DEVELOPING AGENT-ORIENTED E-LEARNING SYSTEMS

Sabin-Corneliu Buraga

Faculty of Computer Science
"Al.I. Cuza" University of Iasi, Romania
busaco@infoiasi.ro
http://www.infoiasi.ro/~busaco/

Abstract: The paper proposes an agent-oriented extensible framework based on *Extensible Markup Language* (*XML*) family for building a hypermedia e-learning system available on the World-Wide Web. The paper is focused on the implementation solutions of an e-learning (tutoring) Web-based system by deploying mobile agents that can exchange information in a flexible way via XML-based documents (such as RDF assertions or/and SOAP messages).

Keywords: E-learning System, XML, Agent, Knowledge Representation

1. INTRODUCTION AND MOTIVATION

The World-Wide Web space has become an indispensable tool for all specialities' researchers. Not only the hot news, but many consistent or semi-consistent collections of scientific data, in particular disciplines (*e.g.*, Computer Science, Mathematics, Chemistry etc.) are now accessible on Internet. Once simply a means of accessing information stored across diverse platforms, the Web is now a widely used environment for communication and information exchange (Berners-Lee, 2002).

The Web has attracted a great deal of consideration as a medium for delivering distance courseware, in a synchronous and asynchronous manner. Institutions with long-standing contribution in distance education are integrating Web-based elements by providing miscellaneous complex e-learning solutions (Jalobeanu, 2001).

Although Web-based course materials have many advantages over usual textbooks and lecture notes, they have a number of general deficiencies (De Bra, 1996). Most important of these follow:

- The access to course materials is (relatively) slow;
- Courseware does not automatically adjust to the profile of individual students (regarding actual skills, preferences etc.);
- Course's interactivity must be programmed (using different server-side or client-side complicated solutions such as C/C++, Perl, Java, C#, JavaScript or other available programming languages);
- Features of Web processing (caching and client-side information hiding) in most cases obstruct the collection activity of student performance data.

A number of attempts have been made to (partially) resolve some of these problems, but solutions to one problem often obstruct solution of the remaining problems (Buraga, 2001b; Buraga, 2003).

Another issue is the *adaptive annotation* for Web. History-based adaptive annotation is familiar to all

Web users because any actual browser allows them to distinguish visited and unvisited nodes, showing these nodes in different colors. The e-learning system may propose more sophisticated methods of adaptive annotation that could be also very helpful for the Web users. All adaptive navigation support methods can be based on the models of hypertext theory (Trăuşan-Matu, 2000). One of the implemented techniques is *Annotea*, integrated within the experimental *Amaya* browser of the World-Wide Web Consortium (W3C, 2003).

An e-learning system must consider a *decentralized approach* in which overall course management is performed centrally, but course materials (hypertext documents, multimedia documents, technical manuals, scripts and other applications) are served up locally using various pieces of software that runs on the student's workstation. A co-operative e-learning application should make a minimal set of assumptions as possible about the system platform regarding different hardware requirements (system performance, screen resolution, multimedia capabilities, other auxiliary devices – such as video camera, mobile phone, infrared equipments – etc.) and software requirements (operating system, version, installed applications, required plug-ins etc.). Personal computers (especially mobile devices) tend to be unstable and a co-operative application should be tolerant against local breakdowns and should without difficulty allow restarting a system and rejoining an existing group of students.

Interactivity and intelligent tutoring capabilities (i.e. various help facilities) must be provided by client-side software as well (Pecheanu *et al.*, 2001). Therefore, enlarged usage of on-line courses by non-specialists has increased the need for a more successful and friendlier *user-interface experience*. Especially for remote students, it should be as easy as possible to set up and run a co-operative application. Different applications should provide consistent Web interfaces, at least for all aspects of setting-up, e-learning and broad asynchronous or/and synchronous collaboration (Buraga, 2002).

These sources of information and the complex interactions between professors and students obey with a *standard for interoperability*.

One solution of the presented problems is the use of extensible and adaptable uniform methodologies and platform-neutral languages intended for exchanging hypermedia information between the components of the Web-based e-learning applications.

The paper proposes a Web agent-based solution that can be used to design and to implement a tutoring system. The involved resources of the e-learning system can be straightforwardly described by *RDF* (*Resource Description Framework*) assertions (see section 3). To easily exchange information between different modules (agents), an XML-based communication protocol is projected (details in section 4).

2. XML META-LANGUAGE AND ITS APPLICATIONS

2.1 Short presentation

XML (*Extensible Markup Language*) is a well-known recommendation of the World Wide Web Consortium (W3C, 2003) for a meta-language to define markups (annotations) for content publishing on the WWW space and other areas.

The goal of XML is to provide different benefits not available in HTML, such as arbitrary extensions of a document's elements (tags) and their attributes, support for documents with complex structure, and validation of document structure with respect to an elective document-structure grammar, called a *DTD* (*Document Type Definition*). A DTD specifies what elements may occur and their order of occurrence and how the elements may nest in an XML document that conforms to this DTD. Also, instead of DTD, an alternative object-oriented method for validation of XML documents can be used: an *XML Schema* (W3C, 2003; Buraga, 2001a).

2.2 XML-based languages and applications

Since several years, XML has developed into a large family of standards integrating key technologies from three previously independent domains: documents, databases, and the Internet. Several examples follow (Buraga, 2001a):

- XHTML (Extended HyperText Markup Language – HTML in the terms of XML),
- MathML (Mathematics Markup Language),
- SMIL (Synchronized Multimedia Integration Language),
- RDF (Resource Description Framework),
- XUL (Extensible User-interface Language),
- WML (Wireless Markup Language).

Using the XML meta-language, the semantics and the structure of the data exchanged by different Web applications is preserved. One of the main XML's advantages is that the data can be structured as in an object-oriented database. The XML can be the optimal solution for data integration from multiple sources, too.

As XML is format-independent, there is possible to generate multiple – XHTML, SMIL, WML or XUL – outputs very easily by transforming XML documents via *XSL* (*Extensible Stylesheet Language*) rules (W3C, 2003). Similarly to the *CSS* (*Cascading Style Sheets*), the XSL documents are able to divide the

content from the representation. An XSL stylesheet file can define the formatting characteristics of XML documents on the Web by using platform-independent formatting objects (W3C, 2003).

Actually, the XML documents can be processed either in the Web browser (last generation browsers – Microsoft Internet Explorer, Netscape/Mozilla and Opera – have a good support for XML parsing) or on a Web server (*e.g.*, Apache or Internet Information Server).

On the server, the content can be easily stored into XML documents without the layout or even within *native XML databases* (such as Tamino, eXists, Apache Xindice, or Socrates XML) – for more details see *http://www.rpbourret.com/xmldbms*.

Content contributors (course authors, student annotators, different other persons) can write their content in specialized applications that suit their needs and produce XML files as output. These documents are then automatically stored in a standard relational database or in a native XML database and can be accessed instantly over the Internet. Then, using XML on the client/server-side, it is possible to automatically transform XML – via an XSL stylesheet – to XHTML for rendering data into the Web browser.

For querying purposes, different *XML query languages* can be used (*e.g.*, XPath, XQL, XQuery or XML-QL).

2.3 XML processing methods

To facilitate the XML document processing, the World-Wide Web Consortium proposes an object-oriented model: *DOM* (*Document Object Model*) divided into levels of platform- and programming language-independent functionalities. Actually, DOM – level 1 and DOM – level 2 are already standardized (W3C, 2003).

A flexible and convenient incrementally event-based XML processing technique is given by *SAX* (*Simple API for XML*), with miscellaneous implementations in popular programming languages like C, C++, Perl, PHP and Java (Buraga, 2001a). Notably, the programmer can use the *libxml* library – a freely available SAX implementation under Linux platforms, part of the GNOME (GNU Network Object Model Environment) project.

3. USING RDF TO DESCRIBE RESOURCES OF AN E-LEARNING SYSTEM

3.1. Short presentation of the RDF model

RDF (*Resource Description Framework*) is a standardized basis for processing metadata. RDF consists of a model for representing named properties and property values. RDF properties may be thought of as attributes of resources and in this sense correspond to traditional attribute-value pairs. Each property has a specific meaning, defines its permitted values, the type of resources it can specify, and its relationship with other properties (via a RDF Schema). RDF properties also represent relationships between resources and a RDF model can therefore resemble an entity-relationship diagram. In object-oriented design terminology, resources correspond to objects and properties correspond to instance variables (W3C, 2003; Buraga, 2001a).

The concrete RDF syntax uses XML constructs. The RDF documents can be validated and processed by various tools, such as SiRPAC (Simple RDF Parser and Compiler) – a freely available Java servlet based on the Megginson's SAX processor (W3C, 2003).

3.2. Modeling information with RDF

The RDF facilities are decisive in modeling of a knowledge based e-learning system (Ştefănescu *et al.*, 2001; Pecheanu *et al.*, 2001), for the representation of several information such as:

- instructional content divided into different modules for multiple use and re-use;
- abstract pedagogical entities;
- tutor and student profiles;
- domain, pedagogical, and student knowledge bases (facts and rules) used by the system's inference engines.

Also, the RDF constructs may be used successfully to specify the relationship between various components of the e-learning system and between the participants of the on-line courses. RDF metadata can also describe client's session information (Buraga, 2001b; Buraga, 2003):

- host and user profiles (tutors, students);
- user profiles of participants of previous and current sessions;
- session profiles (*e.g.*, timing, available resources, navigation history, etc.);
- private user applications and resources (viewers, plug-ins, content editors, etc.).

In the following example, the list of the on-line courses accessed by a particular student is expressed by several RDF constructs:

```
<rdf:RDF>
  <rdf:Bag ID="courses">
    <!-- Web Technologies course -->
    <rdf:li
        rdf:resource=
            "http://www.infoiasi.ro/courses/web" />
    <!-- Operating Systems course -->
    <rdf:li
```

```
        rdf:resource=
            "http://www.cs.pub.ro/teach/os" />
    <!-- Java Programming course -->
    <rdf:li
        rdf:resource=
            "http://www.infoiasi.ro/java_prog" />
    <!-- other courses… -->
  </rdf:Bag>
  <rdf:Description rdf:about="#courses">
    <t:User t:id="74" t:type="student">
    <rdf:Description
        rdf :about=
          "http://students.infoiasi.ro/~stud">
        <t:Name> ... </t:Name>
        <t:Year> 4 </t:Year>
        <--other useful information -->
    </rdf:Description>
  </rdf:Description>
</rdf:RDF>
```

The namespace prefix *t* refers to a specific namespace prefix chosen by the author of the RDF expression and defined in an XML namespace declaration such as *xmlns:t="http://some.host.com/xml-schema"*.

The *rdf* namespace is defined by the World-Wide Web Consortium to be specified in every RDF statement. The XML namespaces are used to avoid parsing conflicts for identical elements or attributes names included in the same XML document (W3C, 2003; Buraga, 2001a).

Also, RDF can be used to express a high-level platform-independent model for accessing and discovering hypermedia resources of an Internet-based distributed system (for more details see Buraga, 2001c; Buraga and Ciobanu, 2002; Alboaie et al., 2003).

4. IMPLEMENTATION PROPOSAL

The rest of the paper proposes an agent-oriented approach to effectively implement an e-learning Web-based system – an intelligent tutoring system. The intelligent tutoring system has the structure presented in Pecheanu et al., 2001.

4.1 Internal architecture of the tutoring system

The intelligent tutoring system is composed of four major components:

- *Domain Module* – contains representations of the knowledge intended to be communicated to the student, including descriptions of diverse concepts and skills of an expert in a particular domain;
- *Student Module* – encloses information about the student's understanding of the domain knowledge and is responsible to and dynamically builds a model of how students learn and utilize the diagnostic tools contained within the Pedagogical Module;
- *Pedagogical Module* – contains rules or other decision making instruments that allow it to evaluate how well the student's understanding of the subject domain matches actual knowledge structure;
- *Interface Module* – presents a uniform Web-based environment within which instruction, diagnosis, remediation, and user driven learning may take place.

Information processed by each module can be consistently stored by XML documents.

The inference rules stored within the pedagogical module can be expressed by RDF constructs. First, a domain conceptual structure must be designed and a conceptual map of the domain being taught must be built (Pecheanu et al., 2001). One of the possible solutions is to implement the hierarchy of concepts of the domain knowledge. This domain knowledge can be represented by semantic network architectures, consisting of conceptual nodes and relations between nodes. The semantic network is intended to be denoted by the RDF assertions generated by a special software tool.

The interface module will be able to present – via XSL stylesheets – information on Web by using HTML, SMIL, XUL or WML constructs from the same primary data marked-up in XML, in order to give users the desired Web adaptive interface (Buraga, 2002).

4.2 E-learning system's functionality

To implement the functionality of the system, an agent-based approach is proposed. The fundamental resources that computers expose to the software components (i.e. operating system, applications) or users are the following:

- computing capabilities,
- (volatile or non-volatile) memory,
- local and remote data (documents),
- local and remote metadata (different descriptions about several properties of the system's resources: content, structure, layout or interface, dynamics, security issues, etc.).

Also, the WWW space can be viewed as a distributed hypermedia system that uses Internet technologies

(TCP/IP protocol family), a global system of heterogeneous networked computers. Advances in networking and Web/Internet technology are leading to a network-centric computing model, and the Web and Internet itself, of course, are evolving into the infrastructure for global network computing. By populating this infrastructure with object-based components and combining them in various ways, the development and deployment of interoperable distributed object systems can be enabled on the Web (Alboaie *et al.*, 2002).

Using software mobile agents A mobile object, usually called an *agent* when operating on behalf of a user, is a downloadable, executable object that can independently move (code and state) at its will – the mobile agent is not bound to the system in which it began the code execution and can travel from one node (host) on a network to another (Bradshow, 1997).

Mobile agents can present the following attributes:

- *reactive* – the capability to respond to changes within agent environment;
- *autonomous* – the mobile agent is able to exercise control over its own actions (decisions);
- *goal-oriented* – the agents have an intended itinerary, they do not simply act in response to the environment;
- *communicative* – the capacity to communicate with other agents or processes, by exchanging information (knowledge);
- *mobile* – the mobile agents can transfer themselves from one machine (Internet host) to another.

Mobile agents can be used to access and administer information that is distributed over large areas and they successfully can implement various modules (such as the interface module or the evaluation module) of the considered e-learning system. The main benefit is that the software components can be integrated into a coherent and reliable software system – e.g. a multi-agent system – in which they work together to better meet the needs of the entire application (utilizing autonomy, responsiveness, pro-activeness and social ability). These features are very significant in the case of an intelligent tutoring system, too.

Domain module can be implemented as an intelligent agent that will process the XML/RDF statements. Also, a pedagogical agent needs to be designed and implemented (Ştefănescu *et al.*, 2001), in order to facilitate the process of acquisition and modeling of knowledge. This problem is still a key phase in the whole development process of the system. A special type of an intelligent agent could take all the pedagogical decisions.

Communication between modules One of the major problems is to use a platform- and language-independent protocol (Alboaie et al., 2003) for exchanging information between the agents of the e-learning system or between the agents and other entities (i.e. final users, processes, modules).

A Web-based e-learning system engages multiple and various sources of information. Also, the interactions between professors and students comply with a standard for interoperability.

One solution is to consider an XML-based protocol – *SOAP* (*Simple Object Access Protocol*). SOAP is a simple lightweight protocol that can be used for structured and strong-type information exchange in a decentralized, distributed environment (W3C, 2003).

The SOAP protocol consists of three main parts:

- An *envelope* that depicts the contents of the message and how to use it;
- A set of *rules for serializing data* exchanged between Web applications;
- A *platform-neutral manner to represent remote procedure calls*, that is the way in which queries and the resulting responses to the procedure are represented.

Similar to object distribution models (e.g., CORBA IIOP and DCOM), SOAP can call methods, services, components, and objects on remote servers. However, unlike these protocols, which use binary formats for the calls, SOAP exploits text format (Unicode), with the help of XML, to structure the nature of the data exchanges.

SOAP can generally operate with several protocols, such as *FTP* (*File Transfer Protocol*) or *SMTP* (*Simple Mail Transfer Protocol*), but it is particularly well-suited for the HTTP (*HyperText Transfer Protocol*).

Because RDF is based on XML, the RDF messages can be used over SOAP in order to exchange knowledge between the agents of the e-learning system. Using this approach, the multi-agent e-learning system could manage the semantic information about the involved resources.

5. CONCLUSIONS

Multiple sources of information and the interactions within an e-learning system available on Web comply with a standard for interoperability and an effective flexible implementation.

One solution is to use the extensible and adaptable standardized methodologies and platform-independent annotation languages for exchanging hypermedia information between the components of the Web-based e-learning system.

The paper investigates an agent-based approach that can be used to design and implement an intelligent

tutoring system. Also, to represent the knowledge base of the system a RDF solution is proposed. The RDF assertions can be used to model the relationships established between various components of the courseware system and between the participants (e.g., tutors, students) of the on-line courses.

The functionality of the system is given by a Web agent-oriented approach. For this, a flexible mobile agent architecture is proposed as a solution for implementation. To easily exchange information between different modules (agents), the SOAP communication protocol over HTTP is used. The SOAP messages could encapsulate the semantic content stored into RDF statements.

Of course, further theoretical and practical studies will give a more detailed internal structure of the proposed agent-based system and will give the possibility to design and to implement a completely functional e-learning system.

REFERENCES

Alboaie, S., **Buraga, S.**, and **Alboaie, L.** (2002). An XML-based Object-Oriented Infrastructure for Developing Software Agents. In: *Scientific Annals of the "Al.I.Cuza" University of Iaşi* – Computer Science Section, Tome XII, "Al.I.Cuza" University Press House, Iaşi.

Alboaie, S., **Buraga, S.**, and **Alboaie, L.** (2003). An XML-based Serialization of Information Exchanged by Software Agents – accepted for presentation at *Agent-Based Computing 2003* – special session in conjunction with the 7th World Multiconference on Systemics, Cybernetics and Informatics – SCI 2003, Orlando.

Berners-Lee, T. (2002). *The World-Wide Web – Past, Present and Future*. Japan Prize Speech.

Bradshow, J. (1997). *Software Agents*, AAAI Press.

Buraga, S. (2001a). *Tehnologii Web* (in Romanian), Matrix Rom, Bucharest.

Buraga, S. (2001b). An Extensible Framework for Building Interactive Courses on Web. In: *The 5th International Symposium on Economic Informatics – IE 2001 Proceedings*, Bucharest.

Buraga, S. (2001c). A Model for Accessing Resources of the Distributed File Systems. In: *"Advanced Environments, Tools and Applications for Cluster Computing" Proceedings of the NATO ARW Mangalia*, Romania, 1-6 September 2001, D.Grigoraş *et al*. (eds.), *Lecture Notes in Computer Science – LNCS 2326*, Springer Verlag, Berlin.

Buraga, S. (2002). *Proiectarea siturilor Web* (in Romanian). Polirom, Iaşi.

Buraga, S., **Ciobanu, G.** (2002). A RDF-based Model for Expressing Spatio-Temporal Relations Between Web Sites. In: *The 3rd International Conference on Web Information Systems Engineering – WISE 2002 Proceedings,* IEEE Press, New-York.

Buraga, S. (2003). An XML-Based Agent Oriented E-learning System. In: *The Annals of "Dunărea de Jos" University of Galaţi* (to appear).

De Bra, P. (1996). Teaching hypertext and hypermedia through the Web. In: *WebNet96 – World Conference of the Web Society Proceedings,* AACE, 1996.

Jalobeanu, M. (2001). *WWW în învăţământ* (in Romanian). Casa Corpului Didactic, Cluj.

Pecheanu, E., **Ştefănescu, D.**, **Buraga, S.**, and **Istrate, A.** (2001). Integrating Hypermedia Objects in an Intelligent Tutoring System. In: *The Annals of "Dunărea de Jos" University of Galaţi*, fascicle III, pp.92-99.

Ştefănescu, D., **Pecheanu, E.**, **Buraga, S.** (2001). Pedagogical Agents in Intelligent Tutoring Systems. In: *The 11th International Symposium on Modeling, Simulation and System's Identification – SIMSIS 2001 Proceedings,* Galati.

Trăuşan-Matu, Ş. (2000). *Interfaţarea evoluată om-calculator* (in Romanian), Matrix Rom, Bucharest.

World-Wide Web Consortium (2003). *Technical Reports*. Boston: http://www.w3.org/