication support. While the general one leads to a powerful service-oriented framework to support efficient communication between component-based EIS, the minimalist framework is intended to provide an efficient current solution for supporting shareability and exchangeability of systems resources. We started with defining the general framework with the intention to further constrain it to the desired minimalist one.

4 General architecture for component-based EIS interoperability

If we consider the problem of efficient construction of EIS systems that complement (serve as advisors to) each other by sharing resources and components, the obvious answer is modularised building of such systems. This implies a component-based architecture that allows sharing knowledge (e.g. domain ontologies, learning resources, course models and user models) and components (e.g. user modeling, course sequencing, ontology visualisation and keyword search). The systems will typically have different domain (and other) models and the interoperability between them should include support for:

• mapping between systems’ domain representations
• sharing learning resources
• sharing learning sequencing components (educational context, results, etc.)
• sharing user models.

4.1 On implementing system communication

To define such a general architecture we address the main research questions formulated in Section 3. With regard to the grain size of exchanged information (Question 1), we consider two issues as important for the communication quality and efficiency between the systems:

• conciseness of the request and the reply
• completeness of the request and reply.

These factors suggest that a system request has to ask for precise information related to a particular user query (topic) and the reply is to be as specific and detailed as possible.
This prompts us to propose that a finer level of granularity of information exchange is more appropriate. In some cases, however, as in Scenario II, we might require that the framework supports importing or merging of the complete domain models/ontologies (e.g. TM4L domain ontology merged with OntoAIMS domain ontology). We propose that this is realised through using appropriate communication support services and not through the ordinary request/response type of communication between the systems.

Concerning Question 2 in Section 3, to ‘understand’ each other both systems must ‘know’ the basic terms for structuring and using ontology-based learning resource repositories, such as ‘concept’, ‘relation’/‘association’, ‘role in a relation’, ‘resource’, etc. which makes a common ground for the semantic understanding. In other words, the different systems must know how to map their internal knowledge to the basic concepts of this common ground. This lightweight mapping process is a key aspect of the proposed approach, as opposed to very expensive reasoning processes. To specify precisely the basic terms of this common ground we propose a communication ontology, which we discuss in Section 6. This ontology defines what exactly communication input/output can be, that is, what requests will be allowed and what information will be returned.

Considering Question 3, we propose XML-based protocols, so that any application can ‘understand’ them. In relation to Question 4, we propose that the systems share a common user model, possibly through using a user modelling service. In that case the responding system will not need to ask the requesting system about the current user knowledge on the requested topic. A common (shared) user model is only feasible here since all EIS systems within the framework are assumed to have a concept-based representation of their subject domain.

4.2 The architecture

The proposed general architecture, reflecting the above discussion, includes (see Figure 2):

- standalone, component-based independent EIS using their private subject domain ontologies
- information brokerage bureau where all applications are registered
- services to support systems communication, for example, for ontology mapping
- communication ‘bridges’ between the systems supporting standardised transport mechanisms and a common interaction protocol.

The main purpose of the architecture is to support sharing and exchange of information between standalone EIS. This is achieved through a direct communication between the systems (or their components) and use of services included in the framework to facilitate systems’ communication. The services are intended to support different specific aspects of the communication. A significant class of services is ontology-related services since ontology alignment and translation are very important when different applications deal with different ontologies. This is exactly the case here as the different EIS in the framework have different domain ontologies.
A communication is an interaction between two software systems (agents) guided by an interaction protocol. The communication between the systems requires not only standardised transport mechanisms and communication languages, but also common content languages and semantics. The popular languages used to represent the content, embedded in messages in Agent Communication Languages (ACL), such as KIF,5 SL6 and Prolog, are ‘logic’ content languages, which aim at representing knowledge as logic expressions. For our purposes, we consider more attractive ‘information’ content languages, that is, languages that set rules to describe the types of information elements, since we do not need to represent information as logic expressions. For this reason, we consider XML very appropriate to represent the content embedded in messages in our proposed architecture. The vocabulary specified by an XML Schema can then correspond to the ontology of the messages, that is, content ontology (see Section 6.1). Note that XML can still be embedded within ACL messages to support better the communication between the different EIS (software agents).

To support the direct interaction between two applications we propose a communication ontology that defines the vocabulary of terms to be used in the messages at both layers: the message layer and the content layer (see Section 6). To interpret the requests and answers, standardised domain ontologies, UM ontologies, as well as upper-level ontologies, such as WordNet,7 SUMO,8 etc. can be used.

The information channels are ‘bridges’ between the systems realising the actual communication. They support standardised transport mechanisms and a common interaction protocol.

To ‘collaborate’ the applications should register within ‘Information Brokerage Bureau’. So, when a system needs help, it sends a request to this agency and it distributes the request to the other registered systems. In this regard, the applications can be seen as software agents and their communication can be supported by using, for example, the Knowledge Query and Communication Language (KQML) (Finin et al., 1994).
Our next step is to constrain the proposed general architecture by considering only two communicating systems that ‘know’ and ‘trust’ each other. We believe that this will be in the near future the common case and the goal is to find the minimal configuration that could support communication and allow sharing of knowledge and resources between such systems.

5 Minimal requirements and architecture for ontology-based EIS communication

Since our present goal is to support communication between already developed CBEIS, each system is assumed to be a standalone application and is not required to have a particular architecture or to ‘adapt’ to the other system in the framework. We make two important presumptions and use them as a basis of our constraints to the general architecture:

- The two systems know and are committed to communicating with each other. This implies that the systems will communicate directly and there is no need of Information Brokerage Bureau for registering the applications. Note that one system can communicate with more than one other system in such a direct mode.
- Concerning the services related to ontology mapping, an important presumption for our simplified framework is that the domain ontologies are created and used within one professional community, which facilitates a lot the task since an existence of an agreed upon understanding of the basic terminology used in this community is expected that favours the sharing of knowledge. Indeed, the goal of the EIS systems that we consider is to support learning in a specific course (discipline), for example, a Database course. The community of users includes potential instructors (authors) and learners. Since the authors are knowledgeable in the specific subject domain, in this case databases, it can be assumed that in defining the domain ontology they will use terms (concepts) that are accepted and agreed upon in that community. This will remove the necessity of alignment and translation of domain ontologies. That is why ontology-related services are not included in the minimalist architecture. Note that it will be useful though to include a service for merging ontologies. Currently, we delegate this task directly to the applications.

Table 1 compares the components in the general and minimalist architectures.

<table>
<thead>
<tr>
<th>Components</th>
<th>General architecture</th>
<th>Minimalist architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone EIS based on domain ontologies</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Information Brokerage Bureau</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Communication-supporting services</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Communication bridges</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
As given in Table 1, the proposed minimalist architecture includes only the two communicating systems and a communication bridge between them (see Figure 3). Since in this minimalist architecture there are only two ‘committed’ communicating systems, there is no real need of agent communication management.

**Figure 3** Minimalist integration framework for AIMS and TM4L

We define a common interaction protocol for the communication between Concept-Based EIS (CBEIS IP) built over a SAAJ-enabled SOAP transport layer. In the next sections, we first propose a communication ontology for the minimalist framework and then discuss implementation details of the proposed architecture, including the proposed interaction protocol.

### 6 Communication ontology

For the communicating applications to understand each other we need an ontology to provide a basis for sharing a precise meaning of the terms exchanged during communication. An ontology denotes a representation vocabulary of a specific domain and more precisely the conceptualisations that the terms in the vocabulary are intended to capture. In our case the domain of consideration is the communication between two CBEIS and includes two parts, corresponding to both layers of an interaction between two communicating systems, the message layer and the content layer. Consequently, we propose a Communication Ontology (CO) that consists of (see Figure 4):

- communication content ontology (corresponding to the content layer) that describes the content (knowledge) that can be exchanged by the systems
- interaction protocol ontology (corresponding to the message layer) that specifies interaction communicative act types.
6.1 Content ontology

The content ontology defines the terms (concepts) needed to exchange messages, that is, gives the meaning of the symbols included in the content expression. When two CBEIS exchange data, the message content will typically include two types of terms (concepts): terms belonging to the domain ontology of the sender (the application sending the message) and terms categorising domain term(s). The latter belong to the general information model of the CBEIS. For example, the sender can send a request for information of the kind ‘Send me all relationships in which you believe concept ‘ER-model’ is involved’. In this message ‘ER-model’ is a term (concept) from the subject domain of the requesting application (databases), while ‘relationship’ and ‘concept’ are terms belonging to the information model of the CBEIS.

Thus, in our framework, the content ontology consists of two parts: the Domain Ontologies (DO) of the involved EIS and a domain-independent ontology defining the concept-based information model of EIS (EISO). The latter includes the basic terms describing the information model of concept-based EIS, such as concept, concept name, relationship type, relationship role, etc. Figure 4 presents an excerpt from this ontology. In the proposed framework, each application uses the common EISO ontology and its own domain ontology. For this reason we have depicted domain ontologies separately from EIS ontologies in Figure 2.
6.2 Interaction protocol ontology

The Interaction Protocol (IP) ontology defines the terms related to message types, reasons and preconditions. While the communication content ontology is generally independent of the framework’s functionality, the IP ontology has to reflect its functionality (e.g. whether it supports agent communication).

Messages represent communicative acts denoting the actions related to communication. In general, communicative acts (performatives) include queries, responses, informational, capability definition, generative and networking (Finin et al., 1994). Since the two applications in our minimalist architecture are ‘committed’ to collaborate, the communication between them is very simple and does not require either the typical variety of message types, for example, types such as ‘agree’, ‘accept’, ‘cancel’, ‘propagate’ and ‘refuse’ or defining message preconditions and reasons. Thus, in our case the IP ontology (Figure 5) includes the following message types.

Figure 5 An excerpt from the IP ontology

6.2.1 Status
- **Failure**: informing that an action was attempted but the attempt failed.
- **Not understood**: message or DO term.

6.2.2 Query
- **Query-know**: asking whether the receiver knows about an object corresponding to an EISO term/category (e.g. specific concept, relationship, etc.)
- **Query-confirm**: asking whether a proposition is true
- **Query-object**: asking for an object or all objects of specific category in the EISO ontology (e.g. concept, relationship, etc.)
- **Query-if-object**: asking for objects as in ‘query-object’ but in case a specified proposition is true.
6.2.3 Response

- **Response-know**: informing the receiver whether or not the sender knows about the specified object
- **Response-confirm**: confirming to the receiver that the specified in the query proposition is true or not
- **Response-object**: sending to the receiver the objects specified in the request
- **Response-if-object**: sending to the receiver the objects specified in the request only if the specified proposition is true.

7 Communication bridge

As a basis of the transportation mechanism in our framework we have chosen SOAP, which is a standard lightweight protocol for exchanging information in a decentralised, distributed environment. It complies with the WS-I Basic Profile 1.0 specifications and therefore supports interoperability across platforms, operating systems and programming languages. It actually permits an exchange of messages in XML format between physically distributed machines. More specifically, the communication bridge is based on using the SOAP with Attachments API for Java (SAAJ). The SAAJ API allows creating XML messages that conform to the SOAP 1.1 and WS-I Basic Profile 1.0 specifications. A SAAJ client is a standalone client. It sends point-to-point messages, that is, a message goes from the sender directly to its destination. Messages sent using the SAAJ API are request–response messages. They are sent over a SOAP connection, which sends a message (request) and then blocks until it receives the reply (response). A SOAP message is an XML document. It always has a required SOAP part, and it may also have one or more attachment parts (that can contain any kind of content). The SOAP part must have an envelop, which contains a SOAP body.

To realise the communication between two concept-based EIS, we propose an interaction protocol (IP), CBEIS IP, which provides the real semantics of the communication between them. Since the message content language in the framework is XML, we have defined a DTD for XML files representing the content of interaction messages that conform to this protocol. The DTD definition is based on the developed CO. An excerpt of the DTD document is given below:

```xml
<|ELEMENT message (queryMessage | responseMessage) | statusMessage>
  <|ELEMENT queryMessage (query-know | query-confirm | query-object | query-if-object)>
    <|ELEMENT query-know (commitOntoTerm, dmOntoTerm)>
    <|ELEMENT query-confirm (proposition)>
    <|ELEMENT query-if-object (proposition, query-object)>
    <|ELEMENT query-object (objectSpec, categorySpec)>
  <|ELEMENT proposition (relation, dmOntoTerm, dmOntoTerm)>
  <|ELEMENT categorySpec (commitOntoTerm)>
    <|ATTLIST categorySpec type (category | ALL )>
    <|ELEMENT objectSpec (relOperator, commitOntoTerm, dmOntoTerm)>
      <|ATTLIST objectSpec type (object | ALL )>
    <|ELEMENT relation (#PCDATA)>
    <|ELEMENT dmOntoTerm (#PCDATA)>
    <|ELEMENT commitOntoTerm (#PCDATA)>
```

---
An approach to interoperability

This DTD allows sending messages like the following:

- A request asking whether the recipient ‘knows’ the concept ‘relational model’:

```xml
<queryMessage>
  <query-know>
    <commOntoTerm> concept </commOntoTerm>
    hoverOntoTerm> relational model </hoverOntoTerm>
  </query-know>
</queryMessage>
```

- A message, containing a ‘yes’ response to the previous request:

```xml
<message>
  <response-know type = known/>
</message>
```

- A message, requesting the relationships in which concept ‘ER-model’ is involved:

```xml
<message>
  <query-object>
    <objectSpecification>
      <relationalOperator type = equal/>
      <commOntoTerm> ER-model </commOntoTerm>
    </objectSpecification>
    <categorySpecification type = category>
      <commOntoTerm> relationship </commOntoTerm>
    </categorySpecification>
  </query-object>
</message>
```

The CO interface modules in our architecture (see Figure 3) are responsible for translating the messages (requests and responses) from the native language of EIS (e.g. TM4L or OntoAIMS) into the language of the universal CBEIS Protocol and vice versa. We plan to develop an API for Java (EISPAJ), to be used by the CO interface for creating programmatically and interpreting XML files (representing the content of interaction messages) that conform to the CBEIS Protocol. The CO interface is built on the top of a SAAJ module and uses it to realise the CBEIS Protocol with SOAP messages (the CBEIS commands are embedded within the SOAP body).

Thus, in the proposed minimalist framework, two independent systems can share and interchange information solely through ontology-based communication without sharing (by direct access) data stores. This removes any constraints on the systems architecture as well as the necessity of developing a ‘wrapper’ system, that is, an environment that hosts the communicating systems. The only requirement for the
systems is to be furnished with a plug-in realising a CO interface that enables sending and receiving messages conforming to the proposed CBEIS IP (through a SAAJ client and a SAAJ servlet).

8 Conclusion

We believe that the time for implementing large-scale educational web service frameworks has not come yet. Thus, our efforts are focused on increasing the use and efficiency of present, that is, already developed or currently being developed systems, more specifically ontology-based educational information systems. We propose to complement the functionality of such systems by supporting them to ask external ‘known’ peer systems for information, possibly involving information-providing processing.

We approach the problems related to systems integration and communication at two levels: a general level, by proposing a powerful service-oriented framework to support efficient communication between component-based EIS and a minimalist one, which illustrates an efficient proof of concept for supporting shareability and exchangeability of systems resources, applicable in the context of the current educational advancement. We believe that the minimalist architecture will help filling the gap between the desired future educational semantic web and the current realistic situation. As part of the framework we have defined a communication ontology consisting of communication content ontology and IP ontology and have embedded the latter within the proposed CBEIS communication protocol. We have discussed the concrete realisation of the IP ontology within the minimalist architecture. This way, we have shown how two independent systems could share and interchange information solely through ontology-based communication without sharing data stores.

The proposed framework for supporting communication between applications will eliminate in many cases the need for exporting the entire domain or other application model to another application. Thus, this will be an alternative to interchanging and merging domain models. The advantage is in eliminating duplication of information, which is unlikely to be often used. In addition, if an application has a specific concept-based application model with no corresponding model in the other system (as is the case with the course model present in OntoAIMS but not in TM4L), a domain model import will not work and the proposed communication would be the only way for the second system (TM4L) to use information from the first one (OntoAIMS). This will also solve some problems related to information shareability and reusability for already developed applications that do not use standards-based information but rather their own internal representations.

Acknowledgements

This material is based upon the work supported partly by the PROLEARN EU Network of Excellence (WP1: Personalised Adaptive Learning) and by the US National Science Foundation under grant no. DUE-0333069 ‘NSDL: Towards Reusable and Shareable Courseware: Topic Maps-Based Digital Libraries’.
References


Notes