A SCORM Thin Client Architecture for E-learning Systems Based on Web Services

Giovanni Casella, Dipartimento di Matematica e Informatica, Italy

Gennaro Costagliola, Dipartimento di Matematica e Informatica, Italy

Filomena Ferrucci, Dipartimento di Matematica e Informatica, Italy

Giuseppe Polese, Dipartimento di Matematica e Informatica, Italy

Giuseppe Scanniello, Dipartimento di Matematica e Informatica, Italy

ABSTRACT

In this paper we propose an architecture of e-learning systems characterized by the use of Web Services and a suitable Middleware component. These technical infrastructures allow us to extend the system with new services as well as to integrate and reuse heterogeneous software e-learning components. Moreover, they let us better support the "anytime and anywhere" learning paradigm. As a matter of fact, the proposal provides an implementation of the Run-Time Environment suggested in the Sharable Content Object Reference Model (SCORM) to trace learning processes, which is also suitable for mobile learning.

Keywords: e-learning system; run-time environment; SCORM; traceability; Web services

INTRODUCTION

During the last decade a surprising evolution of electronics, computer systems, and information technologies, together with the worldwide accessibility to the Internet, have made available an incredible set of applications. This trend has significantly conditioned the emergence and evolution of new academic and industrial opportunities. In particular, the possibility to easily reach an extremely large number of users with significantly low costs has motivated the development of an increasing number

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

of Web applications for educational purposes. As a result, e-learning has achieved a worldwide acceptance in several domains, such as universities, secondary schools, companies, and public institutions. Moreover, in the last years the popular emphasis on "anytime and anywhere" has determined the need of a new kind of e-learning, named m-learning (mobile learning), meant to take advantages from mobile computing devices (mobile laptops, PDAs, mobile phones, etc.), which are becoming more and more pervasive.

Currently, several relevant challenges characterize the e-learning research, such as interoperability, reuse, and extensibility of software components. In order to understand that, it is worth noting that the success of e-learning has caused the proliferation of several kinds of e-learning-related software applications, from content delivery to collaborative environments. As a result, many companies are entering the learning market, developing new products, and often combining existing components opportunely configured. Thus, it is important to understand how these systems relate to each other and how they fit into a complete e-learning environment. Moreover, the issue of component interoperability is especially relevant to promote reuse of services and develop e-learning systems starting from heterogeneous components.

In order to address the above issues we propose a software architecture of e-learning systems based on Web Services (Bosworth, 2001;Roy & Ramanujan, 2001) and a suitable Middleware component (McKinley, Malenfant, & Arango, 1999). Web Services technology provides a common infrastructure to integrate heterogeneous software components, thus enhancing interoperability between different components and component reuse, whereas the extendibility feature of e-learning systems is ensured by the use of a specific Middleware component.

Further appealing features characterize the proposed architecture. First of all, special focus has been put on e-learning standards that have been recommended in the last years. In particular, the proposal provides a suitable implementation of the Run-Time Environment (RTE) (SCORM RTE, 2004) suggested in the Sharable Content Object Reference Model (ADL SCORM, 2004) by the Advanced Distributed Learning (ADL) consortium (ADL, 2004). The SCORM RTE addresses an important issue, namely the traceability of student learning processes. In particular, to enable the traceability of student's activities it defines the format of messages exchanged between the content and the e-learning system. It is worth noting that the effectiveness of the e-learning paradigm can be heavily affected by the quality of the traceability process. Indeed, the collected information can be exploited to personalize knowledge contents, thus improving learning performances and welfare of the students. Moreover, to realize an accurate evaluation of each student,

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

instructors can benefit from some information on course enjoying, such as the time spent to consume a Learning Object or to complete a quiz.

Another notable characteristic of the proposal is that it is also suitable for m-learning. In order to understand this aspect, it is worth noting that when designing e-learning applications that should be also apt for m-learning, the limitation of resources of mobile devices has to be considered. In particular, software architectures that do not require specific resources on the client side should be conceived. To this aim, we propose a thin client Web architecture, thus ensuring that the delivered e-learning courses are suitable for any kind of client devices and also for mobile devices equipped with only an HTML browser and a wireless connection.

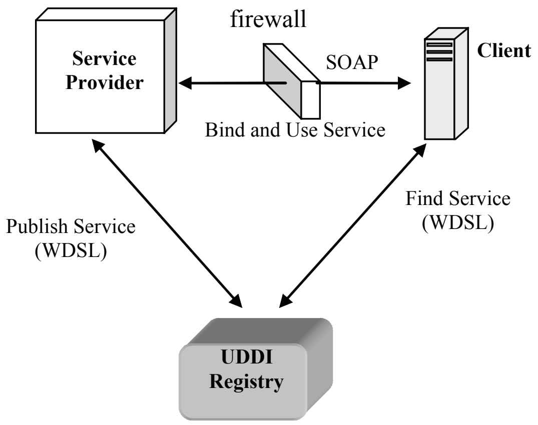
The remainder of the paper is organized as follows. The key features of Web Services and the main concepts related to the SCORM standard are reported in the first two sections, respectively. Then, we describe the proposed SCORM compliant software architecture. Related work and final remarks conclude the paper.

WEB SERVICES

Web Services are modular, self-describing applications universally accessible in a standardized and platform independent way (Bosworth, 2001;Roy & Ramanujan, 2001). These applications communicate by standardized XML messages. Web Services are based upon three technologies: Web Services Description Language (WSDL), Universal Description Discovery and Integration (UDDI), and the Simple Object Access Protocol (SOAP).

In a typical Web Services scenario (see Figure 1) the service provider

Figure 1. The scenario of using Web Services



Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

publishes all the information about a service in a UDDI registry. This registry stores descriptions and metadata for each registered service using the WSDL language, a common XML format. Once the service is deployed, each client application can ask for it from the registry. This registry sends to the client information to bind and use the required service. The client application sends a request to a service at a given URL using the SOAP protocol over HTTP. The service receives the request, processes it, and returns a response. It is worth noting that Web Services can be also directly used by accessing to the provider.

The most known advantages provided by the use of Web Services are described in the following:

- *Interoperability* – they improve the interoperability among different systems and operate on the "system boundaries" for communicating in a standardized way. They are described using a standard language to let external applications understand and invoke them.
- *Easy to use* – by their use the core business logic of systems is easily offered over the Web without knowledge of their target system's environment. Client applications can use one or more Web Services to combine their results to accomplish a task. Moreover, developers use Web Services without changing their own component object model, architecture, and implementation strategy.
- Reusability – Web Services are software components that are reusable and extendable. They can be consumable by both humans and computers, for example, through a desktop application or API.
- Ubiquity – Web Services respect existing security systems and as a consequence they are accessible from anywhere.

For the above reasons in this paper we propose an architecture of e-learning systems based upon Web Services to integrate different software components. Moreover, this allows us not only to easily extend the system with new features but also to improve interoperability among different e-learning systems.

E-LEARNING STANDARDS

Components in an e-learning system can be distributed on different machines and provided by different suppliers. To support this distribution, ensure interoperability across systems, and guarantee reusability of instructional contents, several standardization initiatives have been launched.

This section describes the main concepts related to e-learning standards and focuses on the SCORM RTE proposal (SCORM Sample, 2004). In order to better understand these notions, we start by briefly outlining the main components of an e-learning system and stressing the importance of student's activity traceability.

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

The Main Components of an E-learning System

Usually e-learning systems provide a variety of capabilities aimed at supporting their users with different purposes. Nevertheless, a general agreement seems to exist regarding the core functionality of modern e-learning systems. They are usually grouped in two components: the Learning Management System (LMS) and the Learning Content Management System (LCMS).

Teachers and instructional designers exploit an LCMS for creating and modifying activities composing e-learning courses. In particular, the LCMS component supports authors in the creation and management of digital instructional contents, which are stored in a repository (Apple, Nygren, Williams, & Litynski,, 2002; Campbell & Mahling, 1998; Douglas, 2001). Thus, the tool provides functionality to define tests (Safoutin et al., 2000), didactic contents (Designer's Edge, 2003), collaborative environments (Chang, Chen, Liu, & Ou, 1996; Cuthbert, 1999), and course editing (Goodyear, 1997; Kasowitz, 1997; Vrasidas, 2002).

In contrast, an LMS provides features to manage all the activities surrounding learning deployed via Web, such as user authentication, course deployment, and management of collaborative synchronous and asynchronous environments. Moreover, an LMS interacts with the run time environment, which is addressed by learners and traces the student learning process while the e-learning course is enjoyed. The information collected during the traceability is processed and organized to obtain statistics on learners' behaviours and to improve the overall quality of the learning process. For instance, traceability information can be used to personalize knowledge contents, thus improving students' welfare and their learning performances. Moreover, during assessment tests, information about each given question and the time spent to answer it can be usefully exploited by the instructor to get a deep evaluation of each learner.

The SCORM Standard

In general, the purpose of e-learning interoperability standards is to provide standardized data structures and communication protocols for e-learning content objects and systems. The use of these standards in e-learning products enables on one side instructional designers to reuse knowledge contents, and on the other side organizations to purchase system components from multiple vendors with confidence that they will effectively work together (Rosenberg, 2001; Smythe, Shepherd, Brewer, & Lay, 2002; SUN, 2002).

Several international organizations, such as IEEE's Learning Technology Standards Committee (IEEE LTSC, 2004), IMS Global Learning Consortium (IMS, 2004), US Department of Defense, Aviation Industry CBT Committee (AICC, 2004), Advanced Distributed Learning consortium (ADL, 2004), ARIADNE (ARIADNE, 2004), and PROMETEUS (PROMETEUS,

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

2004), are contributing to this standardization.

In particular, the Advanced Distributed Learning (ADL) initiative is a collaborative effort among government, industry, academia, and organizations to establish an environment that permits the interoperability of learning tools and course contents.

Starting from these collaborations ADL produced the SCORM standard (ADL SCORM, 2004), which specifies consistent implementations that can be used across the e-learning community. SCORM is composed by some specifications that enable interoperability, accessibility, and reusability of Web-based learning contents.

In the following we describe the main features of the SCORM standard, which are relevant for our proposal, namely Metadata, Content Packaging, Content Sequencing and Navigation, and Content Communication.

To enable a common nomenclature for the indexing, storage, discovery, retrieval, and exchange of e-learning resources, SCORM provides Metadata to be applied to knowledge contents at several granularity levels (IEEE LTSC, 2004). Improving the completeness, carefulness, and flexibility of metadata, e-learning resources are better described and can be also reused in several learning contexts with different aims.

Responsibilities and requirements for Content Packaging are defined in the SCORM Content Aggregation Model and are characterized by Content Aggregations, Activities, Sharable Content Object (SCO[1]), and Assets. As a consequence, a content package is described by a manifest, which bundles content objects with content organization. A SCO can be meant as a course, a lesson, a module, or simply a collection of related content objects (Asset). It is worth noting that only SCO didactic resources can be traceable.

The SCORM Sequencing and Navigation book describes how the contents are sequenced through a set of learner or system navigation events (SCORM SN, 2004). The branching and flow of the contents are described by a predefined set of activities, which are typically specified by the teacher at the design time. ADL describes the content sequencing and navigation in terms of an activity tree, based on the results of learners' interactions with content objects and a designed sequencing strategy. LMS using that data structure is able to manage learning processes as the suitable sequence of content objects to deliver.

Content Communication defines standard methods to establish and exchange information on the learning process to implement the learner traceability. ADL describes the communication between content (in particular, a SCO element) and LMS in the RTE book (SCORM RTE, 2004). Indeed, SCORM standardizes a common way to launch contents, to enable the communication with the LMS, and predefined data elements that are exchanged between an LMS and content during its enjoying. Thus, in the SCORM RTE three aspects

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

are specified, namely the Launch, the Application Program Interface (API), and the Data Model.

The Launch process defines a common way for LMS to start content objects. Moreover, it specifies procedures and responsibilities for establishing the communication between the launched content object and the LMS.

Application Program Interface is a set of predefined functionalities that should be considered by both LMS and LCMS vendors to enable communication between them. These functionalities complete the launch process by establishing a "handshake" between the SCO and the LMS that launched it and breaking that "handshake" once the SCO is no longer needed. Assessment process, content enjoying, and errors that occur during these learning processes are traced by using SCORM content, which set and get data on the LMS.

In the Data Model vocabulary we can find the definitions of the actions "to get and set data from and to an LMS" when SCORM API functions are called. For instance, when passing a test score from a learner, a SCO would use the SCORM Data Model element known as cmi.score.scaled to inform the LMS on the learner's performance.

THE SYSTEM ARCHITECTURE

In this section we describe the main features of the software architecture that we propose. In order to better understand this architecture, we first report the implementation of the SCORM RTE suggested by the ADL organization, and then we outline our proposal. A deeper description of the core of the proposal is also provided. Finally, a preliminary experimental use of the platform implementing the system architecture is described.

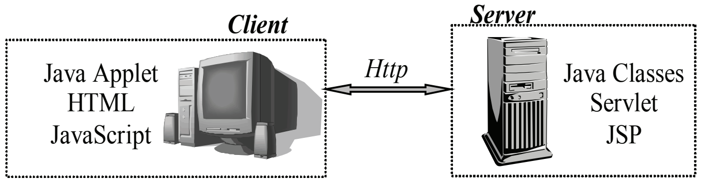
ADL Implementation of SCORM RTE

Although the goal of the ADL initiative is not to propose an effective implementation; nevertheless it suggests examples to fill the gap between the early standardization stages and the widespread adoption by industry, technical ideas, tools, and implementation. In particular, a sample of SCORM RTE implementation is provided by realizing the API and allowing communication between learning contents and LMS (SCORM RTE, 2004). This communication implements the learning traceability and the content sequencing and navigation. Thus, we highlight how this sample establishes content launch process and describe the standardized data model elements used for content communication.

It is worthwhile to start by noting that the ADL implementation is based on a Web client/server architecture, that is, the meaningful architectural elements are a Web server and a browser client. In Figure 2 is shown the communication conceptual schema proposed by SCORM. In particular, the server component implements the RTE by using HTML pages and J2EE technology, such as Java Classes, Servlets, and

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Figure 2. The communication conceptual schema of the SCORM sample



Java Server Pages (JSP). The client component needs a Web browser supporting ECMA script and Java applets to allow content communication. Indeed, the learning contents include ECMA script functions, which invoke an applet. This applet calls via HTTP the LMS, and more specifically, the RTE API as defined in SCORM.

It is worth noting that the sample implementation proposed by ADL is not suitable to ensure also an effective m-learning. Indeed, it requires that the browser is able to support technologies such as Java and ECMA script, which could not be the case in the context of m-learning due to the limited resources available on the client side.

The Proposed Architecture

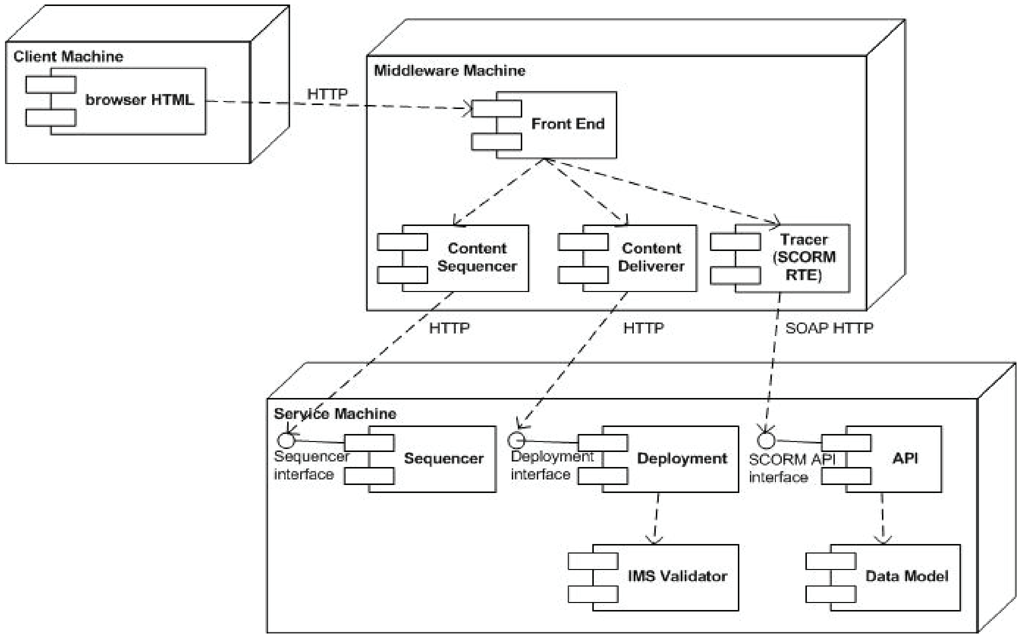
To simplify on the client side the technology infrastructure proposed by ADL, we propose an alternative implementation of the communication schema, which is based on pure HTML. This is possible thanks to a suitable Middleware, which has been conceived to allow the learner environment to properly communicate with the components of the LMS. In Figure 3 the role of the Middleware in the architecture is highlighted. The Middleware invokes the LMS software component which provides SCORM API to manage traceability and content sequencing. In our proposal, LMS publishes its functionality by using Web Services. As a result, the core of the LMS software component can be implemented by using any kind of technology. Moreover, the use of Web Services allows for extending the e-learning framework for supporting new functionality.

The Middleware is composed by three components: Sequencer, Tracer, and Deliverer. The former proposes to the learner the suitable knowledge content to consume. The Tracer is used to trace the learning process by interacting with the Service Machine. Finally, the Deliverer component allows the instructional designer to deploy courses. In order to manage the above functionalities the components on the Service Machine node are invoked.

It is worth noting that the components in the Middleware node have been separated for flexibility design goal. Moreover, all the components that realize the LMS functionality have been depicted in the service machine even if they might be on different nodes. The

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Figure 3. A UML deployment diagram representing the allocation of components to different nodes



proposed LMS can be extended with a new feature by introducing an entry in the Middleware that invokes the component that implements it.

The proposed architecture has been developed using J2EE technologies. We have exploited Apache (Apache, 2004) and Tomcat (Apache Jakarta Tomcat, 2004) as Web Server and Web container, respectively. Moreover, to integrate the component to trace the learning process we used a Web Service. It has been implemented by exploiting AXIS SOAP engine (Apache Axis, 2004), which is a framework developed in Java by the Apache Group for constructing SOAP processors such as clients and servers.

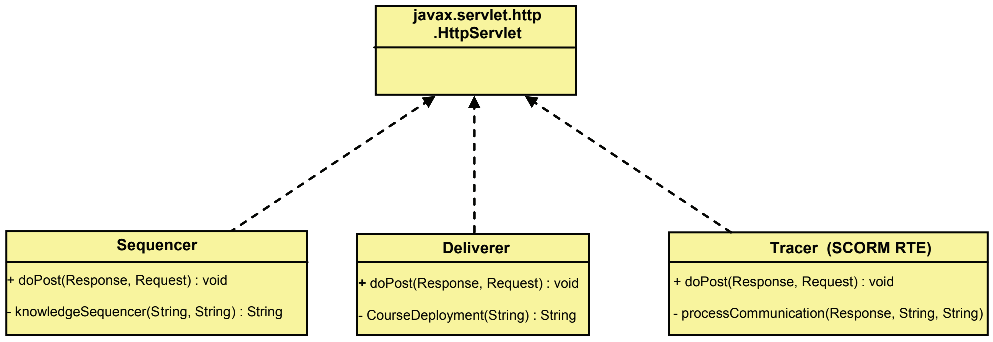
The Middleware

In the sequel we provide a deeper description of the Middleware using the class diagram in Figure 4. The Middleware contains the servlets Sequencer, Tracer, and Deliverer. These servlets extend the javax.servlet.http.HttpServlet abstract class, and exploit the HTTP protocol as communication infrastructure with the client machine. It is worth noting that the servlets are also used from the graphical front-end of the e-learning framework to allow the communication between the Middleware and the components on the service machine node.

The Sequencer obtains the URL of the knowledge content to present via

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Figure 4. The Middleware Class Diagram


javax.servlet.http
.HttpServlet
Sequencer
+ doPost(Response, Request) : void
- knowledgeSequencer(String, String) : String
Deliverer
+ doPost(Response, Request) : void
- CourseDeployment(String) : String
Tracer (SCORM RTE)
+ doPost(Response, Request) : void
- processCommunication(Response, String, String)

the knowledge sequencer method. This URL is used to redirect the client to the suitable knowledge content encoded in a HTML page. The knowledge sequencer method is private and is invoked by doPost to let the learner navigate among knowledge contents.

The e-learning courses are deployed by the method course deployment in the Deliverer servlet. The aim of this method is to validate and deploy a course in the LMS.

The Tracer class uses the process communication method to elaborate the request of the client. This method is invoked by doPost and uses client parameters encoded in the HTTP request. This class is used to manage the traceability by communicating with the component API on the Service node. SOAP on HTTP is used as a communication infrastructure between the Middleware and the SCORM API on the service machine. To enable the communication between the latter components we used the AXIS API (Apache Axis, 2004).

The LMS Component

In this section, we describe the classes to manage the content sequencing, the course deployment, and the LMS RTE. To implement the linear sequencing of the knowledge contents we have developed the Linear Sequence class. This class implements the interface Sequencing Interface, as shown in Figure 5, to allow content navigation among knowledge contents. It has three methods: start, next, and previous. The former is invoked when the learner takes a course up, while the other two methods are used to present the next and previous content. It is worth noting that any kind of sequencing can be obtained by suitably implementing the Sequencing Interface. As an example, this can be useful to get the presentation of adaptive contents (Casella, Costagliola, Ferrucci, Polese, & Scanniello, 2004).

To deploy a course in the LMS we have implemented a suitable component, which contains a parser to validate its SCORM IMS Manifest. After the

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Figure 5. The class diagram for content sequencing

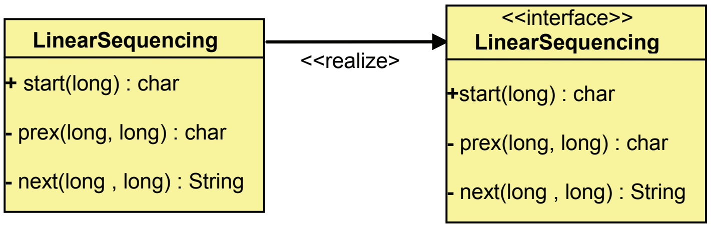
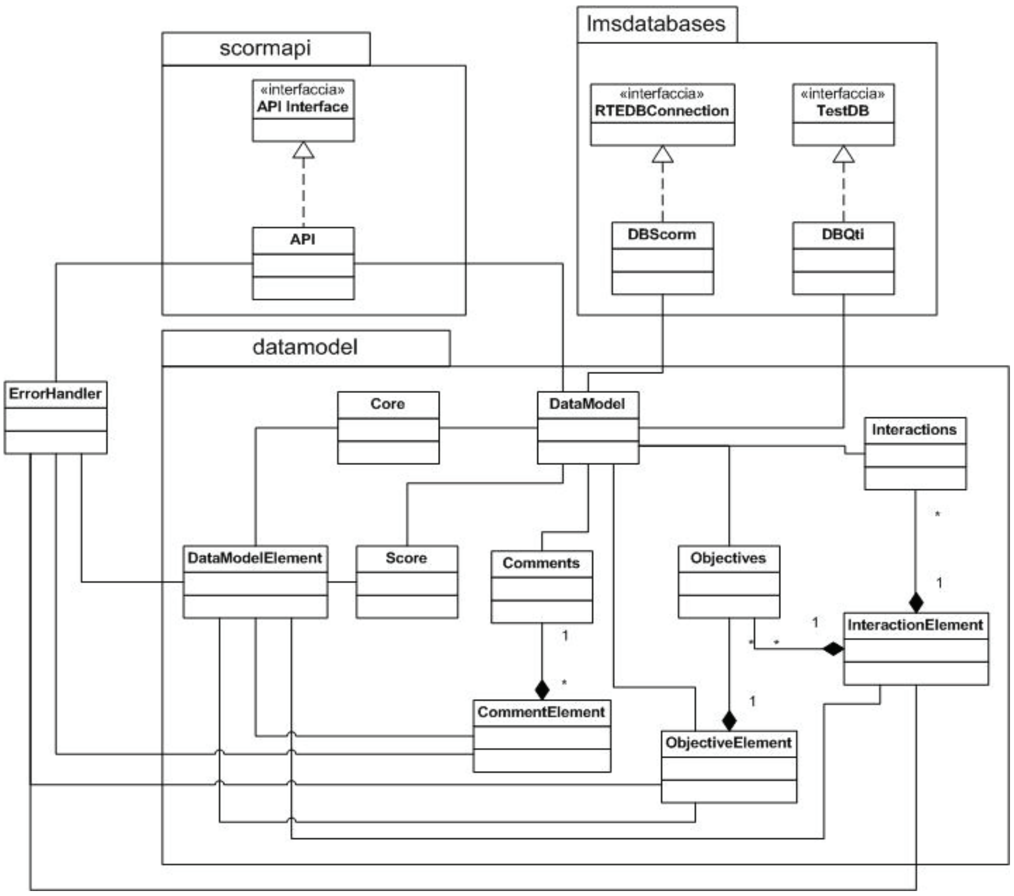
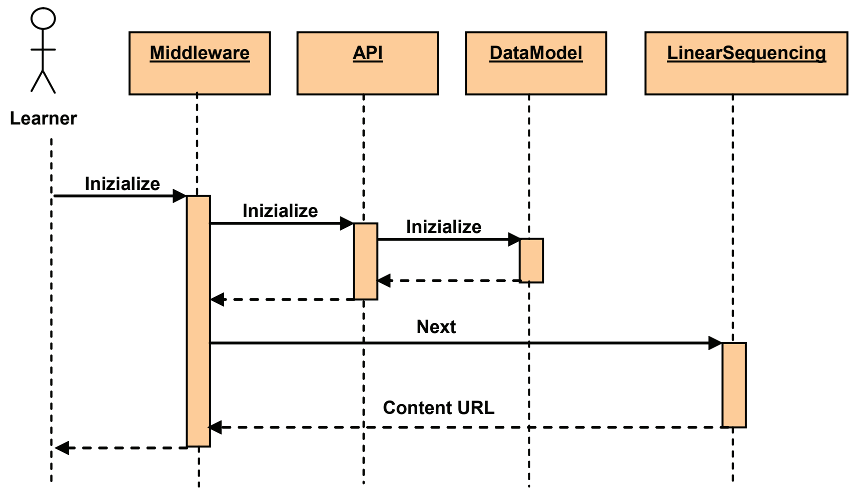

LinearSequencing
+ start(long) : char
- prex(long, long) : char
- next(long , long) : String
<<realize>
<<interface>>
LinearSequencing
+start(long) : char
- prex(long, long) : char
- next(long , long) : String

Figure 6. The RTE class diagram



Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Figure 7. Sequence diagram to initialize a learning session



validation, the knowledge contents composing the course are stored on the Server and the LMS Database is updated. Once the deployment process has been completed, the courses can be enjoyed by the learners.

The class diagram of the LMS RTE is depicted in Figure 6. The diagram includes the SCORM API, the SCORM Data Model, the Error Handler, and the LMS Databases.

APIInterface is the interface that contains the methods required to be compliant with SCORM. This interface is implemented by the API class, which uses the Data Model class to manage the learner traceability. Moreover, the API class exposes its methods as a Web Service. Information on the learner traceability is stored in a database using the DBScorm class. The DBQti class is used for storing information about learners on a given test. Finally, the errors are handled by the Handler Error class as specified in the SCORM standard.

To describe the interaction between actors and the above modules we use the Sequence Diagram in Figure 7. This picture shows the sequence for initializing the learning traceability process and getting the next educational resource.

When the learner starts by enjoying a SCO, an LMSInitialize call is sent to the RTE by the Middleware. The API returns the control to the Middleware once the DataModel class has executed its tasks. Then, the Middleware requires the next knowledge content for the learner via the Linear Sequencing class. This class returns the URL of the required knowledge content.

A Preliminary Use

In this section we outline a use example of the e-learning platform implementing the described software architecture. In particular, we adopted

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

this platform in a teaching experience carried out by undergraduate final year students of the Computer Science program at the University of Salerno. To this end, a group of ten students attending the Web Development Technologies course (WDT) was recruited. The WDT course aims at bridging the knowledge gap onto design and implementation of Web applications using XML and Java technologies. Thus, to carry out the preliminary experiment we defined linear knowledge contents for the XML topic. The linear knowledge contents were defined by using a particular version of the ASCLO-S editor (Casella et al., 2004).

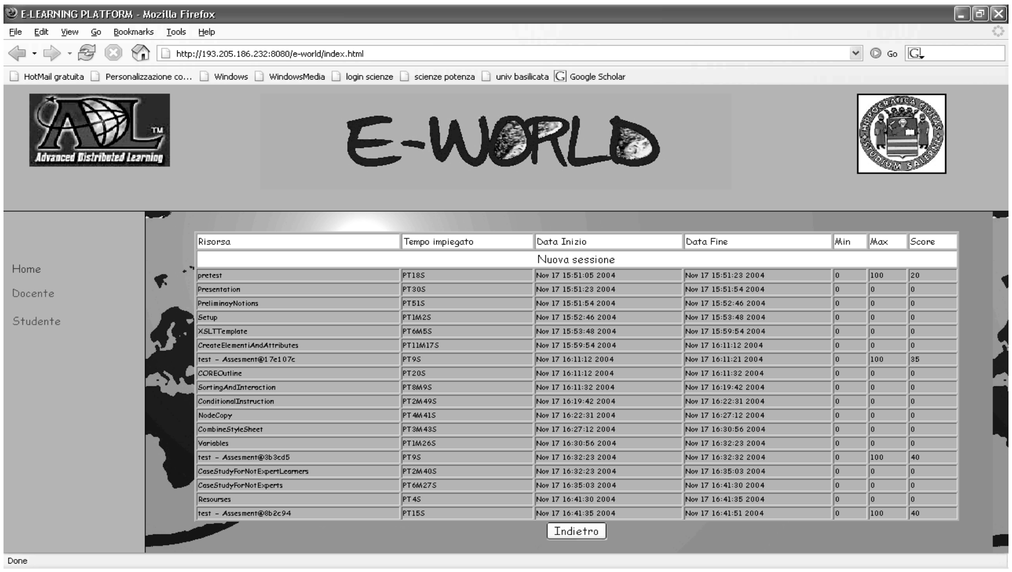
The experiment was divided into three steps and was carried out without invoking any kind of tutor support and time limit. First, an introductory course of ten minutes was presented to provide information about the experiment. After that, we asked to explore the platform to become familiar with it. The third step consisted of performing four tasks: creating an account to use the platform, selecting the course, enjoying the course, and accomplishing the test to assess the acquired knowledge.

The traceability report of a given student is shown in Figure 8. Traceability reports show both the time spent on each SCO and the scores of the tests accomplished during the learning process. Information on the sessions composing learning processes are shown when they are suspended and then resumed.

RELATED WORK

Today, there are many e-learning products on the market, which are

Figure 8. A platform snapshot


E-LEARNING PLATFORM - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://193.205.186.232:8080/e-world/index.html
Go
HotMail gratuita
Personalizzazione co...
Windows
WindowsMedia
login scienze
scienze potenza
univ basilicata
Google Scholar
Advanced Distributed Learning
E-WORLD
HIPPOCRATICA CIVITAS STUDIUM SALERNI
Home
Docente
Studente
Indietro
Done

Risorsa	Tempo impiegato	Data Inizio	Data Fine	Min	Max	Score
Nuova sessione						
pretest	PT18S	Nov 17 15:51:05 2004	Nov 17 15:51:23 2004	0	100	20
Presentation	PT30S	Nov 17 15:51:23 2004	Nov 17 15:51:54 2004	0	0	0
PreliminayNotions	PT51S	Nov 17 15:51:54 2004	Nov 17 15:52:46 2004	0	0	0
Setup	PT1M2S	Nov 17 15:52:46 2004	Nov 17 15:53:48 2004	0	0	0
XSLTTemplate	PT6M5S	Nov 17 15:53:48 2004	Nov 17 15:59:54 2004	0	0	0
CreateElementiAndAttributes	PT11M17S	Nov 17 15:59:54 2004	Nov 17 16:11:12 2004	0	0	0
test - Assesment@17e107c	PT9S	Nov 17 16:11:12 2004	Nov 17 16:11:21 2004	0	100	35
COREOutline	PT20S	Nov 17 16:11:12 2004	Nov 17 16:11:32 2004	0	0	0
SortingAndInteraction	PT8M9S	Nov 17 16:11:32 2004	Nov 17 16:19:42 2004	0	0	0
ConditionalInstruction	PT2M49S	Nov 17 16:19:42 2004	Nov 17 16:22:31 2004	0	0	0
NodeCopy	PT4M41S	Nov 17 16:22:31 2004	Nov 17 16:27:12 2004	0	0	0
CombineStyleSheet	PT3M43S	Nov 17 16:27:12 2004	Nov 17 16:30:56 2004	0	0	0
Variables	PT1M26S	Nov 17 16:30:56 2004	Nov 17 16:32:23 2004	0	0	0
test - Assesment@3b3cd5	PT9S	Nov 17 16:32:23 2004	Nov 17 16:32:32 2004	0	100	40
CaseStudyForNotExpertLearners	PT2M40S	Nov 17 16:32:23 2004	Nov 17 16:35:03 2004	0	0	0
CaseStudyForNotExperts	PT6M27S	Nov 17 16:35:03 2004	Nov 17 16:41:30 2004	0	0	0
Resourses	PT4S	Nov 17 16:41:30 2004	Nov 17 16:41:35 2004	0	0	0
test - Assesment@8b2c94	PT15S	Nov 17 16:41:35 2004	Nov 17 16:41:51 2004	0	100	40

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

implemented using different technical infrastructures. Often these products are not compatible with each other. To overcome the incompatibility problem Web Services represent a valid solution. In fact, Web Services provide a standard means of communication among different software applications, running on a variety of platforms. Xiaofei, El Saddik, and Georganas (2003) also propose a functional architecture based on Web Services for building standard-driven distributed and interoperable learning systems. The functional architecture defines components that make up an e-learning system and the objects that must be moved among these components. However, they do not use a Middleware to supportm-learning.

A proposal for a Middleware component was suggested by Apostolopoulos and Kefala (2003) with the aim to bridge the lack of a consistent management scheme in the integration of e-learning services. They implemented the e-learning components as agents, which are maintained in a local management information base, and can communicate with the agent manager through the SNMP protocol. Nevertheless, their LMS is not compliant with any e-learning standards and does not provide support for m-learning.

An architecture based on J2EE has been proposed from Wang and Zhang (2003) to run the e-learning services on different platforms. In their system, Platform Independent Model artefacts are modelled using UML, whereas Platform Specific Model artefacts are modelled using the UML profile for EJB for the target platform.

Shih et al. (2003) suggested the use of SOAP to implement API Adapter and transport parameter part in LMS. The LMS Web Services are implemented in .Net, while to enjoy knowledge contents the client needs specific software components. The components include a SOAP engine to allow the communication between LMS and client machine.

Finally, it is worth noting that the market and the academic community propose to use Web Services and other technologies based on software components with different aims. These technologies are useful for integrating new software components in e-learning systems and can provide a new communication schema between the server and the client machines.

CONCLUSION AND FUTURE WORK

In this paper, we have proposed an architecture of an e-learning system aiming to address several important issues. First of all, it is worth noting that current advances in computing and the development of pervasive applications intensify the diversity problem, giving rise to many variations in terms of performance, environments, and device characteristics. The use of a Middleware provides us with an integration framework for multiple and potentially diverse computing platforms. Moreover, the synergistic use of a Middleware component and

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Web Services turns out to be a suitable solution to integrate different software components, to easily extend the e-learning system with new features, and to improve interoperability among different systems.

Another important issue addressed by the proposed architecture is the effective support of student learning process traceability. As a matter of fact, the architecture integrates components, which have been specifically conceived to trace the student learning process, to deploy e-learning courses, and to manage the knowledge sequence to be presented to a learner. Moreover, the software that enables the learner traceability has been developed to be compliant with SCORM standard. Finally, the system architecture has been also conceived to effectively realize the learning anytime and anywhere. To this aim our efforts were addressed to let a learner enjoy knowledge contents via pure HTML browsers.

In the future, we aim to expand the LMS with new components, by exploiting the extensibility feature of the proposed architecture. In particular, our purpose is to manage adaptive contents and to support synchronous and asynchronous learning activities. On one hand, this will allow us to personalize the knowledge enjoying for each student. On the other hand, groups of students will be able to work together to solve problems while keeping their diversities. As a result, the learning environments will encourage individual accountability, prompt feedback, high self expectations, and students' welfare.

REFERENCES

ADL. (2004). Advanced Distributed Learning (ADL) Initiative. http://www.adlnet.org

ADL SCORM. (2004). Sharable Content Object Reference Model (SCORM). http://www.adlnet.org/index.cfm?fuseaction=scormabt

AICC. (2004). Aviation industry CBT committee. http://www.aicc.org/

Apache Axis. (2004). http://ws.apache.org/axis/

Apache. (2004). HTTP server project. http://httpd.apache.org/

Apache Jakarta Tomcat. (2004). http://jakarta.apache.org/tomcat/

Apostolopoulos, T. K., & Kefala, A. (2003). An e-learning service management architecture. *Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies* (pp. 140-144). Athens, Greece:.

Apple, D. K., Nygren, K. P., Williams, M. W., & Litynski, D. M. (2002). Distinguishing and elevating levels of learning in engineering and technology instruction. *Proceedings of ASEE/IEEE Frontiers in Education Conference* (pp. 6-9). Boston.

ARIADNE. (2004). ARIADNE foundation for the European knowledge pool. http://www.ariadne-eu.org/

Bosworth, A. (2001). Developing Web services. *Proceedings of 17th International Conference on Data Engineering* (pp. 477-481).

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Heidelberg, Germany.

Campbell, J. D., & Mahling, D. E. (1998). A visual language system for developing and presenting Internet-based education. *Proceedings of IEEE Symposium on Visual Languages* (pp. 66-67). Nova Scotia, Canada.

Casella, G., Costagliola, G., Ferrucci, F., Polese, G., & Scanniello, G., (2004). Visual languages for defining adaptive and collaborative e-learning activities. *Proceedings of IADIS International Conference: e-Society 2004* (pp. 243-250). Avila, Spain.

Chang, C. K., Chen, G. D., Liu, B. J., & Ou, K. L. (1996). A language for developing collaborative learning activities on World Wide Web. *Proceedings of 20th International Conference on Computer Software and Applications Conference* (pp. 548-552). Seoul , South Korea.

Cuthbert, A. J. (1999). Designs for collaborative learning environments: Can specialization encourage knowledge integration? *Proceedings of the Computer Support for Collaborative Learning* (pp. 117-126). Palo Alto, CA: Stanford University.

Designer's Edge. (2003). *The industry standard instructional design tool*. From http://www.allencomm.com/products/authoring_design/designer/

Douglas, I. (2001). Instructional design based on reusable learning object: Appling lessons of object-oriented software engineering to learning system design. *Proceedings of ASEE/IEEE Frontiers in Education Conference*, Reno, NV.

Goodyear, P. (1997). Instructional design environments: Methods and tools. In S. Dijkstra, N. Seel, F. Schott and D. Tennyson.

IEEE's LTSC. (2004). Learning technology standards committee, http://ltsc.ieee.org

IMS. (2004). Global learning consortium, http://www.imsproject.org

Kasowitz, A. (1997). Tool for automating instructional design. *ERIC e-learning house in Information Technology in Education*, Syracuse, NY, from http://ericit.org/digests/EDO-IR-1998-01.shtml

McKinley, P. K., Malenfant,A. M., & Arango, J. M. (1999). Pavilion: A middleware framework for collaborative Web-based applications. *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work* (pp. 179-188). Phoenix, Arizona..

PROMETEUS. (2004). European partnership for a common approach to the production of e-learning technologies and content, http://prometeus.org

Rosenberg, M. (2001). Quick tips for surviving the interoperability myth. *e-Learning Magazine*.

Roy, J., & Ramanujan, A. (2001). Understanding Web Services. *IT Professional*, *3*(6), 69-73.

Safoutin, M. J., Atman, C. J., Adams, R., Rutar, T., Kramlich, J. C., &

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Fridley, J. L. (2000). A design attribute framework for course planning and learning assessment. *IEEE Transaction Educational*, *43,* 188-199.

SCORM RTE. (2004). *Run-time environment book*. Retrieved from http://www.adlnet.org/screens/shares/dsp_displayfile.cfm?fileid=996

SCORM SN. (2004). *Sequencing and navigation book*. Retrieved from http://www.adlnet.org/screens/shares/dsp_displayfile.cfm?fileid=998

SCORM Sample. (2004). *Sample of run-time environment*. Retrieved from http://www.adlnet.org/screens/shares/dsp_displayfile.cfm?fileid=1103

Shih, T. K., Chang, W., Lin, N. H., Lin, L. H., Hsu, H., & Hsieh, C. (2003). Using SOAP and .NET Web Service to build SCORM RTE and LMS. *Proceedings of the 17th International Conference on Advanced Information Networking and Applications* (pp. 408-413). Tamsui, Taiwan.

Smythe, C., Shepherd, E., Brewer, L., & Lay, S. (2002). *IMS question & test interoperability overview. version 1.2*. Retrieved from http://www.imsglobal.org/question/qtiv1p2/imsqti_oviewv1p2.html

SUN. (2002). E-learning interoperability standards. *Sun White Paper*, Retrieved from www.sun.com/products-n-solutions/edu/whitepapers/index.html

Vrasidas, C. (2002). A systematic approach for designing hypermedia environments for teaching and learning. *International Journal of Instructional Media*. Retrieved from http://www.cait.org/vrasidas/hypermedia.pdf.

Wang, H., & Zhang, D. (2003). MDA-based development of e-learning system. *Proceedings of the 27th Annual International Computer Software and Applications Conference* (pp. 684-689). Ottawa, Ontario, Canada.

Xiaofei, L., El Saddik, A., & Georganas, N. D. (2003). An implementable architecture of an e-learning system. *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering, vol. 2* (pp. 717-720). Ottawa: Ottawa University Press.

ENDNOTE

[1] It is worth noting that often the knowledge content is meant as SCO.

Giovanni Casella is currently PhD student at the University of Salerno. His research interests include e-learning, visual languages, and software agents. He received the Laurea degree in computer science from the University of Salerno.

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Gennaro Costagliola is currently professor and director of the Laurea degree courses in computer science at the University of Salerno. His research interests include programming languages, visual languages, parsing technologies, multimedia databases, web technologies and e-learning. He received the Laurea degree in computer science from the University of Salerno, Italy in 1987, and an MS in computer science from the University of Pittsburgh in 1991. He was guest coeditor of the February 2002 Special Issue of the Journal of Visual Languages and Computing on Querying Multiple Data Sources. *He is a member of ACM, IEEE, and IEEE Computer Society.*

Filomena Ferrucci received the Laurea degree in computer science (cum laude) from the University of Salerno (Italy) in 1990. In 1995 she received the PhD in applied mathematics and computer science at the University of Naples (Italy). From 1995 to 2001 she has been a research associate at the University of Salerno where she is associate professor in computer science since November 2001. She was program co-chair of the 14th International Conference on Software Engineering and Knowledge Engineering and guest editor of the special issue of the International Journal of Software Engineering and Knowledge Engineering *dedicated to a selection of the best papers of the conference. She has served as Program Committee member for several international conferences. Her research interests are in the fields of human computer interaction, e-learning, and software engineering. She is co-author of about 60 papers published on international journals and proceedings of international conferences.*

Giuseppe Polese is associate professor in the department of mathematics and computer science at the University of Salerno, Italy. His research interests include visual languages, multimedia databases, e-learning, and multimedia software engineering. He received the Laurea degree in computer science from the University of Salerno, an MS in computer science from the University of Pittsburgh, and a PhD in computer science and applied mathematics from the University of Salerno.

Giuseppe Scanniello is research fellow in the department of mathematics and computer science at the University of Salerno (Italy) and teaching assistant in the department of mathematics at the University of Basilicata (Italy). His research interests include visual languages, software engineering, reverse engineering, reengineering, workflow management, and e-learning. He received the PhD and the Laurea degree in computer science at the University of Salerno in 2004 and 2001, respectively.

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.