

# MULTIMEDIA TECHNOLOGY INVOLVED IN E-LEARNING PLATFORM GENERATORS

Paulina Mitrea\*, Ovidiu Buza\*, Delia Mitrea\*

*\*Technical University Cluj-Napoca- Computer Science Department*

**Abstract:** Our paper presents an eLearning portal generator environment with powerful MultiMedia facilities, consisting in: advanced multimedia objects handling techniques, synchronization procedures, graphic object oriented data structures, image processing elements (based on texture analysis), moving objects extraction from the video frames, etc. First we present the general design of the e-learning site generator environment and the general scheme of the major .jsp pages, after being detailed some multimedia object handling techniques, as well as the mathematical background used in the synchronization between the moving images and the audio-video sequences.

**Keywords:** eLearning portal, MultiMedia technology, broadband connection, adaptive algorithm, embedded objects, synchronization, script file, tree datastructure, control kernel, histogram.

## INTRODUCTION

In order to obtain fully satisfying results in implementing a complete eLearning solution, the usage of the most appropriate MultiMedia technology is very important (*Webref7, Webref8*).

If the method to use the CD as physical information support is easy enough -because it suppose only the usage of some very well known utilities, a complete eLearning solution, consisting in a professional eLearning portal involve a completely different technology, as follows:

The eLearning software platform, which facilitates the entire course creation process, presentation, administration, involve:

- ❑ The Operating System, powerful database server
- ❑ The hardware (server) hosting both the Software Platform and the Database
- ❑ Internet broadband connection - necessary to assure the on-line communication
- ❑ The individual PC-stations, also connected to the Internet

### 1. THE STRUCTURE OF THE MULTIMEDIA E-LEARNING PLATFORM GENERATOR

The MultiMedia eLearning platform generator consists in 3 modules, that are in a continuouse interaction (*Webref9*):

(1.1) *Audio-Video Module*

(1.2) *Lesson Presentation and Management Module*

(1.3) *GraphicModule*

*1.1 The Audio-Video Module* consists from 3 submodules :

#### -Audio-Video signal acquisition component

Has the role to acquire the audio-video signal from an external device (video camera, video player,etc.), also to convert them in standard digital audio-video format by means of a specialized acquisition card (Microsoft Corporation, 2000, *Microsoft Windows: Programmer's Reference*); the aim is to include audio/video sequences in the context of the on-line lessons.

#### -Submodul destined to code the signal into files

This component use the functionalities of Windows Media Encoder to encode the AVI video file in formats that are characteristic for the transmtion of multimedia data on the Internet, like Data Stream. These formats are WMV, WMA and ASF (*Webref1, Webref2, Webref3, Swan, T., 1993*).

-Storage component - this module store the data that are previously codified, in a librarie organized on the hard-disk

#### *1.2 Lesson Presentation and Management Module*

-Submodule for the management of graphic projects of the on-line lessons: supervise the organization of graphic presentations from the context of the on-line lesson in so called *graphic projects*. Their structure is of tree form, each node of the tree being a certain graphic presentation, having associated an appropriate database.

-Submodule for the management of the on-line lesson presentation: supervise and assist the manner to build graphic presentations based on some HTML

templates, as well as movie sequences in stream format. The HTML template gives the graphic layer of the presentation, hosting the movies, the static images and the animations.

**- The Control Kernel:**

It is the main module of the application, that controls all the other modules, and coordinates the entire lessons presentation building process (McIntyre, D, 1992), beginning from the acquisition and coding of the movies and finishing with the integration of the graphic components of the application.

**-Export component:**

After the lessons are built and integrated in projects, they may be stored on the hard-disk or on external storage devices like CD or DVD.

**-Publishing submodule (on Internet).**

The project is published on a www server. The publishing manner may be automatic or half-automatic, function of the capabilities of the server to accept the remote update of data.

1.1 Graphic Module is charged to effectively build the graphic presentations. We have here 3 sub-modules :

**(1.1)1.Template management submodule**

-provide predefined graphic presentation templates to the user; provides also a template editor, to enable the creation of own graphic presentation templates.

**(1.1).2 Synchronization submodule**

It's a basic component of the application, that enable to synchronise a movie introduced in a lesson with the corresponding graphic objects; these graphic objects may be: texts, static images, animations. Other type of synchronization will be made between the moving images and the audio sequences, that will be described in details in the following section.

**(1.1).3 Visualizing submodule (preview).**

In the final of the presentation design process, the entire presentation will be immediately visualized in a window, similar with the Internet browser (ex. Internet Explorer). The user will be able to see the result of any combination between a certain template, a movie and the graphic objects laying in the page.

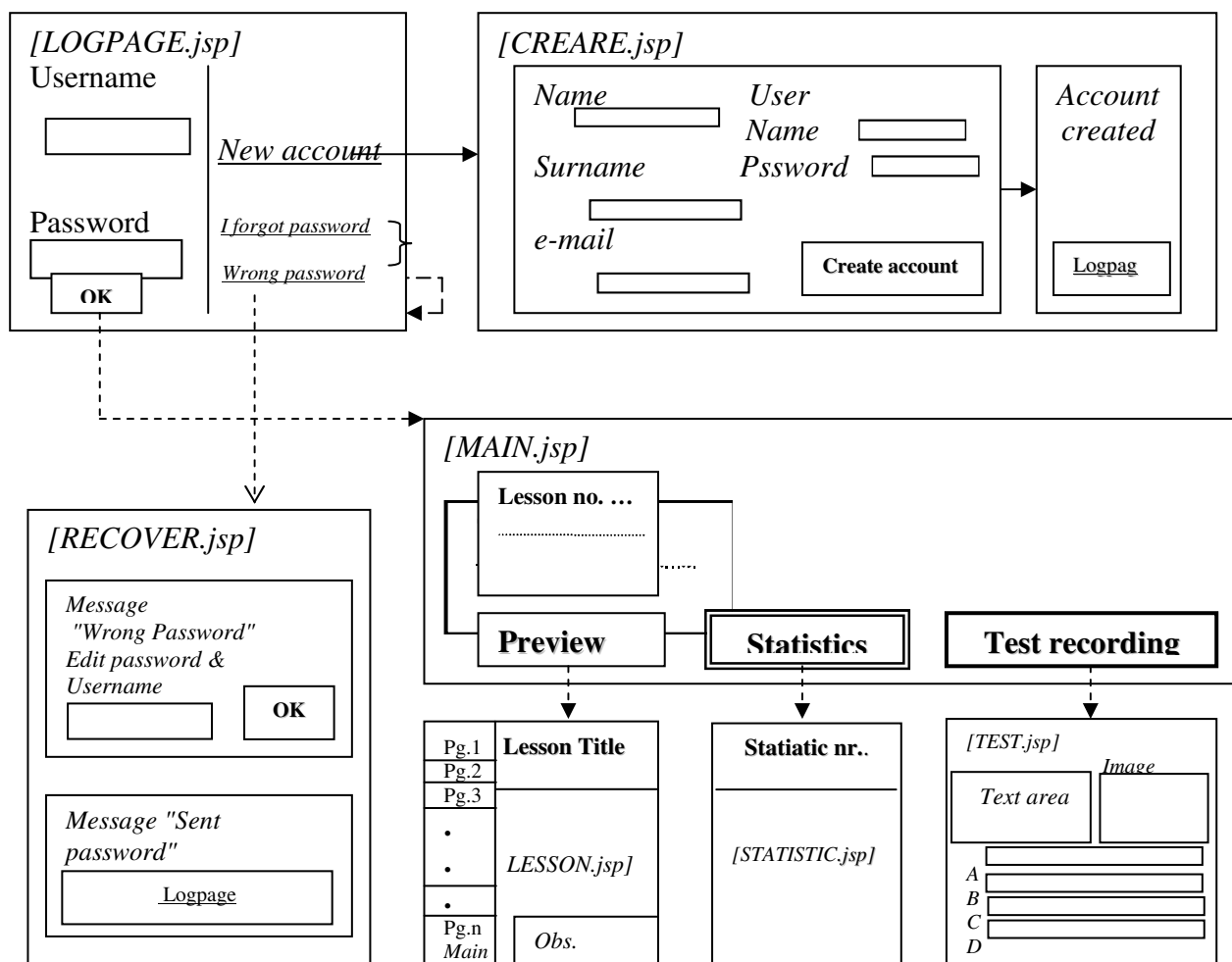


Fig. 1 Jsp pages of the software implementation

## 2. SYNCHRONIZATION TECHNIQUES FOR HANDLING MULTIMEDIA OBJECTS

### 2.1 The Synchronization between the Graphic Elements and the Movie

The eLearning site generator is provided with a set of robust procedures (routines) aimed to synchronize the movies - integrated in the context of a web page - with the graphic elements. The procedures were implemented based on Windows Media technology, which suppose to include a Windows Media Player control in the context of the graphic page (Microsoft Corporation, 2000, *Microsoft Windows: Programming Tools*).

There exists two ways to integrate the movie sequence in the context of the web page:

- (a) in the context of a pop-up window;
- (b) embedded in the context of the page.

The simplest way is the first one /solution (a)/ enabling the user to have access to all the facilities provided by Windows Media Player: control buttons, selection menus (Microsoft Corporation, 2000, *Microsoft Windows A Guide to Programming*).

The movies included in the page as embedded objects, are specified in the context of the HTML code in a similar way as the images. The associated buttons like „Play”, „Pause”, „Stop”, „Forward”, „End” etc, are also specified in the context of the HTML code.

Three type of procedures were created (Webref4, Webref5, Webref6), corresponding to the synchronization with three type of graphic objects, as follows:

- (1) images
- (2) animations
- (3) text.

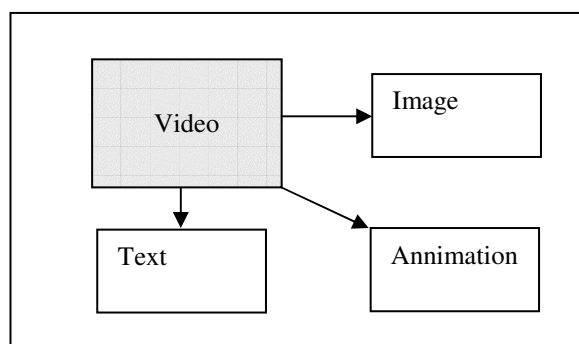


Fig.2 Scheme of the three type of synchronizations between the video sequences and the graphic objects

The synchronization is made based on Windows Media technology, which enable to include the synchronization information inside the video stream. That is made by means of an ASCII type script file, containing the synchronization information as short description in text form. This file is copied in the header of the video stream and will generate script

based events of the browser at each distinct moment of the synchronization.

The events generated in this way are used to display (to show) or to hide the objects from the client application's page, to move these objects etc.

### 2.2 The synchronization between moving images and the audio sequences, based on texture analysis

The frames of the moving images will be analysed based on their texture structure. The principles and the mathematical background will be presented in that follows.

The kernel of the processing oriented on this goal consists in a texture based method for detecting moving objects (Marko Heikkilä, 2002).

The moving object detection, also called as *background subtraction*, is one of the most critical components in many machine vision applications, especially in proactive ones. The goal is to extract the moving objects from the video frames captured from a camera.

The background subtraction can be seen as a two-stage process:

(2.1) *Foreground detection*

(2.2) *Background modeling*

(2.1) *Foreground Detection* consists mainly in the comparison of the input frame with the current background model. The areas of the input frame that do not fit to the background model are considered as foreground.

(2.2) *Background Modeling* is the most important part of any background subtraction algorithm. The goal is to construct and maintain a statistical representation of the scene that the camera sees.

Instead of just using the pixel intensity, it is recommended to utilize image texture when modeling the background. Also is recommended to use the *Local Binary Patterns (LBP) algorithm*, as texture measure, because of its good properties:

- Invariant to monotonic changes in gray scale, which makes it robust against illumination changes.
- Computationally very fast.

The algorithm operates under the assumption of a stationary camera with fixed focal length.

The description of the algorithm is as follows:

The background is divided into blocks. We suppose that they are partially overlapping.

Each block is modeled as a group of  $K$  *weighted LBP histograms*:

$$\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K\}.$$

The background model is updated with the information of each new video frame, which makes the algorithm adaptive.

The update procedure is identical for each block.

The steps of the algorithm are:

*Step 1.* Calculate a LBP histogram  $\mathbf{x}_t$  for the block of the new video frame.

*Step 2.* Compare the new block histogram  $\mathbf{x}_t$  against the existing  $K$  model histograms  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K\}$  by using the *histogram intersections* ( to obtains the distance measure).

(a). If none of the model histograms is close enough to the new histogram, the model histogram with the lowest weight is replaced with the new histogram and is given a low initial weight.

(b) If a model histogram close enough to the new histogram was found, the bins of this histogram are updated as follows:

$$x_{k',t}[i] = \alpha_b \lambda_t[i] + (1 - \alpha_b) x_{k',t-1}[i], \alpha_b \in [0,1]$$

Furthermore, the weights of the model histograms,  $w_{k,t}$ , are updated as follows:

$$w_{k,t} = (1 - \alpha_w) w_{k,t} - 1 + \alpha_w M_{k,t}, \alpha_w \in [0,1]$$

where  $\alpha_w$  is the learning rate, user defined,  $M_{k,t}$  has the value 1 for the histogram  $x_k$  and 0 for the other cases.

*Step3.* Next, we need to decide which of the histograms of the model are most likely produced by the background processes. We use the persistence of the histogram as an evidence for this. Because the persistence of the  $k$  th model histogram is directly related to its weight, the model histograms are sorted in decreasing order according to their weights.

As a result the most probable background histograms are on the top of the list.

*Step4.* As a last phase of the updating procedure, the first  $B$  model histograms are selected to be the background model as follows:

$$w_{1,t} + \dots + w_{B,t} > T_B, T_B \in [0,1]$$

Foreground detection is achieved via comparison of the new block histogram  $\mathbf{x}_t$  against the existing  $B$  background histograms  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_B\}$  selected at the previous time instant. If a match is not found, the block is considered to belong to the foreground. Otherwise, the block is marked as background.

The performance of the proposed algorithm was evaluated using several test sequences. Both indoor and outdoor videos were included. The algorithm has also real-time performance, obviously very important in eLearning.

## CONCLUSION

Involving in our approach both advanced MultiMedia techniques and a strong mathematical background in the most sensitive components of the eLearning generator environment, we are able to provide a powerful software tool, as basis to extend the eLearning capabilities not only in the educational institutes, but also in administration, business and industry.

## REFERENCES

- Webref1:<http://www.microsoft.com/Softlib/MSLFILE/S/ENMETA.EXE>  
 Webref2:<http://www.microsoft.com/Softlib/MSLFILE/S/METAFILE.EXE>  
 Webref3:<http://www.microsoft.com/Softlib/MSLFILE/S/PLAYMETA.EXE>  
 Webref4:<http://www.microsoft.com/developr/MSDN/OctCD/EMFDCO.ZIP>  
 Webref5:<http://www.microsoft.com/developr/MSDN/OctCD/ENHMET.ZIP>  
 Webref6:<http://www.microsoft.com/developr/MSDN/OctCD/METAFLI.ZIP>  
 Webref7:<http://science.coe.uwf.edu/NARST/NARST.html>  
 Webref8:<http://science.coe.uwf.edu/NSELA/nSELA.html>  
 Webref9:<http://www.enmu.edu/cetp>  
 Marko Heikkilä (2002), **University of Oulu - Machine Vision Group FINLAND: A texture based method for detecting moving objects**  
 Microsoft Corporation, (2000), *Microsoft Windows: Programmer's Reference, Volume 4: Resources*, Microsoft Windows Programmer's Reference Library, Microsoft Press, Redmond, WA.  
 Microsoft Corporation, (2000), *Microsoft Windows: Programming Tools, Microsoft Windows Programmer's Reference Library*, Microsoft Press, Redmond, WA.  
 Microsoft Corporation, (2000), *Microsoft Windows A Guide to Programming, Microsoft Windows Programmer's Reference Library*, Microsoft Press, Redmond, WA.  
 Swan, Tom, (1993) *Inside Windows File Formats*, **Sams Publishing**, Indianapolis, IN, Microsoft Corporation