DEVELOPMENT AND IMPLEMENTATION OF A BIOMETRIC VERIFICATION SYSTEM FOR E-LEARNING PLATFORMS

Elisardo González-Agulla, Enrique Argones-Rúa, Carmen García-Mateo and Óscar W. Márquez-Flórez
Signal Processing Group, Signal Theory & Communications Dept, University of Vigo, Spain

Abstract: We describe a biometric verification system to improve the security and reliability in services offered by e-learning platforms. We developed a prototype in the platform ILIAS to improve its online identification service. We use our extension of VoiceXML called BioVXML [1]. BioVXML is focused on user biometric verification. It is able to handle heterogeneous biometric samples, such as voice samples or frontal face images and takes them into account in order to do a multimodal biometric verification. This prototype is used to acquire a biometric database, the BioDB. This database will serve us as a test environment for the different biometric verification methods.

Key words: VoiceXML Biometric Verification; Biometric Algorithms; Internet identification.

1. INTRODUCTION

E-learning has emerged in the recent times as an alternate solution to the traditional education. One of the main drawbacks of the e-learning is the difficulty to do an online user verification. The current e-learning platforms that provide online user identification implement methods that are not secure in order to prevent an impostor to supplant a user [2-5].

Our objective is to build an independent verification system based on processing biometric features such as voice or frontal face images. This system can be embedded in web applications and is intended to improve

security and reliability in the access to some web services. In particular we want to improve the e-learning platform ILIAS [5] developing a biometric verification module to avoid user supplantation.

The verification tasks are modeled as human-machine dialogues. The main contribution of this work is the way these dialogues can be specified. They are defined in an extension of the markup language VoiceXML. This extension provides multimodal verification support to the system, and we decided to name it as **BioVXML.** BioVXML enables us to use different types of biometric features, such as voice or frontal face images, and an easy way to incorporate new biometric algorithms and modalities. BioVXML separates the application development from biometric algorithmic development, so the application developers do not need a background on biometrics to incorporate them to its client/server application.

The behaviour of the system is specified in the dialogues stored in the BioVXML files associated to the enrollment task and the verification task. Each BioVXML file defines an interaction between the user and the client of the verification system. This interaction is implemented by a graphical interface. The user collaborates in the biometric features acquisition process following the instructions of the application.

The verification system reports the e-learning platform on the verification process result. The e-learning platform calls the verification system for user authentication, and it must handle the results from the verification process.

2. DESCRIPTION OF THE BIOMETRIC VERIFICATION SYSTEM

Below we describe the environment where the verification system must be integrated, how the human-machine dialogues are built, the structure of the system, its architecture and its implementation.

2.1 The environment

The environment we use to test the feasibility of BioVXML is the e-learning platform ILIAS. This e-learning platform uses the traditional login and password in order to gain access to its different tools. Our goal is to improve these techniques using biometric procedures to tighten the e-learning platform security up. This improvement is achieved by using different authentication algorithms based on face and speech. User-tracking must be carried out by the system in order to avoid dynamic supplantations.

Also, a report file must be stored to deal with possible problems detected during the session.

2.2 Design of the verification dialogue system

In the early design phase of the verification system some alternatives were analyzed in order to get an appropriate model for the verification process. Standards such as BioAPI, VoiceXML and X+V [6-9] have been studied, but we have finally opted for making an extension of VoiceXML 1.0 due to this markup language provides flexibility and an adequate client/server approach.

VoiceXML was originally intended for applications where the access is usually done by telephone and it just manages audio data. Our application is over an IP network and must manage multiple types of multimedia data. In order to take these aspects into account the VoiceXML must be extended appropriately.

2.3 BioVXML specification

We need a simple way to define human-computer dialogues, but these dialogues must be able to make all the biometric identity verification tasks, so we must extend VoiceXML to get our objective. BioVXML [1] must be able to handle the different kinds of biometric data, such as voice and face images. Therefore we added the new tags **enroll** and **verify** to the *VoiceXML.dtd,* and we modified the tag **record.** We named the DTD obtained so as *BioVXML.dtd.* Below we detail the functionality and syntax of the new or modified tags.

- Tag **<record>:** this tag is used for recording audio in VoiceXML. We added the new attribute **src** to it. It indicates the kind of biometric data to be recorded. Our parser BioVXML allows *src* to have two different values, *voice* and *face,* corresponding to audio samples or frontal face images. The parser could implement in a future any biometric value, just adding the associated functionality to the parser. If the attribute *src* is omitted, we will regard it as an audio source, keeping compatibility with VoiceXML.
- Tag **<enroll>:** this new tag is used to make the enrollment task. Its attributes are **name,** name of the variable associated with the result of the enrollment; and **type,** this is used to distinguish the different enrollment algorithms. The biometric data and configurable parameters for the methods are added like BioVXML parameters.
- Tag **<verify>:** this new tag is used to make the biometric identity verification. It has the same attributes than the tag enroll (**name** and

type), with the same meaning. The biometric data and configurable parameters for the methods are also added like BioVXML parameters.

A parser BioVXML was developed to interpret the BioVXML documents. These documents must abide by the BioVXML.dtd specified.

3. ARCHITECTURE AND STRUCTURE OF THE SYSTEM

The biometric verification system is a web application with client/server architecture. It provides identity verification services through man-machine dialogues defined through BioVXML documents. This verification system can be used by E-learning platforms to increase security in user accesses. Figure 1 shows the modular architecture of the verification system. Three main parts can be distinguished: The client application, the server application and the biometric application. Next, we describe the functionality of each of these parts:

The client application must handle the multimedia devices, such as the microphone and the webcam, to acquire biometric data, and send these features to the server. The application must show an appropriate graphical user interface, monitor the users, it sends reports and, eventually, a video/audio flow to the server.

Inside the server, we can distinguish several independent modules. One of them, the so-called Biometric Application embeds different multimodal algorithms for biometric user authentication. This module has an own database where templates, logs, scores and enrolment data (wav and jpeg files) are stored. The module constitutes an independent part of the system, offering a HTTPS access interface, so that different applications can access it. The other main module inside the server, the Server Application, constitutes the kernel of the system, and it offers two access interfaces, one of them towards the web platform in which it is embedded. This way, the web platform is able to:

- Setup the verification system.
- Choose the authentication mode.
- Request a report of the user behaviour during the session, etc.

The other interface is used for communication with the different users, who must be verified by the system.

The kernel of this application manages the Web Platform requests, the client incidences detected and the user verification process. Events happened during this process are annotated in a BioVXML document. So, this file shows a human-machine dialogue, which can be divided into three main parts:

- Capturing samples.
- Invoking authentication algorithm.
- Reporting results.

The BioVXML Interpreter runs the file containing the appropriate authentication algorithm and it must deal with the following tasks:

- Request the necessary samples to the client application.
- Invoke the verification method specified in the BioVXML file (this file is owned by the Biometric Application).
- Report the result of the verification process to the client and to the web platform.

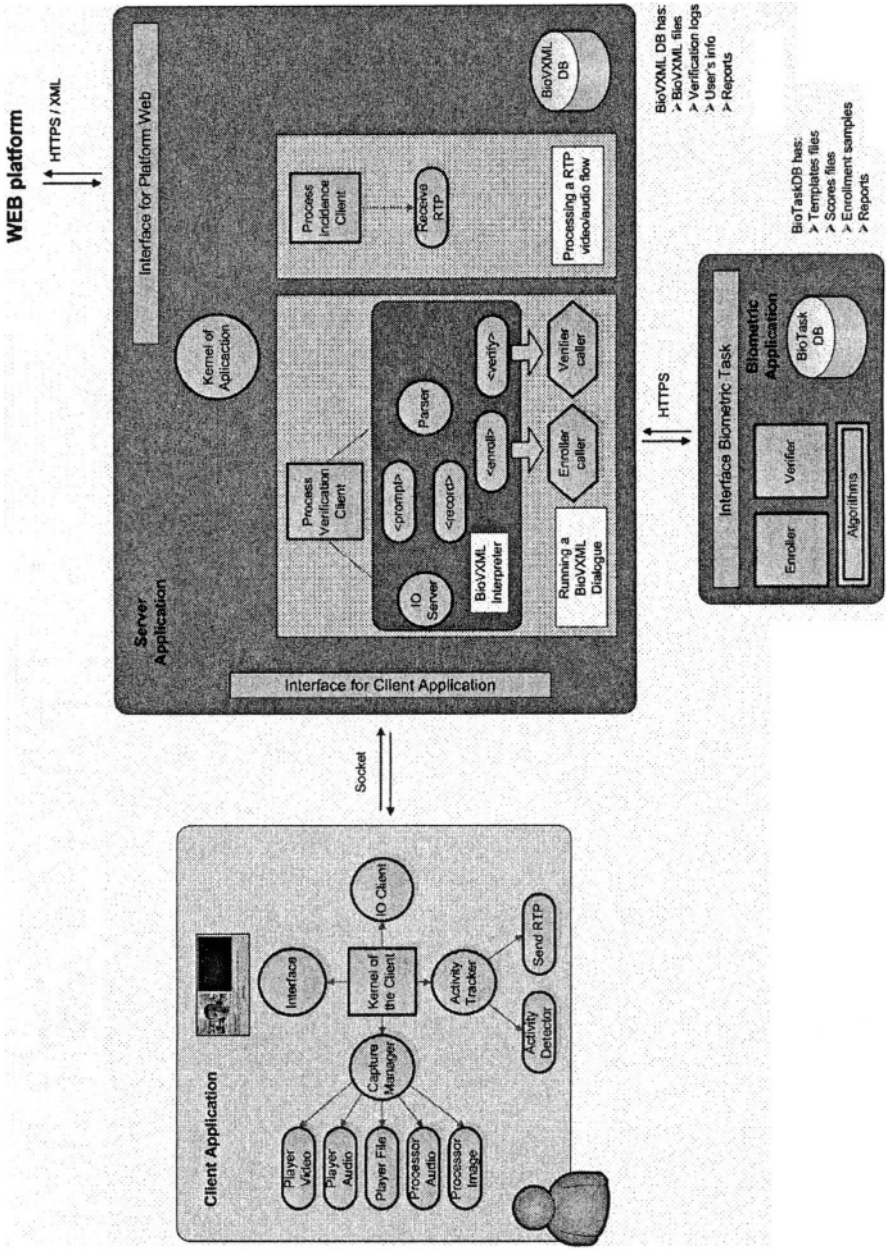

Figure 1. Architecture of the Biometric Verification System

4. IMPLEMENTATION OF THE VERIFICATION SYSTEM

In Figure 1 we show a structural scheme of the prototype we developed. Below we describe the implementation of its components:

4.1 Client description

We have considered the following requirements during the design of the client: Modularity; Multiplatform; Integration in the Web platform; Handling of multimedia devices and multimedia data; and easily to update.

These requirements made a **Java WebStart Application** [10] an appropriate solution because the users run always the last version of our client application. We must use the Java package **JMF** [11] in order to handle multimedia devices and multimedia data and its RTP features; and we use **Swing** in order to provide an adequate graphical user interface in multiplatform environment.

Our application consists of the packages **Client** and **Capture.** It is packed in the file **ClientApplication.jar** to minimize the number of connections to the HTTP server. The graphical interface, as example, is showed in Figure *2*.

Figure 2. Frontal face image acquisition interface

4.2 Server Application description

Our server manages the user verification queries from the Platform Web in an independent way. It makes the verification or enrollment task depending on what the Web Platform is asking for. It manages the possible events sent from the clients, reports them to the administrator and stores the associated video flow sent from the client.

We implemented a BioVXML **parser** in Java. Our **BioVXML Interpreter** uses this parser to work with BioVXML documents. This interpreter uses the different methods provided by the **Implementation Platform** in order to communicate with the clients, access to the BioVXML documents and communicate with the MySQL database.

The server is packed in the JAR file **ApplicationServer.jar** and the **class *ServerApplication*** is the kernel of the server. It manages the client requests and processes them in independent threads. When a verification user request is received from the Web Platform, the server application runs the corresponding BioVXML dialogue, and it sends the instructions to the client by a TCP/IP socket channel. When the application server receives the different biometric samples, it calls to the biometric application by a HTTPS channel to run the verification algorithms.

4.3 Biometric Application description

The biometric application is a servlet java. This servlet processes the verification queries. It must build the call to the different biometric verification, taking into account the claimed identity, the identification of the algorithm to be used and the biometric data needed. Once the algorithm returns the verification result, the servlet returns this result back to the server application.

This servlet uses a database to store data related to every verification session, the biometric enrollment data, the result files, the log files, etc.

5. INTEGRATION OF THE VERIFICATION SYSTEM IN ILIAS

The integration of the verification system in the web platform must be done in the user access control module. More precisely, after a user has been identified by the usual methods. ILIAS was designed according to the layer-based architecture shown in Figure *3* [12]. The verification module that was

developed must be inserted in the **ILIAS Core** layer, so that the authentication process of the system is strengthened.

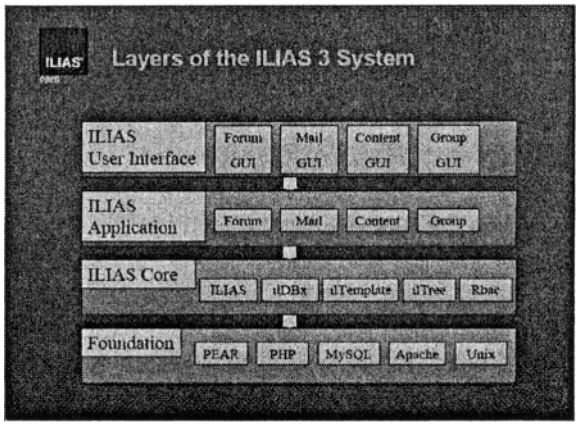

Figure 3. Layers of ILIAS

ILIAS was developed using PHP, it uses a MySQL database. The interface of the server application of the Web Platform offers us functions to do differents tasks such as user verification, functionality configuration of the verification system, selection of the different dialogues BioVXML to run, etc. With this interface the Web Platform invokes in a configurable way the user verification application, and then, this platform must interpret the verification result.

6. CONCLUSIONS AND FURTHER WORK

The verification system developed can be embedded inside any elearning platform due to the independence of its components. Every BioVXML document defines itself a biometric verification dialogue. The BioVXML specification enables us to separate the definition of the different biometric dialogues from the way these are implemented algorithmically. So we can modify the dialogue definition changing the number of biometric data to be recorded, the sentences to be read by the user and the instructions to show him without any change in the structure of the verification system, due to the flexibility provided by BioVXML. BioVXML facilitates the development of multiple verification applications over the same architecture.

Currently this system is still under development. It has just been installed in a web server and it works with a reduced number of users. However it is

soon to evaluate any quantitative results. In order to test different biometric algorithms we are constructing the biometric database BioDB. We use an enrollment BioVXML document to construct this database. With this database we plan to test multimodal biometric algorithms based on face recognition and speaker recognition in future works. Once we do this we will have quantitative real world performance results on biometric identity verification of our system.

On the other hand, the modular design of our verification system allow us to integrate it easily inside any Web Platform independently of its language of development (php, asp, jsp, etc).

ACKNOWLEDGEMENTS

This project has been partially supported by Spanish MCyT and Xunta de Galicia under the projects TIC2002-02208 and PGIDT02TIC32201PR respectively.

We would like to thank Daniel González Jiménez for his collaboration developing the BGM image verifier.

REFERENCES

1. Enrique Argones-Rúa, Elisardo González-Agulla, Carmen García-Mateo, Óscar W. Márquez-Flórez, User verification in a Bio-VXML framework. Odyssey 2004.
2. E-learning Platform WebCT <http://www.webct.com/>
3. E-learning Platform LearningSpace <http://www.lotus.com/products/learnspace.nsf/wdocs/homepage?opendocument>
4. E-learning Platform BlackBoard http://www.blackboard.com/
5. ILIAS open source <http://www.homer.ILIAS.uni-koeln.de/ILIASdoc/doc/html/1.html>2003.15 September.
6. BioAPI Consortium, BioAPI Specification Version 1.1 <http://www.bioapi.org>2003.15 September.
7. VoiceXML 1.0 <http://www.w3.org/TR/voicexml/> 2003.22 September.
8. VoiceXML Forum <http://www.voicexml.org>2003.24 September.
9. X+V 1.1 --- XHTML + Voice Profile <http://www.voicexml.org/specs/multimodal/x+v/11/> 2003.2 October.
10. Java Sun <http://java.sun.com/> 2003.10 October.
11. Java Media Framework API (JMF) <http://java.sun.com/products/java-media/jmf/> 2003.10 October.
12. "The ILIAS Architecture", <http://www.ilias.uni-koeln.de/conference/2003/pdf/WS1-ARCHITECTURE.pdf>