Advanced Concepts for Interactive Learning

Aurel Vlaicu, Sanda Porumb, Cosmin Porumb, Bogdan Orza
Centre for Multimedia Technologies and Distance Education
Technical University of Cluj-Napoca
15 Daicoviciu, 400020 Cluj-Napoca
ROMANIA
http://icar.utcluj.ro

Abstract: - The most important advantages the information technology provides in the Open and Distance Learning area are dedicated not only to the Virtual Universities but also to the *Special Education Programmes*. One of the situations can be reached in the govern institutions where the sacked persons want to find a new job in other domain and they need a lot of training. This article describes new concepts used during the development process of the interactive learning services based on a set of applications designed to provide an infrastructure for webcasting, web conferencing capabilities, synchronized video presentation and automated creation of navigable and searchable multimedia archives.

Key-Words: - Advanced technologies, eLearning, interactive learning, collaborative environments, open architecture.

1 Introduction

The most important objectives of any eLearning provider are related to the interactive courseware for a wide range of users. Our approach, considered a top in this area, is responsive to learners, allowing them to actively participate in the educational process. It provides web access, for both of professors/tutors and students. This way, the course authors can elaborate the lessons and webminars by importing the actual educational materials stored in different formats (PowerPoint, Word Document, Rich Text Format, Text format, etc.) and the tool will automatically convert them in SWF, HTML, or JPG formats. Once the educational material is stored on the media server, the course author can edit the course using the video recorder capabilities elaborating the explanations. The virtual classware also allows the tutor to present the webminars, based on the same approach – the webminar content consists of the educational part (the PPT/DOC/RTF converted in SWF/HTML/JPG) and the video-based explanations (the recorded video).

The students access the web site and participate to the scheduled live presentations or load the recorded lessons.

2 Problem Formulation

Two types of use case scenarios will be provided. The first describes the actions the professors and tutors perform during a session. The other one is dedicated to the student, the consumer that many times pays for the service.

2.1 Providers

The professor section is protected and the authorized people have to use a login name and a password when accessing it.

After logging in, the professor is able to perform a set of actions described just bellow. Some of them refer how to elaborate lessons and webminars and others how to present the materials. The course authors can elaborate the educational material by importing the actual educational materials stored in different formats such as Microsoft PowerPoint, Word Document, Rich Text Format, and the tool will automatically convert it in SWF, HTML, JPG because SWF, HTML or JPG can be easily integrated with the video-based explanations and presented to the student.

When uploading the educational materials to be automatically converted, the Open Office and Java Servlets capabilities will be used. The conversion output, the SWF files e.g., will be automatically stored on the media server. The PPT presentations, DOC or RTF files will be uploaded on the server, via HTTP POST, and a Java Servlet component retrieves the files and communicates to the Open Office, using the socket concept (port 8100), in order to convert the original materials in known formats to be imported in Flash Scenes.

Each professor or tutor can easily browse the existing course materials uploaded on the media server. The system allows the professor/tutor to browse the courses/webminars or to have course/webminars statistics that define the students' interest. The educational materials are stored on the media server in a file structure. Each course will have its folder, and the lessons will be stored into the course folder. Each lesson consists of a set of learning activities (theory, practice, webminar), several lessons complete a unit, a set of units complete a module and a set of modules will be a course.

The system also allows the professor to see the flow of each course in order to know if everything is in good order. If not, this professor is able to update the content.

The professor is able to record his/her explanations using the video recorder capabilities, in an advanced manner – the video explanations will be synchronized with the course presentation. This way, the course can be considered completed and the students see the educational material as an entertainment show.

The tutor has to be on-line during the educational process using a video conferencing tool, this way, the student and the tutor can communicate each other. The tutor controls the educational process using the tool capabilities. He/she also controls the communication session – when a student wants to talk he/she clicks on the button “RAISE HAND” and the tutor enables him/her to talk about (the tutor presses “ENABLE SPEAKER”).

The tutor explains some aspects of the course and has to detail the concept. He/she can load the whiteboard in order to design block diagrams and import files (pictures, documents, presentations, etc.) to be shared with the students.

2.1 Students

In a *Special Programme*, when the students prepare themselves for a new job, the educational system has to be user-friendly, easy-to-use and very suggestive. The transparency is also a very important factor. This kind of students avoids the complex and slow educational systems, not transparent and complicated, because of the situation they have.

The students access the eLearning service using the web browsers. When the web browser sends the request to the web server, the *Student Module* will be loaded in the browser window. This module is based on the *rich client* concept and provides media streaming features, flow control capabilities for the shared multimedia presentations and high degree of interactivity.

The courses will be provided in a multimedia format – PPT/DOC/PDF/HTML originals converted in SWF/JPG/HTML format and the video explanations are stored on the media server and will be loaded in the shared scenes when the educational process starts. This way, the students can see the course author presenting the course, lesson by lesson, and interact with the systems using the tool controls.

The webminars are also presented in a multimedia format – PPT/DOC/PDF/HTML originals converted in SWF/JPG/HTML format and the video explanations are stored on the media server and will be loaded in the shared scenes when the educational process starts. The students can see the webminar presentations and talk to the tutor using the web conferencing capabilities.

3 Problem Solution

3.1 System architecture

The project design involved several technologies used for implementing the functionalities. For instance, the Macromedia Flash Communication Server or Adobe Flash Media Server are used for handling the media content and the shared objects, the Sun Open Office SDK is the open source used for converting the original educational materials in SWF/JPG/HTML format and the Java Servlets components are used for implementing the interface between the Open Office and the media server (Flash Communication Server/Flash Media Server). In conclusion, the system architecture respects the three tier architecture of the enterprise systems and can be illustrated just bellow:

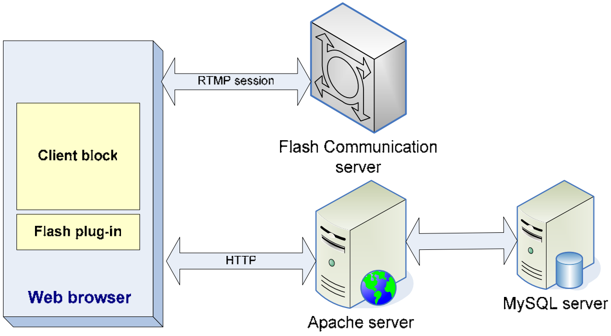


Fig. 1 - System Architecture

The client block is loaded in web browser and consists of the HTML documents dynamically generated by the JSP/PHP/Java Servlets components and the Flash scenes interpreted by the Flash Plug-in. The web server hosts the web components and the Flash Communication Server/Flash Media Server handle the media content. There are 2 protocols used for data transport – Flash Plug-in communicates to the Media Server via RTPM (RTMPT) protocol, and the web browser communicates to the web server via HTTP.

3.2 Upload and conversion tasks

The file upload and conversion tasks are implemented in a Servlet method, called *convertDocument(…)* that converts a document to a given type by using a running *OpenOffice.org* and saves the converted document on the media server. It expects the following parameters: the full path name of the file to be converted, the type to convert to and the full path name of the converted file.

The *Bootstrap* component provides functionalities to obtain a context or simply a service manager. The service manager will be used to create basic services.

```
XComponentContext xcomponentcontext =
com.sun.star.comp.helper.Bootstrap.createIn
itialComponentContext(null);
```

The *service manager* instance is created using the method *getServiceManager()* as in the code just bellow.

```
XMultiComponentFactory
xmulticomponentfactory =
xcomponentcontext.getServiceManager();
```

After creating the service manager, it's necessary to create an instance of the *UnoUrlResolver* component that supports the services specified by the *ComponentFactory* instace:

```
Object objectUrlResolver =
xmulticomponentfactory.createInstanceWithCo
ntext("com.sun.star.bridge.UnoUrlResolver",
xcomponentcontext );
```

When creating an instance of the *UnoUrlResolver* component that supports the services specified by the factory, the most important steps have to be respected:

- Create an instance of *XUnoUrlResolver*

```
XUnoUrlResolver xurlresolver =
(XUnoUrlResolver)
UnoRuntime.queryInterface(XUnoUrlResolver.
class, objectUrlResolver );
```

- Create a service manager from the initial object

```
xmulticomponentfactory=(XMultiComponentFac
tory)
UnoRuntime.queryInterface(XMultiComponentF
actory.class, objectInitial );
```

- Query for the *XPropertySet* interface

```
XPropertySet
xpropertysetMultiComponentFactory =
(XPropertySet ) UnoRuntime.queryInterface(
XPropertySet.class, xmulticomponentfactory
);
```

- Get the default context from the office server.

```
Object objectDefaultContext =
xpropertysetMultiComponentFactory.getProper
tyValue("DefaultContext" );
```

- Query for the *XComponentContext* interface

```
xcomponentcontext = (XComponentContext)
UnoRuntime.queryInterface(XComponentContex
t.class, objectDefaultContext );
```

Desktop is the *environment* for components that can instantiate within frames.

```
XComponentLoader xcomponentloader =
(XComponentLoader )
UnoRuntime.queryInterface(
XComponentLoader.class,
xmulticomponentfactory.createInstanceWithC
ontext(      "com.sun.star.frame.Desktop",
xcomponentcontext ) );
```

The upload consists of several steps detailed just bellow:

- Loading the selected document

```
Object objectDocumentToStore =
xcomponentloader.loadComponentFromURL(stri
ngUrl, "_blank", 0, propertyvalue);
```

- Store a document to a URL

```
XStorable xstorable=(XStorable)
UnoRuntime.queryInterface(
XStorable.class, objectDocumentToStore );
```

- Preparing properties for converting the document

```
propertyvalue = new PropertyValue[2];
propertyvalue[ 0 ] = new PropertyValue();
propertyvalue[ 0 ].Name = "Overwrite";
propertyvalue[ 0 ].Value = new
Boolean(true);
propertyvalue[ 1 ] = new PropertyValue();
propertyvalue[ 1 ].Name = "FilterName";
propertyvalue[ 1 ].Value =
stringConvertType;
```

- Appending the extension to the origin document name

```
stringConvertedFile = stringUrl + "." +
stringExtension;
```

- Storing and converting the document

```
xstorable.storeAsURL( stringConvertedFile,
propertyvalue );
```

- Getting the method dispose() for closing the document

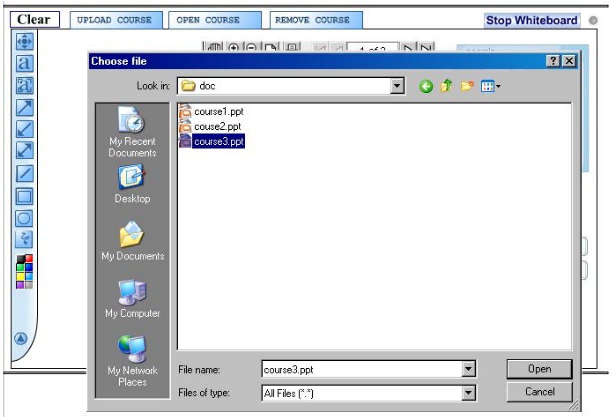
```
XComponent xcomponent = ( XComponent )
UnoRuntime.queryInterface(
XComponent.class, xstorable );
```

- Closing the converted document

```
xcomponent.dispose();
```

3.3 Professors and Tutors

The system allows the professor/tutor to browse the courses and webminars, or to access the statistics. The course author logs in and browses the course content using the *LoadCourse* component illustrated in the figure just bellow



course1.ppt
couse2.ppt
course3.ppt

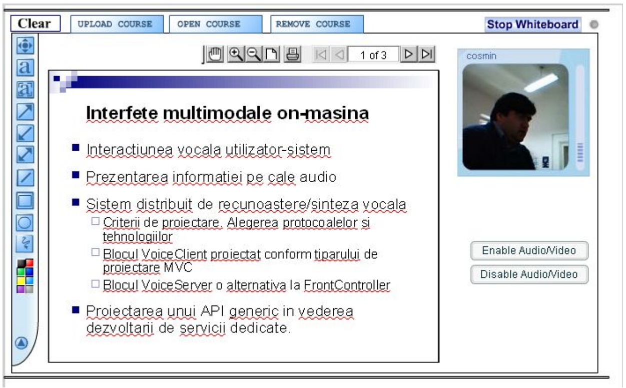
File name: course3.ppt

Files of type: All Files (*.*)

Fig. 2 – Browse courses and webminars

The author will be able to select and load the desired course in order to record the explanations or to edit it, if necessary. Technically, the courses are already converted in SWF/JPG/HTML format, for that reason, the *MainScene* component considers the SWF scenes or the JPG files as symbols that will be loaded in a synchronized manner.

Each explanation the course author records for completing the course materials will be stored in a FLV file on the media server. Macromedia/Adobe provides a platform independent approach for video recording functionalities, simple but with excellent results that needs to be considered one of the top concepts for the moment.



Interfete multimodale on-masina

- Interactiunea vocala utilizator-sistem
- Prezentarea informatiei pe cale audio
- Sistem distribuit de recunoastere/sinteza vocala
  - Criterii de proiectare. Alegerea protocoalelor si tehnologiilor
  - Blocul VoiceClient proiectat conform tiparului de proiectare MVC
  - Blocul VoiceServer o alternativa la FrontController
- Proiectarea unui API generic in vederea dezvoltarii de servicii dedicate.

Fig. 3 – Record explanations (professor)

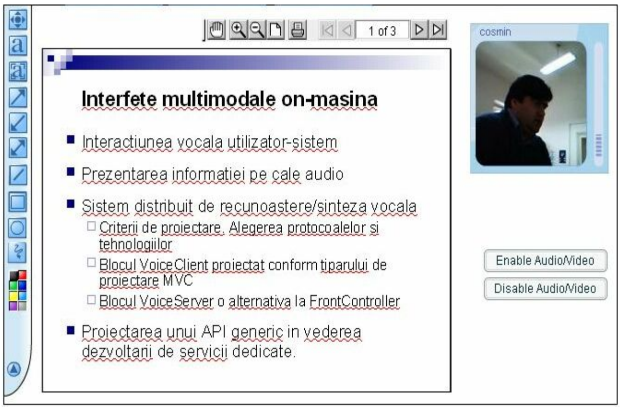
The tutor controls the webminars by using the tool capabilities. Actually, each visual component in the system can be considered as an object shared between the participants:

```
nc = new
NetConnection("rtmp://myHost/whiteboard/");
course_so =
SharedObject.getRemote("course_so", nc.uri,
true);
course_so.onSync = function(){
    d = course_so.data.course;
    course_txt.text = d;
    if (d!="" and d!="undefined"){
      wb_mc.loadFile_mc._visible = true;
wb_mc.loadFile_mc.loadMovie(“myfiles/”+doc_
txt.text); }}
course_so.connect(nc);
```

3.6 Access the courseware

The system must provide features for the individual study, when the student logs in and tries to learn a selected lesson or to study a selected webminar. This aspect is very important when talking about people that change the job or students in a special programme. For that reason, the student module is implemented in order to provide high degree of interactivity, to be easy-to-use, to have a friendly graphical user interface and to be very close to an Entertainment service.

The classware module provides individual and multiple access for the students in order to respect the most important rules of the eLearning concept: geographic, temporal and platform independence capabilities. When a student logs in for accessing the courses, he/she controls the system in order to load the course material. The course will be presented, lesson by lesson by the author that already recorded the explanations/descriptions using the video recorder and the student can interact with the educational material using the controls.



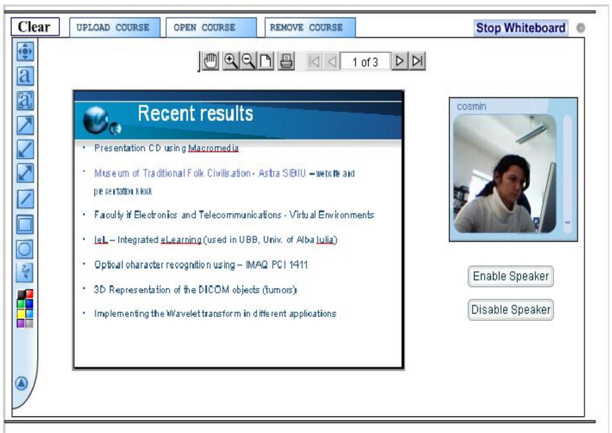
Interfete multimodale on-masina

- Interactiunea vocala utilizator-sistem
- Prezentarea informatiei pe cale audio
- Sistem distribuit de recunoastere/sinteza vocala
  - Criterii de proiectare. Alegerea protocoalelor si tehnologiilor
  - Blocul VoiceClient proiectat conform tiparului de proiectare MVC
  - Blocul VoiceServer o alternativa la FrontController
- Proiectarea unui API generic in vederea dezvoltarii de servicii dedicate.

Fig. 4 – Individual study

The *webminar module* allows the tutor to load and present interesting theoretical and practical subjects to the students. This module provides two levels of access – the tutor that needs to present the educational material and the students that takes part to the presentation and also to the discussions about the subjects presented. Technically, the shared objects concept is used when designing the rich user interface for both of student and tutor/professor modules. This way, the tutor gets the control in a webminar session according to the educational rules: he/she controls the conferencing session, the webminar presentation and also the whiteboard usage.

The students can have their opinion and the module allows them to ask/talk about the subjects as in a normal classroom: when a student has a question or wants to give a personal answer, he/she has to press the button RAISE HAND in order to be in the list and the tutor enables the speaker by using the button ENABLE SPEAKER as in the figure just above.



Recent results

- Presentation CD using Macromedia
- Museum of Traditional Folk Civilisation - Astra SIBIU – website and presentation kiosk
- Faculty if Electronics and Telecommunications - Virtual Environments
- IeL – Integrated eLearning (used in UBB, Univ. of Alba Iulia)
- Optical character recognition using – IMAQ PCI 1411
- 3D Representation of the DICOM objects (tumors)
- Implementing the Wavelet transform in different applications

Fig. 5 – Tutor controlling the webminar

The tutor uses the whiteboard in order to explain more about the subjects of the webminar, by designing block diagrams or importing files (pictures, documents, other presentations) to be shared with the students in a webminar session.

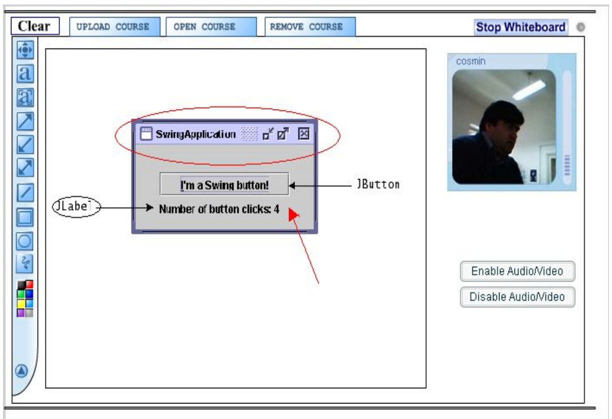


Fig. 6 – Tutor in the whiteboard session

The whiteboard also allows the tutor and students to locally save the visual surface of the whiteboard as picture (JPG format).

4 Conclusion

Current personal computers, workstations and servers are designed to handle traditional forms of data. Their performance is optimized for a scientific or transaction-oriented type of workload. Those systems do not perform well for multimedia data, requiring fast data retrieval and guaranteed real-time capabilities. The I/O capacity is usually a severe bottleneck.

The testing session allows us to see if the system works properly and how the most important parameters respect the indicated values. Those parameters are described just bellow:

Minimal response time is a crucial factor for the success of multimedia services and can be defined as the response time the client sees when using the system. The system must be able to minimize response times to live up to the expectations of the user.

Fast processing capability and low data access rates: the client requests should be processed fast and data access rates should be minimized in order to guarantee fast response times.

Reliability and availability: like any other kind of services, an eLearning service must be reliable. The number of users and volume of data is handled by the server component; more difficult it is when providing reliability. A special software mechanism must be employed in order to provide fault tolerance. Since client requests may arrive at any time, the time the server is unavailable should be minimized.

Ability to sustain guaranteed number of streams consists of a maximum number of client concurrent connections to the eLearning service (the data streams the server can simultaneously handle). Flash Communication Server/Media Server guarantee concurrent connections depending on the product license.

Real-time delivery: a web-based interactive system that provides both of video conferencing capabilities and multimedia presentation is highlighted in that article. The media server needs to be able to guarantee real-time delivery for individual streams as well as for all the streams combined together. The system is developed for accurate real-time operating.

Quality of Service (QoS) requirements: The Quality of Service (QoS) is a set of parameters describing the tolerable end-to-end delay, throughput, and the level of reliability in multimedia communication and presentation. QoS requirements of clients are an important factor that affects the usage of the service. The server should be able to provide and adapt itself to different QoS requirements, according to the characteristics of the client's terminal, the network connection and the requested data type.

Cost effectiveness: A very important requirement governing the future of the interactive learning systems is the cost effectiveness. Now, when the video conferencing tools are improving the educational process, the technology providers changed the target focusing on the narrow bandwidth Internet connections and basic peripherals (cheap web cams and microphones). This means low cost for both of service providers and students.

REFERENCES

[1] Client-Side Manager for Multimedia Distributed Services, Aurel Vlaicu & co, Bucharest, October 2004

[2] Intelligent and Personalized Graphical User Interface in e/mCommerce Systems, Aurel Vlaicu, Cosmin Porumb, Sanda Porumb & co, Salonic, May 2004

[3] Research activity related to the multimedia systems (Ph.D. Stuff), Cosmin Porumb, Cluj-Napoca, December 2003

[4] Technologies and Methods used in IT distributed systems (Ph.D. Stuff), Cosmin Porumb, Cluj-Napoca, September 2003

[5] Management and Assessment/Student Evaluation Tools for Open Virtual Educational Environments, Aurel Vlaicu & co, TEMPUS Programme, Tg Mures, 15.03.2001

[6] Intelligent Communication Environment using multiagent system, Aurel Vlaicu & co, COST #276, Florence, March 2002

[7] Information Management using Multi-Agent Platform Systems, Aurel Vlaicu & co, COST #276“, Budapest, October 2002

[8] Integrated Environment for distance Learning, Aurel Vlaicu & co, COST #276, Budapest, October 2002

[9] Integrated Environment for Distance Learning – IeL, Aurel Vlaicu & co, 2003 EDEN Annual Conference, Rhodes

[10] Programming Macromedia Flash Communication Server, Brian Lesser, Giacomo Guilizzoni, Joey Lott, Robert Reinhardt, Justin Watkins, Editura O'Reilly, ISBN 0596005040, Februarie 2005

[11] Voice User Interface Design, M. H. Cohen, J. P. Giangola, J. Balogh, Editura Addison Wesley, ISBN: 0321185765, 2004

Director 7 Demystified, Jason Roberts, Phil Gross, Editura Macromedia Press, ISBN 0201354454, 2004

[12] Reality J2EE: Architecting for Flash, Steven Webster, Editura Macromedia Press, ISBN 0321158849, Aprilie 2003

[13] Shareable Content Object Reference Model (SCORM), Editura Faulkner Information Services, ISBN 0358003080, 2003

[14] Intelligent virtual agents for contact center automation, M. Gilbert, J. G. Wilpon, B. Stern, G. Di Fabrizio, IEEE Signal Processing Magazine, Septembrie 2005

[15] *Impatica Web Site*: http://www.impatica.com/