

# A Component Model for Standardized Web-based Education

L. Anido-Rifón  
Universidad de Vigo  
ETSI de Telecomunicación  
E-36200 Vigo, Spain  
lanido@ait.uvigo.es

M. Llamas-Nistal  
Universidad de Vigo  
ETSI de Telecomunicación  
E-36200 Vigo, Spain  
martin@ait.uvigo.es

M. J. Fernández-Iglesias  
Universidad de Vigo  
ETSI de Telecomunicación  
E-36200 Vigo, Spain  
manolo@ait.uvigo.es

## ABSTRACT

This paper presents a layered component model to support Web-based interactive and collaborative applications. It is intended to let programmers focus on the particular logic of their applications, avoiding most of the issues related to collaboration, access control and networking management. A Web-based educational application has been developed over this framework. This tele-education system, which follows the recommendations by the main institutions involved in the learning technology standardization process, is the second contribution presented by the authors.

## Categories and Subject Descriptors

K.3 [Computing Milieux]: Computers and Education;  
H.3.5 [Information Systems]: Online Information Services—*Web-based services*; J.1 [Computer Applications]: Administrative Data Processing—*Education*

## General Terms

Standardization, Experimentation, Design, Security

## Keywords

Learning Technology Standardization, Collaborative systems, Authoring tools, Web-based course delivery systems, Practice and Experience, Educational Web Applications

## 1. INTRODUCTION

Designers of online learning systems have an enormous variety of software tools and learning resources at their disposal. Unfortunately, the wide variety of software tools available from many different vendors are not able to share learning resources and interoperate with each other. Nowadays, Web-based education research efforts are focused into the standardization of learning metadata schemes, course

structures and software interfaces to provide interoperability between applications and learning resources. This would allow both instructors to reuse learning resources and developers to reuse educational software.

Software reuse is one of the most outstanding research topics in Software Engineering. One of the current trends in this field is the component-based approach [24]. Components Object Model (COM) [28], Java Beans [30], Enterprise Java Beans (EJB) [29] and the most recent release (3.0) of the Common Object Request Broker Architecture (CORBA) [32] are some examples of technologies that are based on the software component concept. Software reuse allows programmers to focus their efforts in their specific business logic.

We followed this approach to design and develop SimulNet, a framework focused towards Web-based interactive and collaborative educational applications. Collaboration is allowed at software level. Events produced by a given component are handled and delivered to remote components in the same application. Collaboration is also possible at the user level. SimulNet components provide plug and play communication and collaboration tools based upon underlying and yet transparent for the programmer SimulNet services.

SimulNet also provides interactivity. It is designed to guarantee a quick response to any action performed by the user. Sometimes the inherent character of the application (e.g. an interface to access a remote hardware device) does not permit the highest level of interactivity. In that case, SimulNet does its best to enhance the interaction between the user and the software.

SimulNet has been tested in practice by more than 550 university students. Hypermedia and multimedia, the use of highly interactive computer-based simulations (e.g. using Java [23] applets) and the availability of powerful communication tools (e.g. video and audio conferencing, e-whiteboards, etc.) convert the WWW into an attractive platform to learn in an open and flexible environment.

## 2. WEB-BASED EDUCATION STANDARDIZATION

The learning technology standardization process is taking the lead role in the research efforts into Web-based education. Standardization is needed for two main reasons: On the one hand, educational resources are defined, structured and presented using different formats. On the other hand, the functional modules that are embedded in a particular

learning system cannot be reused by a different one in a straightforward way.

Organizations like the IEEE's LTSC [14], IMS project [21], Aviation Industry's AICC [1], DoD's ADL initiative [33], ARIADNE [3], GESTALT project [13], PROMETEUS [34], CEN/ISSS/LT [6], GEM project [12], and many others are contributing to this standardization process. The IEEE LTSC is the institution that is actually gathering recommendations and proposals from other learning standardization institutions and projects. ISO/IEC JTC1 has created the Standards Committee for Learning Technologies, SC36 [22], who is continuing this work to create ANSI or ISO standards.

Learning metadata definition has been one of the main objectives. Its purpose is to facilitate and automate the search, evaluation, acquisition and use of learning resources on a Web environment. The most outstanding result so far is the Learning Object Metadata (LOM) specification [15], proposed by IEEE LTSC, which is becoming a standard de-facto. LOM defines the attributes required to fully and adequately describe a learning resource. Relevant attributes include: general resource (e.g. title, language), life cycle (e.g. version, contributions), meta-metadata (e.g. cataloger, catalog entry), technical (e.g. format, location), educational (e.g. learning context, difficulty), rights, relation, and classification information.

Description of learner's profiles and records has also been studied. They describe information which characterizes a learner and his/her knowledge. This information is used to maintain records of learners and carry out the learning process. The IEEE LTSC Public and Private Information (PAPI) specification [16], describes implementation-independent learner records organized in six categories: personal, relations, security, preference, performance and portfolio. Another important specification is the IMS Enterprise Data Model [19] aimed at administration purposes.

Standardized definitions for course structures are necessary to move courses from one system to another. In a similar way, learning management environments need to understand the course structure to schedule the next student activity. The AICC guidelines for interoperability of Computer Managed Instruction (CMI) systems [2], and the ADL's SCORM [38], based on the AICC specification, deal with this problem. They propose a Web-based run time environment scheme based on the division between the learning contents and the Web-based management system that launches on the browser, delivers through the network and controls them. This allows different learning resources to be managed by heterogeneous management systems.

Other standards are referred to course packaging [18], question and test interoperability [20], student tracking, capability definitions, and many others.

### **3. SUPPORTING WEB-BASED HIGHLY INTERACTIVE AND COLLABORATIVE EDUCATIONAL APPLICATIONS**

#### **3.1 Overview**

Our framework is organized as shown in the layered architecture outlined in figure 1. In the lowest layer SimulNet relies on commercial-of-the-shelf (COTS) services and standard Internet protocols. In the next layer, there are some

common SimulNet services. These are used by the SimulNet Software Components (SSC). Their aim is to make the distributed nature of the WWW transparent to the SSC programmer. The software that actually implements the SimulNet services runs both at client and server sides.

The actual functionality of SimulNet is offered at the components level. Each component is aimed to provide a particular feature typically found in Web-based collaborative and interactive applications. All components are plug and play and, in most of the cases, programmers just need to invoke a Java method to provide their application with a new collaborative feature (e.g. a shared e-whiteboard). The SSC run embedded in applications at client side. They are stored at server side and downloaded in the form of Java applets. Collaboration among distributed components is possible thanks to the use of the underlying SimulNet services. The programmer may include the functionality of these components through the SimulNet API (figure 1).

Finally, the application level is the core for the business logic of particular Web-based applications. This is the programmers layer, where their specific application is developed. Collaborative functionality is automatically included through the SSC APIs. Although SimulNet is oriented towards Java-based applications, in fact, SimulNet services and SSC are 100% pure Java to provide platform independence, it allows interoperability with applications developed in other programming languages. This is possible thanks to the interoperability provided by the CORBA Object Request Broker (ORB). In this case, the client applications provide interactive user interfaces to access the actual applications at the server side together with the collaborative functionality provided by the SSC. Therefore, formerly standalone applications (i.e. Business applications at the server side in figure 1) are transformed into Web-based collaborative systems using a SimulNet envelope.

At the top of figure 1 the dynamic evolution of SimulNet is illustrated. SimulNet programmers would enrich the functionality of our framework with new components, and possibly new services. The business application developers would adopt our framework to develop new Web-based interactive and collaborative applications, focusing their efforts towards their particular business logic.

#### **3.2 COTS Services and Standard Protocols**

On the client side, basic user interfacing and connectivity is provided by a standard WWW browser (e.g. Netscape Communicator or Microsoft Explorer) with some embedded capabilities: a Java Virtual Machine (JVM), drivers for the Java DataBase Connectivity (JDBC), and a CORBA Object Request Broker (ORB). On the server side, HyperText Transfer Protocol (HTTP) and File Transfer Protocol (FTP) servers, a CORBA ORB, and a Database Management System (DBMS) are used to provide access to the SimulNet database. Regarding the network protocols, our services use HTTP, FTP, Internet Inter-ORB Protocol (IIOP), and TCP/IP.

#### **3.3 SimulNet Services**

The SimulNet Services layer is responsible for providing the SimulNet Software Components with the underlying collaborative functionality. Basically, they allow the SSC to share information both on-line, in which virtual memory spaces are provided where one client writes information and

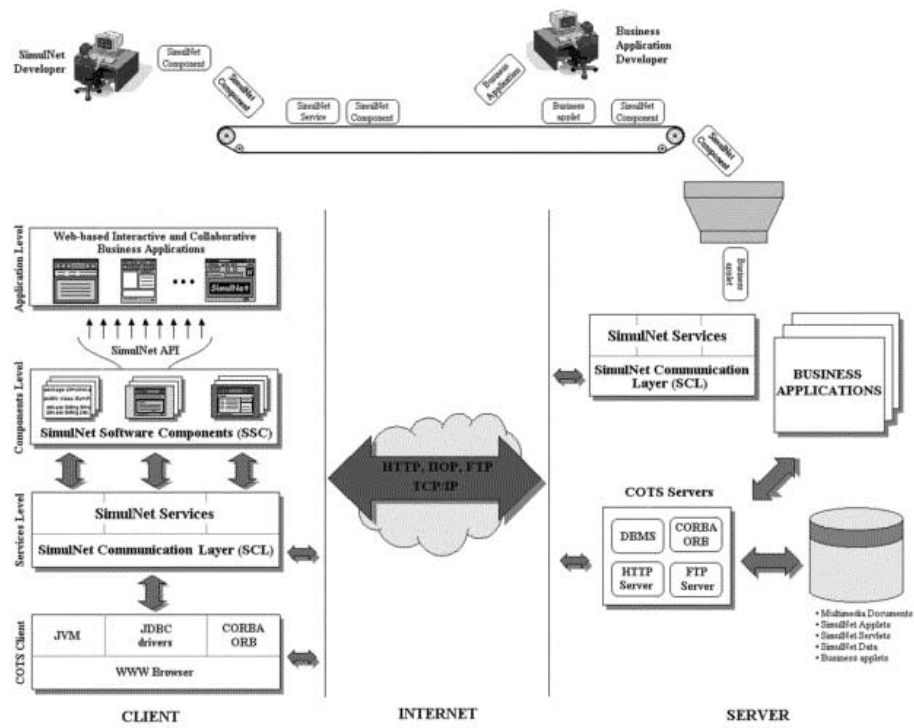


Figure 1: The SimulNet layered architecture.

the other reads it, and off-line, where a virtual shared persistent storage is available. Therefore, SSC avoid dealing with the underlying network. For a given component, the distribution of user input to a collaborative group is performed through the invocation of a local method from the suitable SimulNet service. This basic functionality is implemented by three different services: the Virtual Room Service (VRS), the Virtual File System Service (VFSS) and the Data Base Access Service (DBAS). Underneath them, the SimulNet Communication Layer (SCL) deals with the access to the network. The implementation of all the services discussed is outlined as follows.

### 3.3.1 The SimulNet Communications Layer

This layer provides a network-transparent model to exchange messages between distributed instances of SimulNet services. The same generic message format is used by all SimulNet services. We use serializable objects in the SimulNet messages, which are exchanged through buffered object input and output streams over the network. The type of messages that can be sent are general enough to cover from text to Java events.

A SimulNet message is a serializable object composed by a header and a data component. The header identifies the message type and allows the SCL to forward the data component to the suitable SimulNet service. The data component contains the information itself and the identifier for the data type.

The SCL uses input and output streams to read and write from/to the underlying TCP/IP socket. On the client side, a thread is responsible for these actions. On the server side, we rely on a set of threads to serve messages to/from the different connected clients.

### 3.3.2 The SimulNet Virtual Room Service

The Virtual Room Service (VRS) is responsible for providing a suitable environment for on-line communication among a group of distributed users or software components, avoiding simultaneously interference between different rooms. Conceptually, a virtual room is a shared space where one or more write-enabled members can send information to be received by every read-enabled member in the same room.

A new virtual room is set up whenever a component needs to communicate or collaborate with other remote instances of the same component. Afterwards, all information generated from one client is forwarded to every other client in the room. In addition, it is possible to give different privileges to different members in the same room. For instance, one client may be allowed to read and write, while another may only have read permissions. Each room has its own identifier to avoid any interference. Therefore, the same component in a particular client may be involved in different collaborative sessions.

Since this service uses SimulNet Communications Layer (SCL) messages, the kind of information that can be exchanged is rich enough. Furthermore, the virtual room service is not concerned with the type or particular logic of the messages. It is just responsible for delivering them to the correct addressees. In addition, the VRS maintains the last set of messages generated in each room to be delivered to new clients. The number of messages that are temporarily stored is defined by the software component using the virtual room.

On the client side, this service maintains up-to-date information about each session (virtual room) that its upper components are involved in. It transfers messages between the components and the appropriate virtual room at the

server side, and generates control messages for room creation, new member notifications, privilege management, etc. At the server side, the virtual room manager acts as a factory of new virtual room instances and manages all current active sessions. It also maintains up-to-date information about each client connected to the system.

### 3.3.3 The SimulNet Virtual File System Service

The SimulNet Virtual File System Service (VFSS) is responsible for persistent storage. Its API provides the common functionality of a tree-structured file system. Data and objects stored in files can be shared among different components and users. Control of concurrency is provided through locks and monitors on each shared resource. The underlying storage container is the native file system at the server side.

### 3.3.4 The SimulNet Data Base Access Service

This service handles automatic database connections. Once a connection is successfully established, the Java package `java.sql` provides standard access to the remote Database Management System.

## 3.4 SimulNet Components

The API of the SimulNet Components level provides developers with a facility that allows them to develop their applications more quickly since many of the common features of Web-based collaborative applications can be automatically included in their systems. All of them are supported by the basic functionality provided by the SimulNet services.

Most of these components provide their functionality through their own user interface. To launch a given component, the business application developer just needs to link an event on the application interface (e.g. a double click on a button) to the method `userAction()` on that component. The tool is automatically started in the business application frame. If the user must be identified in the collaborative session that the tool is involved in, the only parameter needed is the user's ID, which is provided by the application loader through its user authentication module.

Although the SimulNet components are designed to be open and flexible enough to suit any kind of Web-based collaborative application, the underlying data model follows recommendations and proposals by the main institutions involved in the learning technology standardization, see section 2. This means that they are able to interact with other Web-based educational systems provided they follow the standards and, at the same time, developers from other fields may use them taking in mind the data model chosen.

### 3.4.1 User Management

One of the key aspects of the design of collaborative tools is the identification of users (usually human) and their roles. Furthermore, resources may have access requirements that depend on user properties.

The SimulNet User Management component provides business application developers with a graphical tool to identify users, roles, resources, and access permissions to them. The tool is generic enough to be used in a wide range of applications. Developers identify roles and resources for their particular applications. Then, this component provides the application manager with a user-friendly interface to manage the system. For example, in a teleworking environment the business developer may identify three different roles: em-



Figure 2: Trace window from the Auditing Tool.

ployers, managers and employees.

The DBAS service supports data management for this service. Data is stored and managed following the recommendation by the IMS project, the Enterprise Data Model. Therefore, it is possible to import and export student profiles from/to other Web-based academic institutions.

### 3.4.2 Auditing Tool

Some business applications may need registering the actions that each particular user executes on them (e.g. which application packages are used, when they are used, for how long and by whom). The SimulNet auditing tool manages all this information. The business application programmer sets up this module with the actions that need to be audited. Whenever the user performs any of them, a message is sent to the suitable supervisor (for on-line auditing) and stored at the server side (for off-line monitoring). This component relies on the VRS service for on-line monitoring, and on DBAS and VFSS services for off-line monitoring. The data model used is the AICC's Lesson Evaluation Data, which comprises Interactions, Objectives Status, and Path groups, plus the Student Data group from the lesson to LMS communication formats, see [2] for further information on student performance data formats.

Figure 2 displays the trace window of the auditing tool, where some student traces are shown. Complete data for each trace is available on request.

### 3.4.3 E-Mail

This component provides an e-mail system. Additionally, an address book is sent through the network to the client's computer containing up-to-date information about the users of the corresponding business application. Users are organized according to different criteria (e.g. roles) and, therefore, it is very easy to find the right addressee for a given message, both individuals and groups. This component uses the underlying VFSS and DBAS services to store messages in mailboxes and access users' data.

### 3.4.4 Bulletin Board

The SimulNet bulletin board is based on the Internet News. Its main aim is to become a platform to share information. Users are allowed to post two kinds of documents: public notes, which can be read by any user, and private notes. Whenever users want to refresh the contents of their boards, they can select the types of documents to be downloaded from the ones they are allowed to access. In addition, users can select to download only documents related to a particular bulletin. This tool also provides information about what kind of bulletins has already been down-

loaded, and, if appropriate, the particular subjects the documents are related to. Bulletins are stored on the server side through the VFSS service.

### 3.4.5 Chat

The main purpose of this tool is to allow a group of users to get in contact with each other in a virtual room. Users can communicate with each other in two different ways:

- Debate mode. Every user involved in the communication may enter information. This information is shown to every user in the virtual room.
- Lecture mode. Only the user acting as the moderator can introduce information to be shown on the other users' display. The moderator can switch between the two communication modes or may include or detach users from the active session.

### 3.4.6 Whiteboard

Virtual electronic whiteboards seem to be one of the most useful tools in collaborative environments. SimulNet provides application developers with a whiteboard component, as shown in figure 3, that can be embedded in their business applications to exchange graphical information. Advanced drawing features are also included as part of the whiteboard functionality. As well as in the case of the chat, there is a moderator who is responsible of user management and for giving different privileges to the whiteboard users (draw, delete, etc.).

It is also possible to manage several documents in a given whiteboard session. The document manager decides which document is shown at a given time. Therefore, it is possible to navigate through the whiteboard contents, showing different information to the users involved in the session. This feature converts the SimulNet whiteboard into a virtual slide projector. The whiteboard, as a standalone drawing tool, can be used to develop a presentation that can be stored and later shown to others using the document management facility.

The whiteboard component relies on VRS and VFSS, and on users' data generated by the user management component.

### 3.4.7 Agenda

This component provides an electronic agenda and a calendar. Its basic functionality involves managing, scheduling, and representing data events in educational settings. We followed the vCalendar specification [17] to allow data agenda exchange.

Users are allowed to customize several agenda features: event types, properties and representation appearance. In addition, they can use the agenda following two configurations: personal or group. Personal configuration makes this component a traditional agenda, where users can schedule their upcoming events and tasks. In the group configuration the agenda can be used like a timetable or calendar. All group members are allowed to display the agenda events, which may be introduced by any of them. Class activities are scheduled using this component, which is implemented over the DBAS.

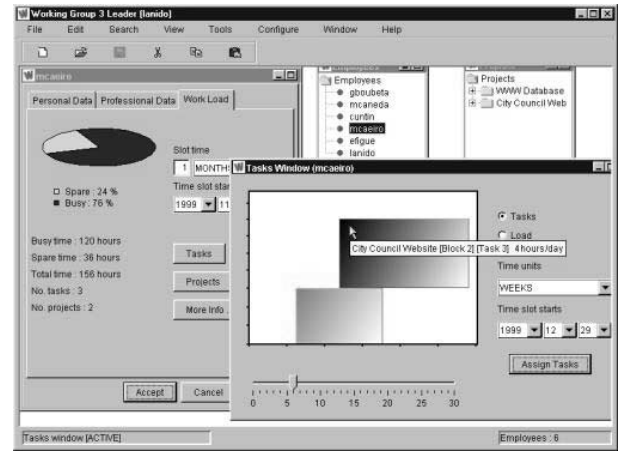


Figure 4: Project Management Tool.

### 3.4.8 Project Management

This component is oriented towards the management of projects carried out by a group of students. It supports project-based learning on the WWW. One of the most challenging aspects in this field is to know the latest progress done. This tool provides automatic management for projects whose working tasks are distributed among users who cooperate from afar.

The project manager distributes the total workload into a set of tasks. Afterwards, tasks are assigned to working groups or individuals. This assignment is supervised by the tool, taking into account the current workload for each group and student and correspondence among different tasks (e.g. task A must be completed before starting task B). The manager can set up the schedule for each working group and individual, as shown in figure 4.

The project management component provides each learner with information about the present-day tasks assigned, tasks that are piling up, etc. The users inform the tool about the development of their assigned tasks to update the progress done on the entire project.

This component uses the VFSS and DBAS services, and data from the user management component.

### 3.4.9 Event Deliverer

SimulNet allows, through this component, to share and distribute events performed on the business application user interface. It is possible to define a group of users whose actions on the interface will be forwarded to every other user in the group. As a result, actions performed on a master-enabled user interface are also performed on the interface of every other user in the same group. A moderator provides master privileges to formerly read-only participants. This component is based on the SimulNet VRS and a proprietary graphical SimulNet package with the same functionality as the provided by the java.awt package plus a new feature to support event delivery through the network.

### 3.4.10 Producer-Consumer Manager

The producer-consumer problem is a common topic in Software Engineering. There are situations where an entity, the consumer, needs some data provided by another entity, the producer. This component addresses the producer-

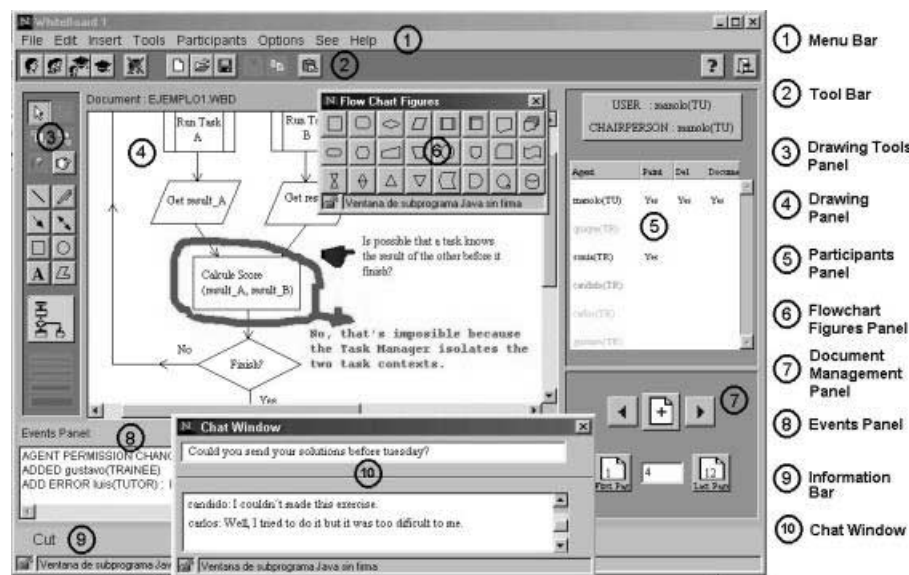


Figure 3: The SimulNet Whiteboard.

consumer problem in a distributed environment (i.e. producer and consumer running on different computers).

The Virtual Room Service is used. A VRS room is shared between the write-enabled producer and a read-enabled consumer. In this way, we are implementing a one-way channel through the network. The consumer accesses the data invoking a method on this component, if it was already delivered by the producer. Otherwise, the component waits until the producer delivers the data. The component allows blocking and non-blocking calls for the consumer. Producer and consumer can exchange their roles just by exchanging their access privileges to the room. This model can be easily extended to a scheme with  $n$  elements using  $n-1$  rooms.

### 3.5 Protection of Information and Resources

The SSC and the data generated by users are resources whose access must be controlled to avoid unauthorized use. Access control is managed using rights and principals. A right is an abstract privilege. A principal uses a right. A principal can be an individual or a group defined for collaborative purposes.

The project management component allows the definitions of working groups composed of individuals. Such working groups can be considered as protection groups, i.e. principals that may be used by several different individuals. Each individual belonging to a group benefits from the access privileges granted to its principal.

For each principal we identify the domain of that principal: all the objects in the system that the principal has been authorized to use. SimulNet uses the two general protection mechanisms described in [40]: capabilities and access control lists.

Access to SSC and business applications is controlled using a capability-based system [7]. The SimulNet capability system is able to grant all-or-nothing and controlled sharing access rights [35]. The former is used to protect some SSC like user management, e-mail, agenda and data sharing components. The all-or-nothing capability allows the principal to use the whole functionality of the component.

Controlled sharing capabilities are used to grant access to SSC like the auditing tool, bulletin board, chat, whiteboard, project management tool and the event deliverer. In this case, the capability includes a 32-bit access mode vector, each bit representing an abstract right. The semantics of each bit depends on each software component and may vary from one component to the next.

An Access Control Lists (ACL) [31] is used to protect each piece of user-generated data. Each file containing data that is susceptible to sharing has an ACL associated with it. This is a list of principals authorized to access the data. Each entry includes a 32-bit access mode vector. As in the case of the capability-based protection mechanism, each bit represents an abstract right. In this case, the first six bits are reserved for the access modes defined in [31]: read, write-append, write-change, write-update and delete. There are 26 more bits whose meaning may depend on the particular business application or SSC. We follow a discretionary control pattern described in [35].

## 4. AN INTERACTIVE AND COLLABORATIVE WEB-BASED LEARNING SYSTEM

The first application of SimulNet was in the area of Web-based education. An educational environment over the WWW is restricted with respect to interactivity and collaboration. Distance education environments need to provide trainees and instructors with a cooperative learning atmosphere, where students can share their experiences and teachers guide them in their learning. Interactivity is another key issue in both conventional and distance education. Student interest in the learning process is drastically reduced when the level of interactivity is low. The aim of developing an educational system over SimulNet is precisely to test its interactive and collaborative features.

On the other hand, the underlying data model used for handling course structure, students' performance data and communication among the learning contents and the management system follows the recommendations by the main

institutions involved in the learning technology standardization process, see section 2. This standardization process seems to be an unavoidable consequence of the successful, but at the same time heterogenous, Web-based educational platforms available on the market. WebCT [39] by British Columbia University of Canada, Learning Space [25] by LOTUS, Top Class [37] by WBT Systems, ToolBookII [4] by Asymetrix Inc., or Web Course in a Box [27] by the Virginia Commonwealth University are some examples of systems that provide different Web-based educational environments, each of them with their specific features [10]. Nevertheless, they all share some underlying common features as their final aim is the same: to deliver learning through a Web environment. All of them manage course structures, learning resources and student profiles, however, they do not provide the means to share that data. SimulNet provides the means to develop reusable and interoperable Web-based learning environments.

#### 4.1 Using SimulNet to Deliver Web-based Learning Contents

As mentioned before, one of the biggest problems when working with different Web-learning platforms is the difficulty of sharing and reusing components between different content management systems. SimulNet follows the guidelines established by ADL SCORM which are based on the AICC document "CMI Guidelines for Interoperability" [2]. In our system, both content and content management are clearly separated in order to get interoperability between different content management systems.

For this to be possible it is necessary that content (assignable units, using ADL terminology) and the Learning Management System (LMS) can communicate with each other through standard data types. LMS sends data to contents with necessary information to launch a lesson. Content provides LMS with information about student progress, tests results, student state within the course lessons, etc. These data allow the LMS to both track student progress and make "routing" decisions through the course.

SimulNet provides a Web-based run time environment through an API that standardizes the way the LMS sends and receives information from the contents. This API provides the needed methods to start and finish learning contents and communicate with them:

- `LMSInitialize()`: must be invoked by contents once they are ready to establish communication with the LMS.
- `LMSFinalize()`: must be invoked by contents once they finalise its execution.
- `LMSGetValue(element_name)`: used to get the value of one data model element.
- `LMSSetValue(element_name,new_value)`: used to modify the value of one data model element.

The SimulNet data model is known by both the LMS and the learning content. API runs as an applet inside an html page and its methods are invoked by the content through JavaScript calls.

We implemented our own LMS that is used to deliver and manage contents, see figure 5. Students' interaction data,

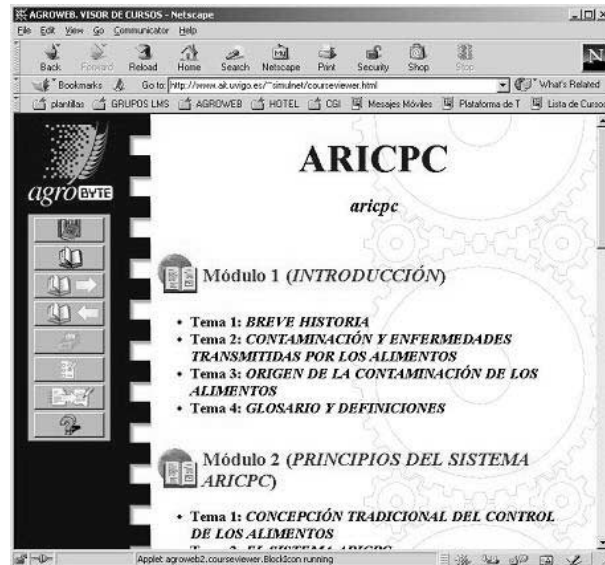


Figure 5: SimulNet's LMS user interface.



Figure 6: SimulNet's course authoring tool.

as mentioned above, is sent from the learning content to the LMS.

The SimulNet authoring tool, which is shown in figure 6, supports the design and creation of learning contents, the definition of the static and dynamic course structure and the creation of tests and questionnaires. The definition of course structure is made following the standard defined by the AICC. Course structure definition is compliant with AICC Level 3a [2]. This level allows the definition of simple and complex prerequisites and completion rules. Both prerequisites and completion rules define the dynamics of the course structure: prerequisites are used to make routing decisions and the completion rules are used to evaluate the status of the learning content (AICC defines the following statuses: pass, fail, complete, incomplete, browse and not-attempted). Figure 6 illustrates the definition of prerequisites for a course element. The definition of questionnaires and tests is also possible using our authoring tool. That definition is made using the IMS Question & Test Specification [20] that specifies the data format for tests interoperability.

## 4.2 Coping with Practical Training: the Learning-by-doing Paradigm

Web-based learning must cope with practical training. There are two main approaches to implement WWW/Internet-based training laboratory environment: distant access to the real equipment and simulation-based laboratories. It is possible to find references to systems that include both approaches. Some tools that provide Web-based access to the lab equipment exist, both for hardware-based labs [5] and for software-based labs [26]. Regarding the simulation approach, we were able to find systems that implement a WWW interface to a simulator that runs at the server-side (e.g. [9]) and Java-based educational simulators that run interactively on the student's computer using a Java-enabled browser (e.g. [11]).

Obviously practical and theoretical contents differ in the way they are designed and handled. Practical training must provide tools that simulates an environment similar to that provided by real laboratories. The difference with theoretical contents is clear: whereas practical contents must be highly interactive and dynamic, theoretical contents are almost static and the interactivity with students is restricted to browsing the contents and in some cases to doing tests. Furthermore, practical training needs high computing requirements whether simulators run in server side or in client side.

Thanks to its layered architecture, see figure 1, SimulNet can deal with all approaches to practical training discussed above. Software tools (i.e. educational simulators or industry applications) or hardware devices running in the lab itself, can be accessed through the CORBA ORB. Business programmers have to develop a software bridge between these legacy systems (business applications on the server side in figure 1) and the CORBA ORB. On the client side, the business applet uses an interactive user interface and offers access to the full cooperative environment provided by SimulNet. The “processing on each client approach” can be implemented using simulators downloaded by the application loader component to the client, where they run interactively as Java applets. We based our experience on this latter model.

Simulators related to Computer Architecture subjects have been developed. Virtual computers are used to introduce students to the world of Computer Architecture, as described in [8], from the simplest to the most complex. Thus, students are guided along the right path, starting from a simple assembler language and ending with the complex world of microinstructions, firmware and datapaths. All of our simulators have several common features: an editor, an assembler and the simulator of the computer itself (see figure 7). Programs written by students in assembler code using the editor are transformed into object code that can be run by the computer model. This task is done by the assembler. The simulator of the computer model executes the object code, which can also be displayed through the memory viewer. It has several execution modes: complete run, trace, step by step.

As it was mentioned in section 4.1 SimulNet implements a run time environment through the DoD's ADL API. Web-based simulators are handled as assignable units that are launched from the LMS. Data information about students interaction with simulators is sent from them to the LMS using the methods provided by the API. This information

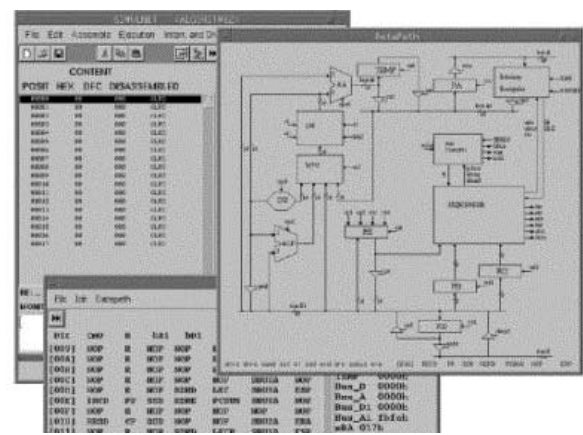


Figure 7: An educational simulator.

is used by the LMS to generate student tracking information that is stored in a file following the format for student tracking information proposed by the AICC.

### 4.2.1 Collaborative Functionality

WWW/Internet-based learning systems need to provide students and teachers with suitable tools that allow them to get in touch with others. SSC are used to give our learners and instructors the needed communication tools. The SimulNet whiteboard, e-mail, chat and bulletin board are included, see section 3.4. It is also possible to create working groups to develop a given training exercise whose workload is distributed among a group of students. All this functionality is embedded in our educational platform thanks to the SimulNet SSC.

## 4.3 Educational System Evaluation

As in every other software development process, feedback from users is necessary to tune up the best possible platform. In this case, the data was collected from surveys among 180 students, and from the information generated by the auditing tool.

The first remarkable result was a general improvement in students' theoretical exam results. The number of students who succeeded in the theoretical exam increased about 14%. In the practical exams, 81% of the grades for students who attended the traditional laboratory were equal or better than B. This percentage was 84% for those using the educational platform. Therefore, the performance of these “telestudents” was as good as that of our “conventional students”. This fact shows that our approach to teleteaching does not necessarily affect the quality of the learning process.

Using SimulNet's Auditing Tool, we collected information about every student's access to the system. Most students use the virtual laboratory between 8:00 p.m. and 12:00 p.m. There are also many accesses to the system late at night, and 90% of the students use SimulNet after everyday activities. Time independence, one of the main advantages of distance learning, is clearly stated here: students take full advantage of the platform's flexibility and use this tool whenever they want.

Electronic mail is the tool students use more often, probably because they do not have to be in the virtual laboratory



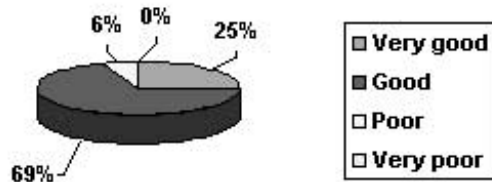


Figure 8: General opinions about SimulNet.

at the same time to communicate. Furthermore, it is an easy to use tool that allows students to send messages to their teachers, without the need of a face-to-face conversation.

Students commonly shy away from conversations with their teacher. Most conversations (65%) between a teacher and a student are started by the teacher. On the other side, when teachers detect that one of their students have problems, thanks to the tutoring facility, they usually start communicating immediately.

Students were also required to express their personal opinions about the SimulNet communication tools. 91% of the students rated these tools as good or very good. Some of them mentioned that they would use them more often if they were provided with a high speed Internet connection. During this survey, most of them accessed the system through a 28 kbps connection, although actual transfer rates were lower.

Students were also required to comment on the educational platform as a whole, from remote access to the simulation of the actual laboratory tools, including the communication tools. Again, 94% of the students rated the system as good or very good, see figure 8. As expected, interactivity and the collaborative feeling this platform provides are the most outstanding features.

Finally, students were asked if this approach to practical training indeed helped to understand the theoretical concepts. As expected, results were coherent with the results of the corresponding exams. Only 7% of the students considered that SimulNet did not contribute to their learning process.

## 5. CONCLUSIONS AND FUTURE WORK

We have presented a framework for developing interactive and collaborative Web-based applications. To test the developed framework we implemented a Web-based distributed educational platform following international standards on learning technologies. The standardized learning data model allowed us to reuse learning contents from others that follow the same standards. Nowadays, we do believe that the learning standardization process is absolutely essential to let Web-based education to become real world wide. Past educational environments did not allow to reuse learning contents and, therefore, a lot of work was needed to share resources. This educational system was our business application for the framework. The testing phase provided

valuable information both at technical and business level.

The SimulNet design process was based on a bottom-up strategy. However, when designing some of the SCC we detected that the available underlying services would not provide the required functionality for that component. For example, the VRS was updated in order to be able to handle GUI events as required by the whiteboard component. As new components are introduced into the system, an upgrade of existing services, even the introduction of new ones not previously considered, may be needed as well. Therefore, the service layer design should be kept simple and open, and separate features should be kept as independent as possible.

As mentioned above, most SSC have their own user interfaces. These components were designed for simple and straightforward integration into collaborative business applications. However, this could not be the best approach when the business application requires a common look and feel for all its modules, including the ones provided by SimulNet. For example, in our educational environment some simulators have user interfaces not fully compatible with the ones in the SimulNet components, where similar commands are accessed in different ways. This can be misleading for students, and completely inconvenient for commercial applications. In fact, there exist recommendations by the AICC [1] that try to standardize user interfaces used in computer-based training applications.

SimulNet is 100% pure Java, and Java is an interpreted language. This limitation becomes apparent as the number of simultaneous users and the number of active components grows, for example, many students accessing our educational system or a full-featured educational system. In fact, most of students' complaints about system performance were about time responses from the server side. One solution we are considering is the optimization of the server modules based on a combination of native and Java code. At this time, we are carefully profiling the system to be able to select the best combination possible.

The server's multitasking model is based on Java threads. From the operating system point of view, there is a single large server process running one thread for each component. We are considering an alternative implementation based on a different process for each component. In this way, the operating system will take care of all tasks related to process management and scheduling. Nevertheless, it is not completely clear for us at this moment how this will affect the overall system performance.

Presently, we are working on new software components to enhance the functionality provided by the framework. Particularly, we are studying ways for SMIL[36] to allow different SMIL documents in different browsers to synchronize. Regarding our educational application, we are currently implementing the software bridge between a microprocessors laboratory, with real hardware devices, and the CORBA ORB, which will adopt the collaborative environment provided by the current available framework in the evaluation.

## 6. ACKNOWLEDGEMENTS

The authors want to thank the European Union for the financial support through projects: FEDER 1FD97-0100 and FEDER 1FD97-0282, Fundación Caixa Galicia - Claudio San Martín and more than 550 voluntary students who used our educational platform, especially to the 180 who participated actively.

## 7. ADDITIONAL AUTHORS

M. Caeiro Rodríguez, (email: [mcaeiro@ait.uvigo.es](mailto:mcaeiro@ait.uvigo.es)), J. Santos Gago, (email: [jsgago@ait.uvigo.es](mailto:jsgago@ait.uvigo.es)) and J. Rodríguez Estévez, (email: [jestevez@ait.uvigo.es](mailto:jestevez@ait.uvigo.es)), Universidade de Vigo

## 8. REFERENCES

- [1] AICC. Aviation Industrie CBT Committee. WWW site. <http://www.aicc.org>.
- [2] AICC. CMI Guidelines for Interoperability, version 3.0.2. WWW site. <http://www.aicc.org/docs/tech/cmi001v301rtf.zip>.
- [3] ARIADNE. The Alliance of Remote Instructional Authoring and Distribution Networks for Europe. WWW site. <http://ariadne.unil.ch>.
- [4] Asymetrix. Toolbookii. WWW Site. <http://www.asymetrix.com/products>.
- [5] L. A. C. B. Aktan, C.A. Bohus and M. Shor. Distance learning applied to control engineering laboratories. *IEEE Transactions on Education*, 39(3):320–326, 1996.
- [6] CEN. European Committee for Standardization (CEN), Information Society Standardization Systems (ISSS), Learning Technologies Workshop (LT). WWW site. <http://www.cenorm.be/iss/Workshop/lt/>.
- [7] J. Dennins and E. V. Horn. Programming semantics for multiprogrammed computations. *Communications of the ACM*, 9(3):143–155, 1966.
- [8] G. Fernández. *Conceptos básicos de Arquitectura y Sistemas Operativos. Curso de Ordenadores*. Sistemas y Servicios de Comunicacin S.L, 1995. ISBN 84-605-0522-7.
- [9] S. Fisher and E. Michielsen. Mathematica-assisted web-based antenna education. *IEEE Transactions on Education*, 41(4), 1998. CD-ROM issue.
- [10] S. Fredrickson. Untangling a tangled web: An overview of web-based instruction programs. *The Journal. Assesing Technology*, 26(11), June 1999.
- [11] D. P. G. Da Bormida and G. Donzellini. Methodologies and tools for learning digital electronics. *IEEE Transactions on Education*, 40(4), 1997. CD-ROM issue.
- [12] GEM. Gateway to Educational Materials. WWW site. <http://www.geminfo.org>.
- [13] GESTALT. Getting Educational System Talk Across Leading edge Technologies project. WWW site. <http://www.fdggroup.co.uk/gestalt>.
- [14] IEEE. Learning Technologies Standardization Committee. WWW site. <http://ltsc.ieee.org>.
- [15] IEEE. LOM Working Draft v.4. WWW site. [http://ltsc.ieee.org/doc/wg12/LOM\\_WD4.PDF](http://ltsc.ieee.org/doc/wg12/LOM_WD4.PDF).
- [16] IEEE. PAPI Specification, version 6. WWW site. <http://edutool.com/papi/>.
- [17] IMC. vcalendar Specification. WWW site. <http://www.imc.org/pdi/>.
- [18] IMS. The IMS Content & Packaging Specification, version 0.92. WWW site. <http://www.imsproject.org/content>.
- [19] IMS. The IMS Enterprise Specification, version 1.01. WWW site. <http://www.imsproject.org/enterprise>.
- [20] IMS. The IMS Question & Test Specification, version 1.0. WWW site. <http://www.imsproject.org/question>.
- [21] IMS. Instructional Management Systems Global Learning. WWW site. <http://www.imsproject.org>.
- [22] ISO. ISO-IEC JTC1 SC36 pages. WWW site. <http://www.jtc1sc36.org>.
- [23] Javasoft. Java. WWW site. <http://www.javasoft.com/beans/>.
- [24] W. Lim. *Managing Software Reuse: A Comprehensive Guide to Strategically Reengineering the Organization for Reusable Components*. Prentice-Hall, July 1998.
- [25] LOTUS. Learning Space. WWW Site. <http://www.lotus.com/products/learningspace.nsf>.
- [26] A. S. M. Chirico, F. Giudici and A. Scapolla. The real experiment execution approach to networking courseware. *IEEE Transactions on Education*, 40(4), 1997. CD-ROM issue.
- [27] MadDuck. Web Course in a Box. WWW Site. <http://www.madduck.com/wcbinfo/wcb.html>.
- [28] Microsoft. Components Object Model. WWW site. <http://www.microsoft.com/com/>.
- [29] S. Microsystems. Enterprise Java Beans. WWW site. <http://www.sun.com/products/ejb>.
- [30] S. Microsystems. Java Beans. WWW site. <http://www.sun.com/beans/spec.html>.
- [31] National Computer Security Center. *A guide to Understanding Discretionary Access Control in Trusted Systems*, 1987. Library No. S-228,576.
- [32] Object Management Group. CORBA 3.0. WWW site. <http://www.omg.org/news/pr98/compnent.html>.
- [33] U. D. of Defense. DoD Advanced Distributed Learning. WWW site. <http://www.adlnet.org>.
- [34] PROMETEUS. PROMoting Multimedia access to Education and Training in EUropean Society. WWW site. <http://prometeus.org>.
- [35] J. H. Saltzer and M. Schroeder. The protection of information in computer science. *Proceedings of the IEEE*, 63(9):1278–1308, 1975.
- [36] Synchronized Multimedia Working Group. *Synchronized Multimedia Integration Language (SMIL) 1.0 Specification*. W3C, Philipp Hoschka edition, June, 1998. <http://www.w3.org/TR/1998/REC-smil-19980615>.
- [37] W. Systems. Top Class. WWW Site. <http://www.wbtsystems.com/solutions/products.html>.
- [38] US Department of Defense. ADL's SCORM. WWW site. [http://www.adlnet.org/Scorm/scorm\\_index.cfm](http://www.adlnet.org/Scorm/scorm_index.cfm).
- [39] WebCT. WWW Site. <http://www.webct.com/webct>.
- [40] M. Wilkes. *Time-Sharing Computer Systems*. American Elsevier, New York, 2nd edition, 1972.