A Platform for Real-Time Multimodal Processing

Antonio Camurri*, Paolo Coletta*, Mirko Demurtas*, Massimiliano Peri*, Andrea Ricci*,Roberto Sagoleo*, Marzia Simonetti*, Giovanna Varni* and Gualtiero Volpe*

* InfoMus Lab-Laboratorio di Informatica Musicale, DIST-University of Genoa, Genova, Italy

{antonio.camurri, paolo.coletta, gualtiero.volpe}@unige.it

{demurtas, minou, max, a.ricci, sax, giovanna}@infomus.dist.unige.it

***Abstract* — We present an overview of the architecture and the main technical features of EyesWeb XMI (for eXtended Multimodal Interaction), a hardware and software platform for real-time multimodal processing of multiple data streams. This platform originates from the previous EyesWeb platform, but the XMI is not simply an updated version; it is the result of a 3-year work, concerninig a new conceptual model, design, and implementation. The new platform includes novel approaches integrating new kinds of interfaces (e.g. tangible acoustic interfaces) and a new set of tools for supporting research in multimodal architectures and interfaces. We discuss two case studies from two research projects at our Lab: the "Premio Paganini" Concert-Experiment and the Orchestra Explorer experiment on a new active listening paradigm. The real-time multimodal data streams processing is discussed. The experimental setups are used to clarify the model, technical issues involved in this new platform.**

***Keywords* — EyesWeb, Multimodal Processing, Multimodal Interactive Systems**

I. Introduction

Our research mainly focuses on exploring and modeling paradigms for non-verbal communication enabling a deeper, natural, and experience-centric approach of the user in human-computer communication.

In particular, our ongoing activities are finalized to design, develop and validate multimodal interfaces/systems based on extraction and analysis/synthesis of information related to high-level expressive content, including audio, video, and physiological signals. In this scenario, expressive gesture is a key concept. In our view, an expressive gesture is a gesture that conveys expressive content, that is information concerning aspects related to feelings, mood, affect, intensity of emotional experience [1]. In this perspective, a gesture is not just each "movement of body that contains information" [6]. In music performance, expressive gesture emerges from non trivial (multimodal) integration of the music signal (audio), the performer movement, the physiological signals. Dealing with expressive gesture therefore involves complex management and real-time processing of multimodal data streams.

A number of real-time platforms (e.g, PureData and Max/MSP) support users in their work on audiovisual streams, but most of them are designed and optimized for a single modality (e.g. either audio or video). Other approaches are focused on specific bimodal processing (e.g. face/lip and voice analysis). We need to face non-verbal multimodal signals and to work at different abstraction levels both in analysis and synthesis. This is a first main requirement that led us to conceive and develop the EyesWeb hardware and software platform [3]. This platform supports research on multimodal Human-Computer Interaction enabling the user to experiment with computational models of non-verbal expressive communication and to map, at different levels, gestures from different modalities (e.g., human full-body movement, music) onto real-time multimedia output (e.g., sound, music, visual media).

Other basic requirements include the need to integrate analysis coming from different sensorial modalities, the need to face and integrate novel paradigms for multimodal interaction (e.g. the tangible acoustic interfaces developed in the TAICHI EU Project - www.taichi.cf.ac.uk), and the need for computational models for analyzing different modalities to enable experiments on cross-modal techniques. These and other needs led us to reconceive and rebuild the EyesWeb architecture toward the version described in this paper [4].

Section II gives a brief overview of the EyesWeb XMI architecture and presents some its new main features. In Section III, two cases studies are used to clarify the potential of the platform in dealing with real-time multimodal processing in concrete music scenarios.

II. EyesWeb XMI: Architecture and Features

Starting from the requirements emerged from ongoing research on multimodal interactive systems and the feedback we received from the wide community of users, we proceeded to a deep revision process of the EyesWeb platform, which originated the EyesWeb 4 prototypes, and which finally led to the complete redesign and development of EyesWeb XMI (eXtended Multimodal Interaction).

The aim was to develop a development environment supporting a faster, easier, and more powerful and transparent modeling and implementation of multimodal interactive systems. This includes the support for experiments and applications for synchronized, multi-sensor data recording, analysis, and processing, based on the exploitation of commodity hardware and using non real-time Operating Systems (MS Windows based systems).

Another important feature of EyesWeb XMI is the portability toward other platform and in particular Linux and mobile systems platforms.

A. Architecture

Like in previous versions [2], the platform consists of two main components: a kernel and a graphical user interface (GUI).

The GUI manages interaction with the user and provides all the features needed to design patches. It is noteworthy that the kernel and the GUI are not tightly coupled: this means that several GUIs may be developed for different applications. As a matter of fact, a set of other simpler interfaces already exists and will be soon delivered on the web site (www.eyesweb.org).

The kernel is a dynamically pluggable component, which takes care of most of the tasks performed by EyesWeb XMI. Since XMI is designed to be extended with third party software modules (that we call *blocks*), the first task performed by the kernel is the registration and organization of such extensions into a coherent set of libraries. Further, the kernel takes care of guaranteeing persistence of the user designed applications (that we call *patches*), by providing save and load functionalities. Moreover, another task managed by the kernel is the mapping of physical devices into logical ones, to simplify porting of patches among different hardware configurations. Finally and most important, the kernel contains the EyesWeb execution engine, which manages the actual execution of patches, handles data flow and synchronization, handles notification of events to the user interface. The engine also provides a timestamping mechanism: any datatype generated during the patch execution is assigned a time code, obtained by the current value of the kernel clock. This time code is used to synchronize different data streams (e.g., one audio and one video streams, or two distinct audio streams) regardless of the delays introduced by the possibly different processing times.

Another tool distributed with the platform is the *EywPatchExecutor*, a simple scheduler of multiple patches which is very useful when a user needs to run a set of patches in sequence. For unsupervised setups, it is possible to use the *EywConsole* application, a command line-based interface for avoiding the usage of the GUI. Thanks to these tools and to the modular architecture of the EyesWeb software platform, it is also possible to develop custom interfaces, as we did for specific installations (e.g., museum installations).

Main Features of EyesWeb XMI

The new features introduced in EyesWeb XMI mainly concern (i) the possibility of connecting a large number of external devices, (ii) the exploitation of features of multi-processors/multi-core architectures, (iii) the optimization of datatypes and software modules and, finally, (iv) the synchronization of multimodal streams of data having different clocks.

As we will show in details in Section III, EyesWeb XMI can support simultaneously and in a transparent way for the user a wide range of devices (e.g., video cameras, microphones, physiological sensors, shock sensors, accelerometers) using COM, parallel, network, and USB connections. This allows a more integrated development, set up and monitoring of expressive multimodal interactive systems.

As for interoperability, an instance of EyesWeb XMI can easily connect with other instances of EyesWeb XMI running either on the same machine or on other machines. EyesWeb XMI can also connect with other applications either locally or remotely, by means of a set of modules implementing standard network protocols (e.g., TCP/IP, UDP).

Another new feature, which greatly simplifies design and debugging of patches, is the possibility to mark some blocks as design-only. A global run-level can be specified when running a patch; currently supported levels are design and production. Design is used when a patch is still under development and is being debugged, whereas production is used when the patch is completed. If some blocks are marked as design-only, they are not executed in production mode. This minimizes usage of resources, without the need to remove the blocks from the patch. This feature is commonly used for displays, which are often placed in the patch to analyze intermediate results, but which may not represent the actual output of the patch.

Kernel optimization, so that major features of multi processors/multi core computers can be exploited at best, increased the overall performance of the whole platform in handling multiple data streams simultaneously. For instance, in applications containing branches in data flow, analysis and processing of different streams can be performed in parallel on each single branch, thus exploiting the multithreading architecture of the platform. Fig. 1 shows a simple application working on a video stream. When the output of the VideoFileReader block is ready, the video stream forks and the modules in the branches A and B can run as independent threads. This branched and decoupled processing stops when the two streams converge to the same module.

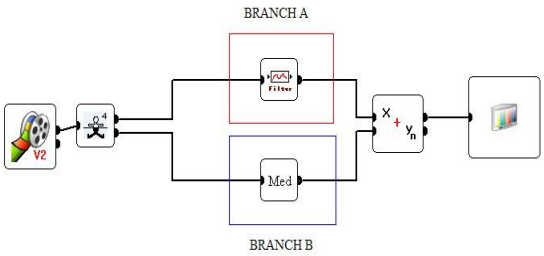

Fig.1.Simple application in which branches A and B can be performed in parallel if more one CPUs is available.

At this point, a resynchronization of the two streams takes place. EyesWeb XMI fully supports the concept of "generic module", which was already partially proposed in the previous version. A generic module is a module able to manage data streams of any implemented EyesWeb datatype. For example, mathematical modules and filter modules can perform common mathematical operations and filtering on datatypes such as scalar values, matrices, pictures and audio buffers. This solution makes the building of multimodal applications simpler for users.

A more complex challenge was to support the synchronization among multimodal data streams coming from real-time and non real-time devices possibly connected to multiple distributed computer systems. In such case, a major problem is to take into account the clock drift among multiple computers and the uneven time

interval between subsequent timestamps, due to the non real-time schedulers on the systems capturing the data.

A new important feature to support synchronization is the availability of the so called *sync-in* and *sync-out* pins. Sync-in pin is an additional input which is available for every block; by means of this pin, the block can be synchronized with any stream of data, not only with the data the block is working on. Another common use of such pin is for synchronizing source blocks with external clocks; many source blocks do not have an inherent clock and might adapt to whichever signal is provided. Blocks reading data from a file, for instance, are not strictly fixed to work with a given clock; if none is provided, it makes sense to use the computer clock as a synchronization signal. However, if an external signal is given, the block adapts to such signal.

The sync-out pin is an additional output, which is available for any block. The sync-out pin provides a way to synchronize different blocks, independently from the data generated by the blocks; in facts, the sync-out pin is signalled whenever a block is activated. Thus, sync-out pins can be used to transform a sink block (which does not generate any output) to a clock source.

Figure 2 shows a *patch* exploiting the sync-in feature. Two WDM Video Input blocks are synchronized with a SerialInput block which acts like an external clock source. The WDM Video Input blocks are enabled when new data are available from SerialInput block. The sync-in pin connection does not cause data transfer, but only information needed for synchronization is spreaded. In the application described in Figure 2, the clock signal for synchronization is extracted from the data stream going out of SerialInput block.

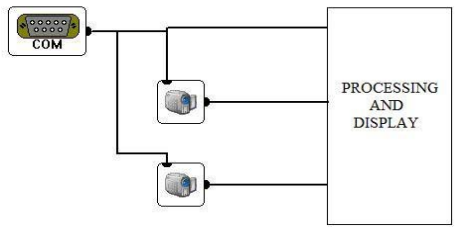

Fig.2. An EyesWeb XMI application using sync-in pin.

Another important feature is a new synchronization feature specifically designed for *super-sampler blocks*. Standard EyesWeb activation modes allowed a block to be synchronized to incoming data (now even to a given clock signal using the sync-in feature) or to use a clock source independent of the patch (e.g., blocks acquiring data from external devices). However, there was a limit of a single output datatype generated per clock cycle; now this limit has been removed and it is straightforward to develop blocks which generate multiple datatype instances per clock cycle.

III. Two Cases Studies: the "Premio Paganini" Experiment and the Orchestra Explorer.

In this Section we present two case studies that show some of the features and potentialities of the EyesWeb XMI.

A. *The "Premio Paganini" Experiment*

One of the test-beds we chose for the testing of the new EyesWeb XMI platform was a concert-experiment we performed in the occasion of the International Violin Competition "Premio Paganini" that took place in Genoa on September 2006. This experiment, carried out in collaboration with Roddy Cowie and Ben Knapp from Queen's University of Belfast and with Carol Krumhansl from Cornell University was performed at the InfoMus Lab site at the Casa Paganini Intl Center of Excellence. The goal was to investigate communication of expressive multimodal content by expressive gestures of violin performers. In this framework, we were interested in considering complex intra personal and inter personal phenomena such as entrainment, empathy and forms of expressing emotions, based on non-verbal communication through music signals, human full-body movement and gesture.

Four selected violin players were involved in the experiment (including violinist from the Premio Paganini intl violin competition) and were asked to play a musical excerpt alone or in duo, and in different environmental and emotional conditions following the instructions provided by psychologists.

In the following we present an overview of the technical aspects of experiment, with particular focus on the management of the different input devices and the synchronization of the different data streams.

1)Technical Set-up

In order to create a reference archive for our research, we recorded a set of multimodal data along several music performances.

Technical set-up complexity resulted from the need to deal with devices having different features and to face the multimodal synchronization problem. We provided a multi-sensor data recording environment employing the following devices:

- four video cameras. We used two ultra-Fast Full-Frame Shutter Digital Cameras (one B&W and one color Silicon Imaging SI-1280F 1280 * 800 pixels, 45 fps) and two Sony video cameras (B&W 800 * 600 pixels, 25fps) to record gestures of the violin players. We employed two video cameras for each player. Two video cameras hung from a steel cable 5-meters high and were placed above the position of the two players; the other two were placed in front of players. This set-up allowed us to record performers' gestures from different viewpoints (e.g., full-body gestures and arms/hand/fingers gestures).
- four microphones (two Neumann KM 184 cardioid microphones and two radio microphones AKG C444) at 48 kHz and 16 bits per channel. These microphones were used to record environmental sound and violin sound respectively;
- BioMuse sensors. BioMuse (www.biocontrol.com/biomuse.html) is a 8 channels biosignal processing device for collecting and processing human bioelectric signals acquired using standard non-invasive transdermal electrodes. Communication between computer and Biomuse takes place

through a standard serial interface. In order to allow the device to communicate with the EyesWeb XMI platform, SARC (Queen's University) developed an application for receiving in EyesWeb BioMuse data from the COM port and for decoding the communication protocol of BioMuse. In our experiment we monitored and recorded the EKG signal and the EMG signal of left forearm muscle of the players during the whole music performance.

Data were recorded on the disks of four computers in an uncompressed custom file format. Each computer ran an EyesWeb XMI instance to manage the different devices and to record and synchronize the data streams.

2)Synchronizing Multimodal Data Streams

To face the problem of data streams synchronization, we implemented a two-steps solution.

Four computers (one master and three slaves) were used and an external sync signal, generated by the master, was sent to them to have a single world clock. Fig. 3 shows a sketch of this set up. Generally, in set ups needing synchronization of multiple signals from video cameras and microphone channels, clap sounds and strobe flashes are used to limit drift among different clocks. Several tools and techniques can be found in the literature to help post-recording synchronization of multimodal data on non-real time Operating Systems and using commodity hardware. Among these, a very interesting approach based on audiovisual synchronization pulses, statistical procedures and software tools is described in [7].

In our case study, however, clap sound and strobe flash was not enough to provide all capture systems with a common reference, because we dealt with physiological signals by BioMuse too. Our solution therefore exploited the synchronization mechanisms provided by EyesWeb XMI and in particular the sync in pin which allows to use external sync signals for controlling the activation of EyesWeb blocks.

Fig. 4 shows the EyesWeb application (patch) we used. The MultimediaFileReader module of the master enables the reading of the other MultimediaFileReader modules sending them the external sync signal previously recorded. This is done forcing the clock source of these modules using the sync in pin; in this way these modules are activated according to the speed of the first block. In addition, a TimeExtraction module extracts from the external sync signal the MediaTime information. *MediaTime* is used to set the file position pointer for data streams of the slave blocks during the synchronized reading.

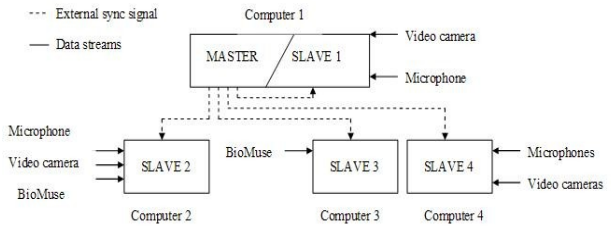

Fig.3. Sketch of computers connections and technical set up, including the capture devices connected to each computer.

B. *The Orchestra Explorer*

The second case study we present here, The Orchestra Explorer [5], was proposed at the science exhibition "Cimenti di invenzione e armonia" in the framework of "Festival della Scienza 2006", at Casa Paganini, Genova.

The Orchestra Explorer implements a novel paradigm for active listening, enabling users to physically navigate inside a virtual orchestra, to actively explore the music piece the orchestra is playing, to modify and mould in real-time the music performance through expressive full-body movement and gesture. Concretely, the virtual orchestra is spread over a physical surface. By walking and moving on the surface, the user discovers each single instrument and can operate through her expressive gestures on the music piece the instrument is playing.

1)Technical Set-up and Synchronization

The installation for "Cimenti di Invenzione e Armonia" covered a surface of about 9 m 3.5 m (the stage of the Auditorium at Casa Paganini). A single video camera observed the whole surface from the top, about 7 m high, and at a distance of about 10 m from the stage. Four loudspeakers were placed at the four corners of the stage for audio output. A white screen covered the back of the stage for the whole 9 m width. A video projector projected on such screen the video feedback. Lights were set in order to enhance the feeling of immersion for the users and to have a homogenous lighting of the stage.

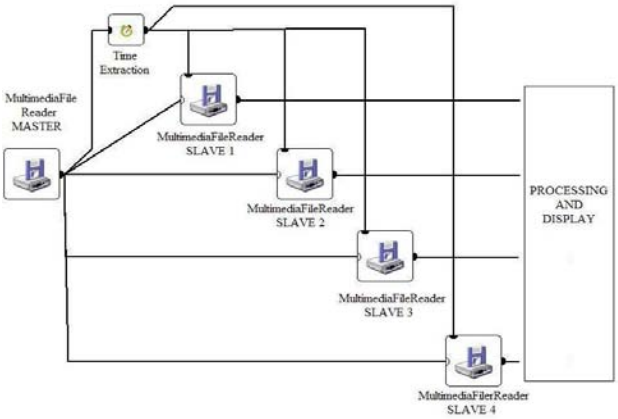

Fig.4. The EyesWeb XMI application we implemented for synchronized recording and reproduction of multimodal data streams.

The music piece "Borderline", by M. Canepa, L. Cresta, and A. Sacco, was selected for the installation, including 13 stereo audio tracks (26 mono channels) that need to be kept synchronized. The video stream from the video camera also needs synchronized processing for extraction of expressive gesture features. Synchronized video output is needed too.

In the Orchestra Explorer synchronization was obtained using the master/slave feature of the EyesWeb XMI input modules. The first audio file reader is set as master and controls the scheduling of the other readers. The audio section of the EyesWeb patch is shown in Fig. 5.

This mechanism allowed us to play and process synchronously in real-time the 13 audio files on the same machine (Dell Precision 380, equipped with two CPUs Pentium 4 3.20 GHz, 1 GB RAM, Windows XP Professional), while at the same time controlling the audio processing by means of expressive gesture parameters and generating visual feedback in real-time.

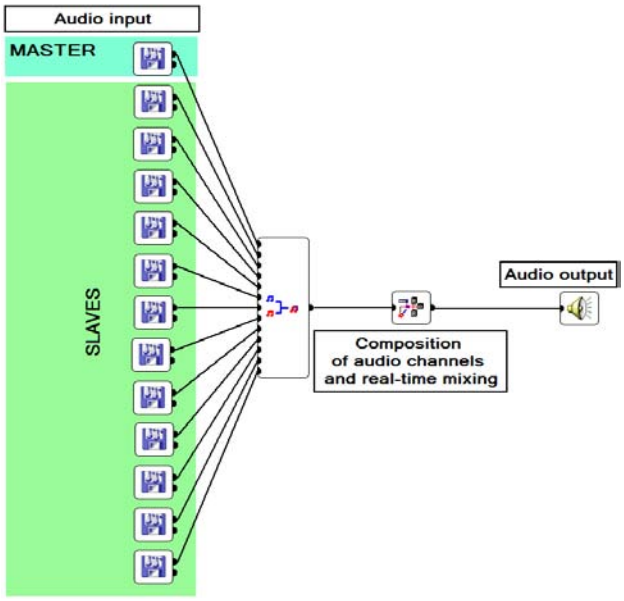

Fig.5. The audio part of the EyesWeb XMI patch we developed for synchronization of 13 stereo audio channels in the Orchestra Explorer installation.

IV. Conclusions

In this paper we presented the new EyesWeb XMI software platform for gesture analysis, synthesis and processing. We presented some main improvements of the EyesWeb XMI platform with respect to its predecessors and two case studies in which the new features were heavily used.

Acknowledgment

The "Premio Paganini" Experiment has been performed with the support of EU-IST Network of Excellence HUMAINE (Human-Machine Interaction Network on Emotion) in the 6th Framework Program (2004-2007).

The Tangible Acoustic Interfaces techniques and software library for EyesWeb XMI has been partially supported by the TAI-CHI EU IST Project (2004-2007).

We thank Benjamin Knapp (SARC, Queen's University, Belfast) and Carol Krumhansl (Cornell Psychology Department, Cornell University, Ithaca) and the violin player Diana Jipa (semifinalist at Premio Paganini) for their contribution to the experiment.

We thank also our collegues C. Canepa, G. Castellano, D.Glowinski, B Mazzarino, for their important contributes to this project.

References

[1] A. Camurri, B. Mazzarino, M. Ricchetti, R. Timmers and G. Volpe, "Multimodal analysis of expressive gesture in music and dance performances," In: Camurri, A., Volpe, G. (eds): *Gestures-based Communication in Human-Computer Interaction. Lectures Notes in Artificial Intelligence,* vol 2915. Springer-Verlag Berlin Heidelberg New York (2004) 20-39.

[2] A. Camurri, P. Coletta, A. Massari, B. Mazzarino, M. Peri, M. Ricchetti, A. Ricci and G. Volpe, "Toward real-time multimodal processing: EyesWeb 4.0," *In Proc. AISB 2004 Convention: Motion, Emotion and Cognition,* Leeds, UK (2004).

[3] A. Camurri, G. De Poli, M. Leman, and G. Volpe, "Communicating expressiveness and affect in multimodal interactive systems," *IEEE Multimedia* (2005, January-March) 12, 43-53.

[4] A. Camurri and G. Volpe, "A multimodal and crossmodal processing in interactive systems based on tangible acoustic interfaces," *Proc. of the 15th IEEE International Symposium on Robot and Human Interactive Communication (ROMAN2006),* University of Hertfordshire, Hatfield, United Kingdom, September 2006.

[5] A. Camurri, C. Canepa, and G. Volpe, "Active listening to a virtual orchestra through an expressive gestural interface: the orchestra explorer," In Preparation.

[6] X. Kurtenbach, X. Hulteen, "Gesture in human-computer interaction," In Laurel,. B (ed) *The Art of Human-Computer Interface Design.* Addison-Wesley Reading MA (1990).

[7] M. Martial, V. Stanford, "Synchronizing multimodal data streams acquired using commodity hardware," *In Proc. 4th ACM Int'l Whorkshop on Video Surveillance and Sensor Network. Santa Barbara, California, USA, 2006* .