



DISSEMINATION LEVEL: PUBLIC

Social Interaction and Entrainment using Music PeRformanceE

SIEMPRE

D5.7 SIEMPRE Database and Repository

<i>Version</i>	<i>Edited by</i>	<i>Changes</i>
1	UNIGE (A.Camurri & D.Glowinski)	
2	UPF (E.Maestre & O. Mayor)	
3	IIT (L.Badino & A. d'Ausilio)	
4	UNIGE (A.Camurri & D.Glowinski)	



October, 31th 2012



Table of Contents

1. INTRODUCTION	3
2. Full Multimodal Recordings Repository.....	3
2.1 Experiments.....	4
2.2 Raw Data	5
2.3 Algorithms.....	6
2.4 EyesWeb Interactive Applications	7
3. Repovizz: Online Multimodal Recordings Repository and Visualizer	8
3.1 Introduction.....	8
3.2 Overview.....	8
3.3 Structured Data Formatting	10
3.3.1 The Datapack Designer.....	10
3.3.2 Data Formatting.....	12
3.3.3 The Datapack	13
3.4 Back-end server.....	14
3.4.1 Database structure	14
3.4.2 Back-end data processing.....	15
3.5 Web-based front-end interface	15
3.5.1 The Repository Navigator	16
3.5.2 The Datapack Visualizer.....	16
3.5.3 Usage of the Visualizer.....	18



1. INTRODUCTION

This deliverable describes the SIEMPRE Database and Repository. It consists of a publicly available annotated database of multimodal behavioural data related to creative social performances available to the research community. This database includes a selection of annotated data (e.g., motion, audio, and biometric features) obtained from the two series of SIEMPRE experiments on string quartet, orchestra and audience. Data are compliant with the formats adopted and agreed in the project (see Task 3.2). A repository of selected algorithms and techniques developed in the project, implemented e.g., as C++ libraries, EyesWeb XMI modules, Matlab toolboxes, is also available for feature extraction, data analysis and visualization.

One of the main objectives and challenges of the SIEMPRE project is the recording, synchronization and simultaneous visualization of multimodal data in each of the three scenarios. The novelty of the experimental design and setup of the multimodal datasets is also a strong component of the SIEMPRE project. The scientific questions we focused on, as well as the far from trivial technological challenges they require to be met, make the datasets themselves and the software tools developed for preprocessing, synchronizing and visualization of the data a valuable resource for a diverse number of research areas ranging from computer music to neuroscience. For those reasons the SIEMPRE consortium has created a repository that is freely available to any scientist interested in it.

The SIEMPRE Repository is structured in two components:

- (1) Full Multimodal Recordings Repository: raw repository of multimodal data and eyesweb patches and matlab scripts for visualization, processing and analysis.
- (2) RepoVizz: an online multimodal recordings repository and visualizer. RepoVizz is a web application for immediate access to the available data

2. FULL MULTIMODAL RECORDINGS REPOSITORY

From the Repository home page (<http://siempre.infomus.org/repository/>) it is possible to access and download datasets (available from the Experiments page: <http://siempre.infomus.org/repository/>) and algorithms (available at <http://siempre.infomus.org/repository/algorithms.php>) created and developed within the SIEMPRE projects by the SIEMPRE partners (see Figure 1).



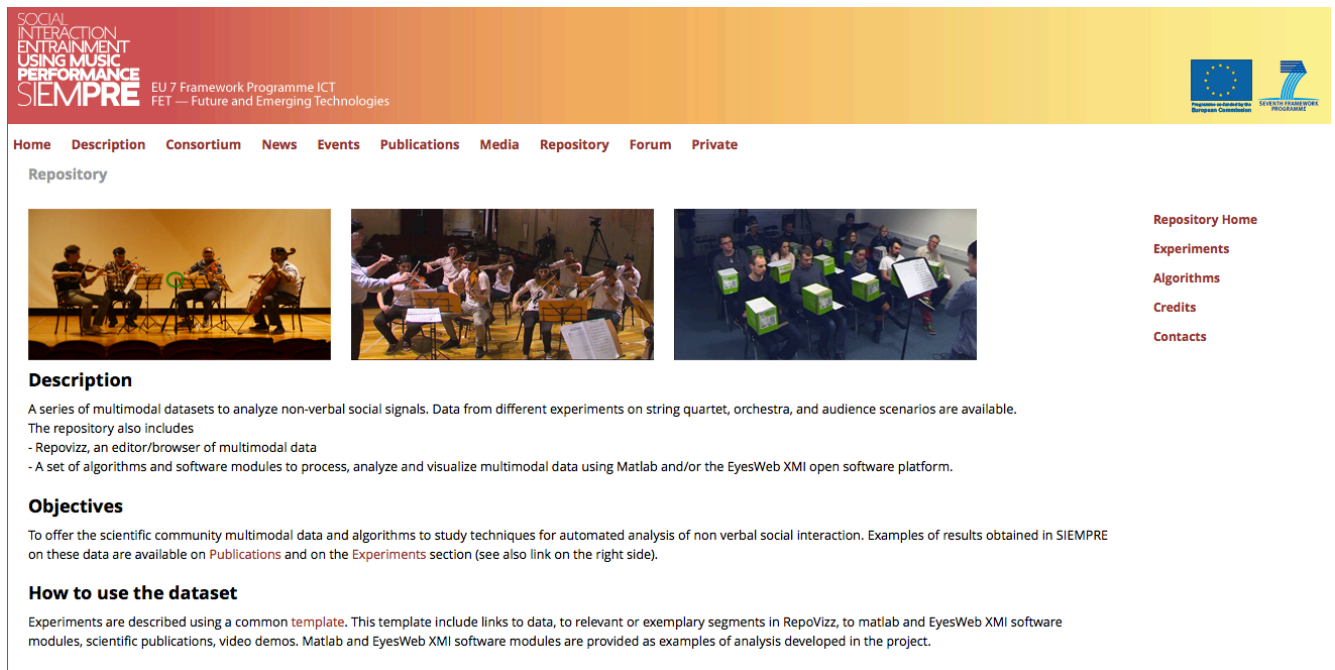
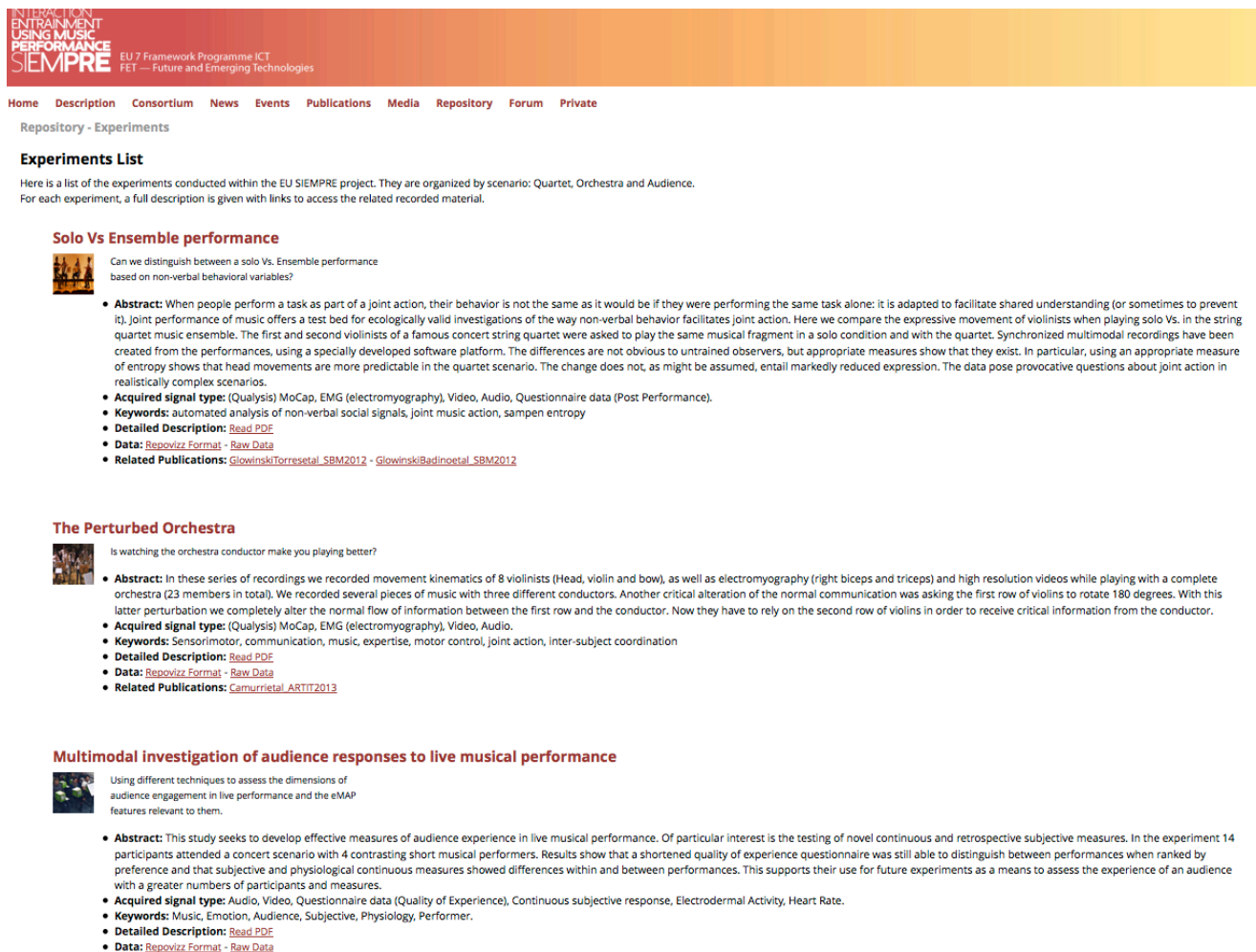


Figure 1. Screen-shoot of the SIEMPRE repository landing web-page

2.1 Experiments

The **Experiments** page lists all the datasets. Each dataset has a) a short description (i.e., an abstract); b) a list of keywords; c) a list of the types of acquired signal; d) a link to a detailed description of the dataset (not only including information about the experimental setup but also, e.g., about the scientific motivations behind the creation of the dataset); e) links to the dataset itself.



Solo Vs Ensemble performance

Can we distinguish between a solo Vs. Ensemble performance based on non-verbal behavioral variables?

- Abstract:** When people perform a task as part of a joint action, their behavior is not the same as it would be if they were performing the same task alone: it is adapted to facilitate shared understanding (or sometimes to prevent it). Joint performance of music offers a test bed for ecologically valid investigations of the way non-verbal behavior facilitates joint action. Here we compare the expressive movement of violinists when playing solo Vs. in the string quartet music ensemble. The first and second violinists of a famous concert string quartet were asked to play the same musical fragment in a solo condition and with the quartet. Synchronized multimodal recordings have been created from the performances, using a specially developed software platform. The differences are not obvious to untrained observers, but appropriate measures show that they exist. In particular, using an appropriate measure of entropy shows that head movements are more predictable in the quartet scenario. The change does not, as might be assumed, entail markedly reduced expression. The data pose provocative questions about joint action in realistically complex scenarios.
- Acquired signal type:** (Qualisys) MoCap, EMG (electromyography), Video, Audio, Questionnaire data (Post Performance).
- Keywords:** automated analysis of non-verbal social signals, joint music action, sample entropy
- Detailed Description:** [Read PDF](#)
- Data:** [RepoVizz Format](#) - [Raw Data](#)
- Related Publications:** [GlowinskiTorresetal_SBM2012](#) - [GlowinskiBadinoetal_SBM2012](#)

The Perturbed Orchestra

Is watching the orchestra conductor make you playing better?

- Abstract:** In these series of recordings we recorded movement kinematics of 8 violinists (Head, violin and bow), as well as electromyography (right biceps and triceps) and high resolution videos while playing with a complete orchestra (23 members in total). We recorded several pieces of music with three different conductors. Another critical alteration of the normal communication was asking the first row of violins to rotate 180 degrees. With this latter perturbation we completely alter the normal flow of information between the first row and the conductor. Now they have to rely on the second row of violins in order to receive critical information from the conductor.
- Acquired signal type:** (Qualisys) MoCap, EMG (electromyography), Video, Audio.
- Keywords:** Sensorimotor, communication, music, expertise, motor control, joint action, inter-subject coordination
- Detailed Description:** [Read PDF](#)
- Data:** [RepoVizz Format](#) - [Raw Data](#)
- Related Publications:** [Camurrietetal_ARTIT2013](#)

Multimodal investigation of audience responses to live musical performance

Using different techniques to assess the dimensions of audience engagement in live performance and the eMAP features relevant to them.

- Abstract:** This study seeks to develop effective measures of audience experience in live musical performance. Of particular interest is the testing of novel continuous and retrospective subjective measures. In the experiment 14 participants attended a concert scenario with 4 contrasting short musical performers. Results show that a shortened quality of experience questionnaire was still able to distinguish between performances when ranked by preference and that subjective and physiological continuous measures showed differences within and between performances. This supports their use for future experiments as a means to assess the experience of an audience with a greater numbers of participants and measures.
- Acquired signal type:** Audio, Video, Questionnaire data (Quality of Experience), Continuous subjective response, Electrodermal Activity, Heart Rate.
- Keywords:** Music, Emotion, Audience, Subjective, Physiology, Performer.
- Detailed Description:** [Read PDF](#)
- Data:** [RepoVizz Format](#) - [Raw Data](#)

Figure 2. Screen-shoot of the SIEMPRE repository experiments web-page

A dataset can be accessed with different modalities and tools: 1) raw data 2) EyesWeb interactive applications 3) RepoVizz. Raw data enable researchers to use whatever techniques and tools to analyze the data. EyesWeb is an open software platform freely available at http://www.infomus.org/eyesweb_eng.php enabling real-time multimodal processing. RepoVizz is an online

2.2 Raw Data

A dataset in the “Raw Data” format is a set of folders containing the multimodal data of the recordings (see Figure 2). Multimodal data include:

- Audio: from ambient stereo and piezo-electric microphones attached to single instrument (.wav)
- metadata: all the documents that give information about the setup, the onset/offset of each take referring to the smpte timecode, questionnaires submitted to performers or audience, etc. (.pdf, .doc, .xls)
- mocap: Motion Capture data recorded from the Qualisys Motion Tracking system (.c3d)
- sync: smpte time code used to synchronize the multimodal data (.wav)
- video: video recording from two HD cameras

The multimodal data, once downloaded, can be visualized or analyzed through a set of Matlab scripts or EyesWeb interactive applications available on the **Algorithm** section of the repository.

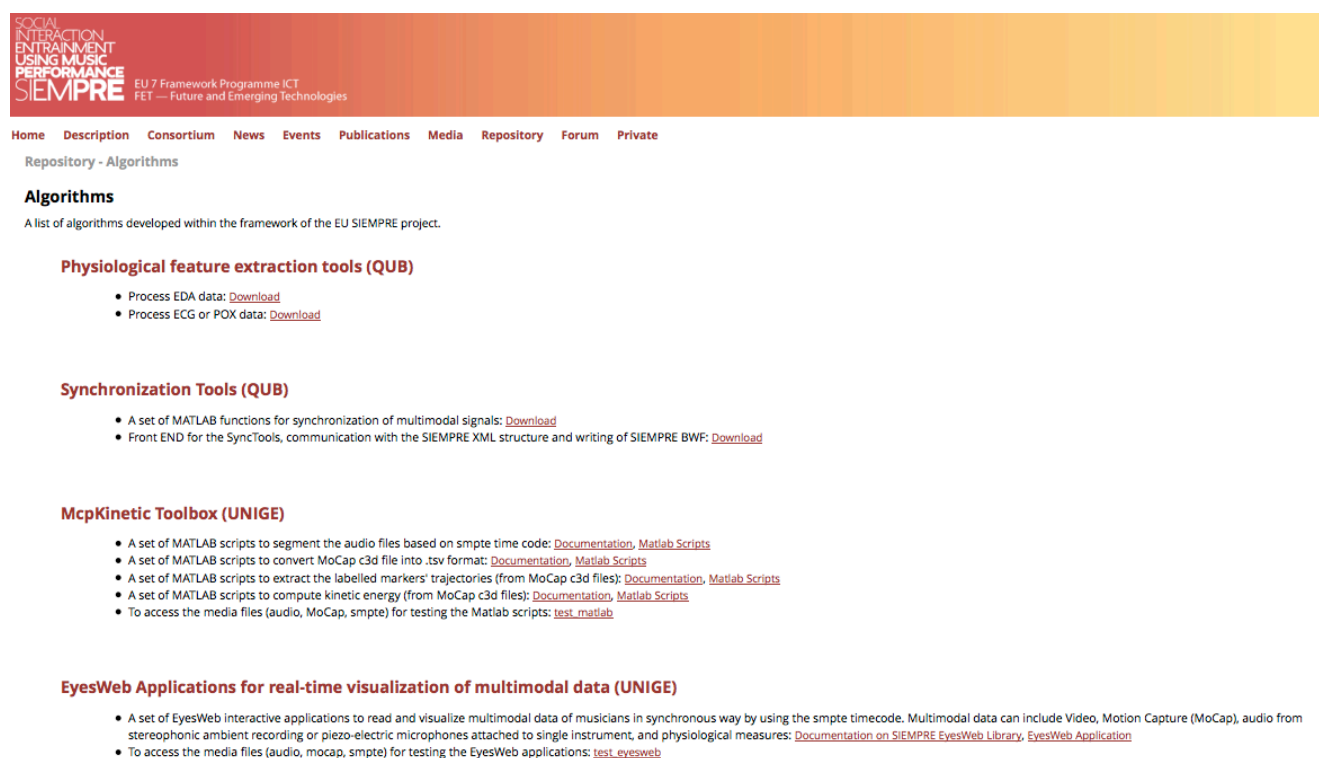
Index of /repository/data/quartet/SoloVsEnsemble

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 audio/	06-Dec-2012 12:20	-	
 metadata/	19-Dec-2012 00:12	-	
 mocap/	06-Dec-2012 12:20	-	
 sync/	06-Dec-2012 12:24	-	
 video/	19-Dec-2012 00:34	-	

Figure 2. Snapshot of what appears to the user when clicking the RawData link in each Experiment Description (the SoloVsEnsemble experiment in this figure). Multimodal data of the recordings are available in their original format and organized by modality (e.g., audio, video, etc.).

2.3 Algorithms

The **Algorithm** page includes a short description on the advanced methods of analyses employed in the SIEMPRE consortium by different partners, a tutorial and multimodal samples for running test (see Figure 3). A complete description of the algorithm can be found in the relative deliverable produced within the project and freely available in the website.



Algorithms

A list of algorithms developed within the framework of the EU SIEMPRE project.

Physiological feature extraction tools (QUB)

- Process EDA data: [Download](#)
- Process ECG or POX data: [Download](#)

Synchronization Tools (QUB)

- A set of MATLAB functions for synchronization of multimodal signals: [Download](#)
- Front END for the SyncTools, communication with the SIEMPRE XML structure and writing of SIEMPRE BWF: [Download](#)

McpKinetic Toolbox (UNIGE)

- A set of MATLAB scripts to segment the audio files based on smpte time code: [Documentation](#), [Matlab Scripts](#)
- A set of MATLAB scripts to convert MoCap c3d file into .tsv format: [Documentation](#), [Matlab Scripts](#)
- A set of MATLAB scripts to extract the labelled markers' trajectories (from MoCap c3d files): [Documentation](#), [Matlab Scripts](#)
- A set of MATLAB scripts to compute kinetic energy (from MoCap c3d files): [Documentation](#), [Matlab Scripts](#)
- To access the media files (audio, MoCap, smpte) for testing the Matlab scripts: [test_matlab](#)

EyesWeb Applications for real-time visualization of multimodal data (UNIGE)

- A set of EyesWeb interactive applications to read and visualize multimodal data of musicians in synchronous way by using the smpte timecode. Multimodal data can include Video, Motion Capture (MoCap), audio from stereophonic ambient recording or piezo-electric microphones attached to single instrument, and physiological measures: [Documentation on SIEMPRE EyesWeb Library](#), [EyesWeb Application](#)
- To access the media files (audio, mocap, smpte) for testing the EyesWeb applications: [test_eyesweb](#)

Figure 3: Screen-shot of the SIEMPRE repository algorithms web-page

2.4 EyesWeb Interactive Applications

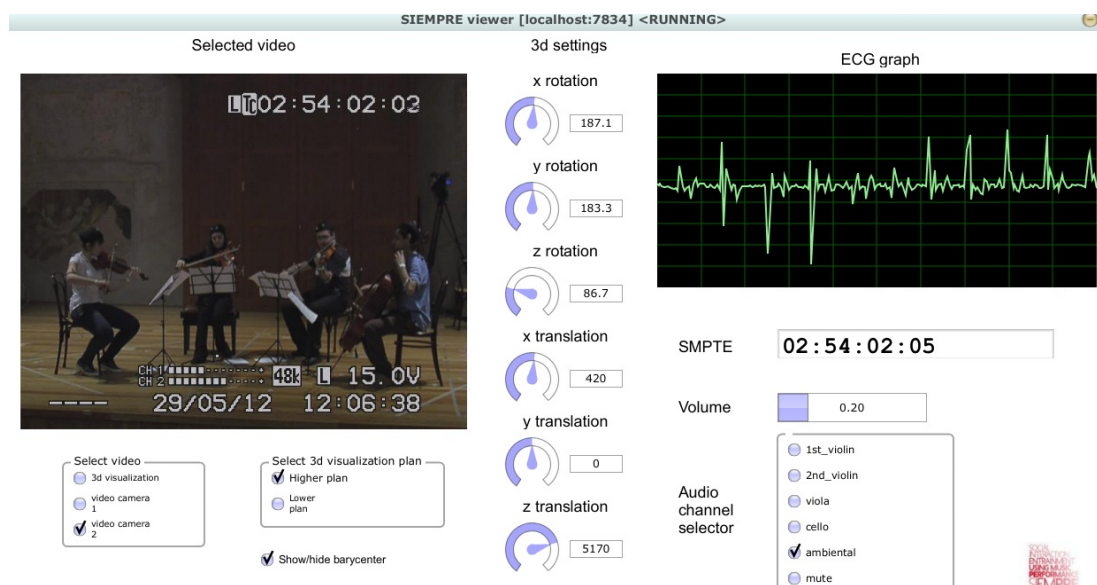


Figure 4. A snapshot of the EyesWeb application for visualizing in real-time the multimodal data related to the String Quartet ensemble performances.

A set of interactive applications based on the SIEMPRE- EyesWeb open software platform (see Deliverable 3.1) are available to read, visualize and analyze the multimodal data of the recordings. The user can download the data of interest from the repository and have an immediate rendering of the music ensemble performance thanks to this set of applications.

The EyesWeb application allows the user to explore in real-time different views (frontal, vertical, etc.) different modalities (video, mocap) altogether, and access ambient as well as single microphone recordings in real-time (see Figure 4). Other proposed applications allow the user to visualize individual and group behavioral features such as the polygon linking the musicians' head, the distance with respect to the center of the string quartet, etc (See Figure 5).

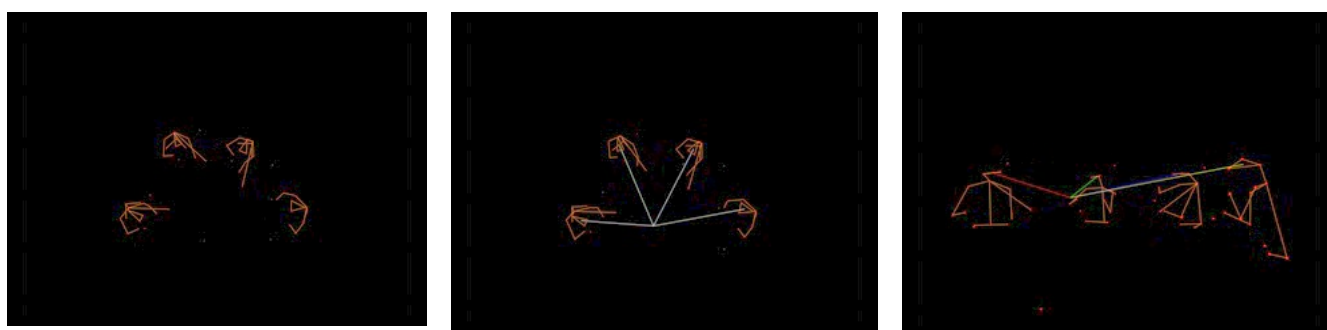


Figure 5. Snapshots of three possible analyses of the String Quartet Musicians using EyesWeb interactive applications. From left to right: (a) vertical view, (b) vertical view with visualization of the distance between musician's head and subjective center of the quartet (ear), (c) frontal view with visualization of musician's head and ear distance.

3. REPOVIZZ: ONLINE MULTIMODAL RECORDINGS REPOSITORY AND VISUALIZER

3.1 Introduction

Multi-modal data stream acquisition and analysis constitute a major pursuit in much of today's data-driven scientific research. The lack of common formats and structured storage tools for heterogeneous data forces researchers to use ad-hoc formatting schemes and software tools developed "in house". Despite the obvious potential offered by collaborative, exchange-driven research, difficulties often found for sharing or browsing data sets (sometimes of large size) are, as of today, impeding research collaboration and cross-fertilization.

repoVizz has been developed by UPF in the context of a number of research projects (including SIEMPRE) that involve storage and manipulation of large-size multi-modal datasets. repoVizz is an integrated solution for structured formatting and storage, browsing, sharing, annotation, and visualization of synchronous multi-modal time-aligned data. repoVizz offers means for organizing heterogeneous data (different modalities, sample rates, etc.) through a totally customizable tree structure that holds identifiers, descriptions, and pointers to data files. Most of operations on datasets are carried out on-line through a powerful HTML5 visual interface that allows working on a browser. This feature, which can be considered a key aspect of repoVizz, represents a step forward because data can be explored, annotated, or downloaded with from any location and with no software requirements other than a browser.

As a response to the needs imposed by the challenges of SIEMPRE and its public data repository, UPF has adapted several of the specifications of the repoVizz platform. The browsing and download of SIEMPRE multimodal recordings through the repoVizz framework is open to all the scientific community via a web portal (<http://repovizz.upf.edu>). Specific links to ensemble performance datasets and a short how-to guiding the access to data are available. Although still under discussion among the project partners, recording data are going to be available through a Creative Commons Non-Commercial license.

repoVizz offers, as a whole, (i) a methodology to define the data structure of multi-modal recordings of music performances, (ii) an on-line repository to store/share all the data collected by the different partners of the project, and (iii) a tool to visualize acquired data and make annotations in a user-friendly, flexible manner. At its current state, RepoVizz is designed to hold synchronized streams of heterogeneous data (audio, video, motion capture, physiological measures, extracted descriptors, etc.), annotations, and musical scores. Non-video data streams are stored as single-channel PCM files. The motion-sequence format MJPEG is chosen for video. Annotations are stored as text files, and MusicXML is used for musical scores. Data is structured by a customizable XML skeleton which is created prior to data acquisition, and provides means to (re-)organize gathered data in any desired hierarchical structure. Once a multi-modal dataset is created and uploaded to the server, XML skeleton files are used both to structure data and to define a repository view. Multi-track multi-modal data visualization is carried out remotely via a HTML5-based front-end that enables web-driven interactive visualization and editing and downloading of datasets.

The complete documentation of repoVizz can be found in: <http://repovizz.upf.edu/docs/>

3.2 Overview

The framework consists of a set of tools for structuring, formatting, storing, browsing, editing, and exchanging multimodal recordings. Two main types of users (not mutually exclusive) have been considered when devising repoVizz. A first group includes users that acquire multimodal data sets (in



our current application context these would be the researchers who carry out experiments on ensemble music performance) and contribute data to the repository, so that it can be shared with other users. A second group encompasses users that access the repository in order to explore, annotate, retrieve (download), and/or process existing data. This group also would include those users who, once a dataset or (a portion of a dataset) has been downloaded, apply some analyses and upload their results as additional streams or annotations. In our context, the second group of users may be musicians, researchers or musicologists who annotate or comment on existing multimodal recordings.

In Figure 6 we depict the process of creating, storing, visualizing and sharing a multimodal recording following all steps from the preparation of the recording to the sharing of it. First, (1) a customized tool is used to define the recording setup of the multimodal recording and write it to a skeleton XML file, then (2) during the recording session, following the setup defined in the previous step, all data streams are acquired. All these streams of data have to be synchronized and stored (3) in a proper format before being uploaded to the server so they are compatible with repoVizz. For the synchronization and preparation of the data, if signals are recorded using different devices, SMPTE-based synchronization and score alignment tools have been developed by the different research groups in SIEMPRE and are being used for that purpose in the SIEMPRE application context.

All these synchronized and processed data files are gathered together and assigned to the previous XML skeleton file to build a package called *datapack* (4), which consists of a compressed file that contains all streams and other associated content. The *datapack* is then uploaded to a folder in the server (5) using the web interface. The server stores all *datapack*'s files in the repository and feature extraction is performed to extract some additional descriptors from the data streams (6). Once the multimodal recording is uploaded to the server and processed, users can navigate through the existing *datapacks* (7a), visualize them (7b) and edit/add annotations (7c) using the powerful & fully configurable web interface based in state-of-the-art HTML5. Moreover, users may select a segment and download a *datapack* containing all synchronized data streams contained in the segment (8). Once the user has downloaded the *datapack* they can perform further analysis on the downloaded multimodal data using standard research tools.



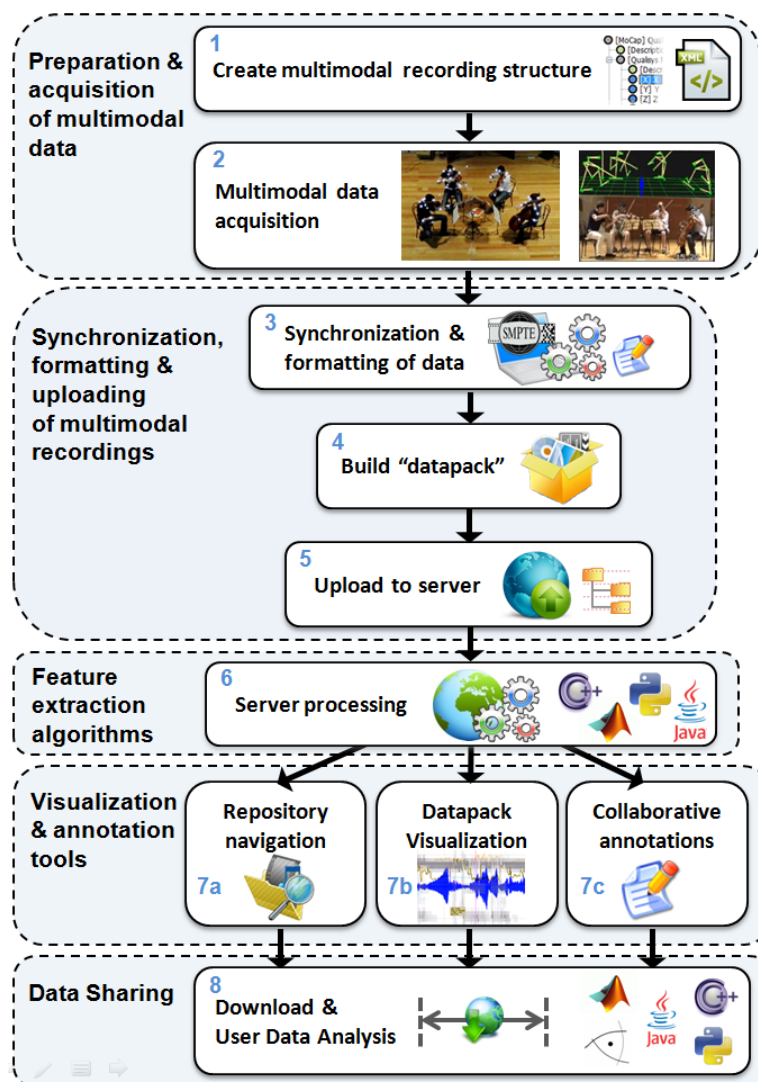


Figure 6. Process of creating, storing, visualizing and sharing a multimodal recording within the repoVizz framework

3.3 Structured Data Formatting

3.3.1 The Datapack Designer

The repository is intended for storing multimodal data from a multimodal recording. The first step before doing a recording is to create a skeleton file that will define what is going to be recorded. For this purpose, the Datapack Designer has been developed. With this tool, users can easily build a tree structure representing all devices and associated data streams that will be acquired from a particular multimodal recording.

The tree structure that defines the skeleton file is organized into nodes and sub-nodes. There are 5 different types of node, each one with a different meaning and represented with a different color (see Figure 7):

- Generic (white) nodes: are used for structuring and grouping data. They operate as folders for defining general categories and sub-categories (audio, video, acquisition device, position markers, physiological sensors, etc). These nodes can contain as a sub-node any of the other node types or even another generic node.

- Signal (blue) nodes: contain the links to the continuous data streams in BWF standard format that will be obtained from the recorded multimodal data. Some examples of these continuous data streams are the sensor position marker data acquired using a motion capture (moCap) device, physiological sensor data or any time series signal.
- Audio (green) nodes: contain audio files in WAV PCM format.
- Video (yellow) nodes: contain video files in MJPEG format.
- File (orange) nodes: are used for storing data streams which cannot be stored in BWF format like for instance music scores, PDFs, images, etc.
- Annotation (pink) nodes: contain xml files with event or segment annotations made by users.
- Description (gray) nodes: contain a descriptive text. These nodes are used for associating a description to any other node.

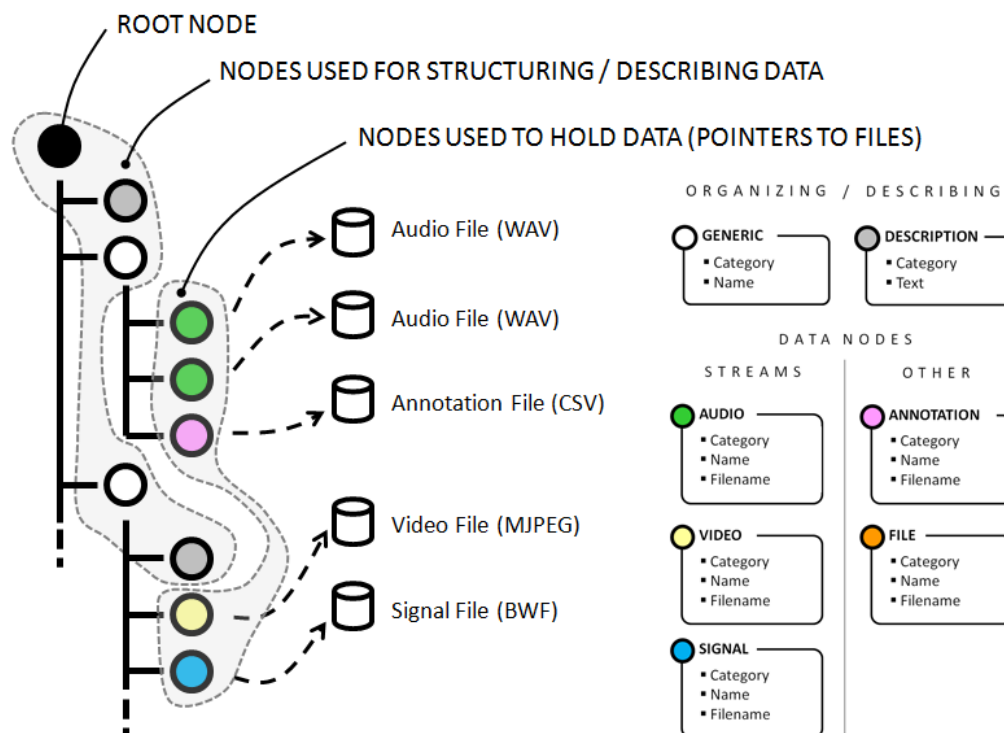


Figure 7. Node types for the structured tree that contains the datapack skeleton.

With the Datapack Designer Tool, using the graphical user interface, any inexperienced user can easily build the XML skeleton file. All the above node types can be inserted in the tree structure that defines the skeleton, assigning names and description for each node and providing the links to the data stream files. It's also possible to save templates containing a set of nodes and sub-nodes for reusing them when a particular structure is repeated along the skeleton file tree (see Figure 8).

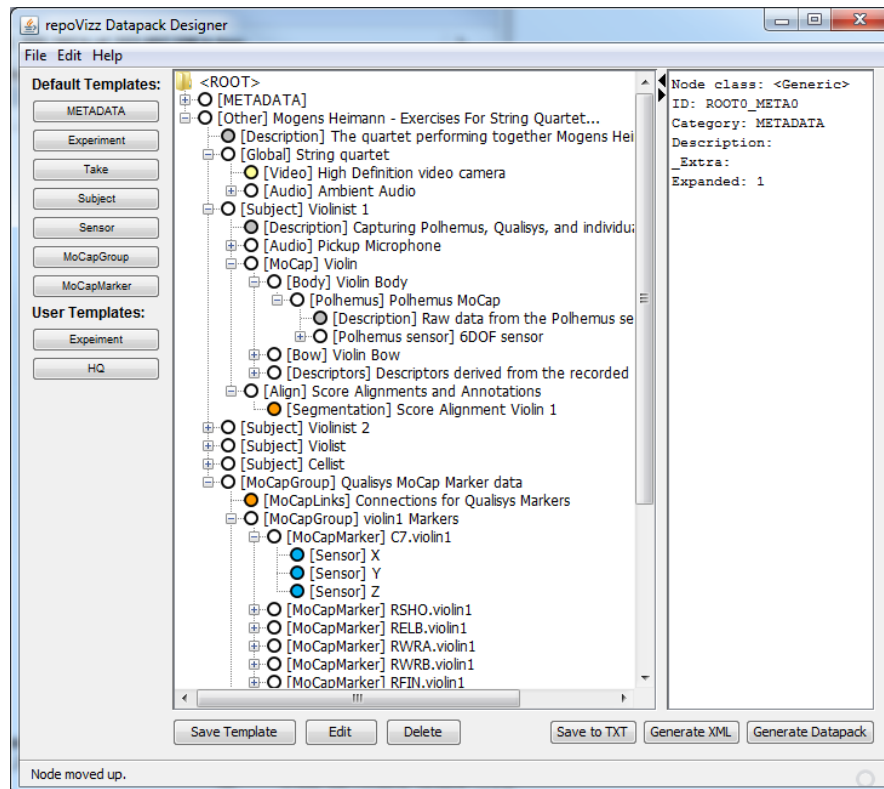


Figure 8. Datapack Designer Tool.

3.3.2 Data Formatting

After acquiring data streams during a multimodal recording, multiple streams come up from different devices. Each device has an independent clock, different sampling rates and for storing the acquired data stream uses its own format. This is a problem when a researcher wants to perform multimodal data analysis because each signal has to be parsed in a different way and all streams need to be synchronized to the same world clock and also adjust the possible time offsets between different signals (all recording devices may not start recording at the same time). Before uploading the multimodal data to the repoVizz repository, it's a requirement to convert all the data to a standardized format and before converting to this format the synchronization between the streams needs to be performed.

For the synchronization, SMPTE streams are used when the recording device accepts it, otherwise pulse streams are used to implement a method for synchronization avoiding drifts between the signals due to inaccuracy in the device clocks. Sampling rates for each stream are stored as metadata, also accepting non integer values. A set of Matlab synchronization scripts have been developed in the context of the SIEMPRE project and are available for download from the project website [6]. These scripts are able to synchronize multiple files acquired using different devices by using the sampling rate of each signal and a SMPTE or pulse synchronization signal. The synchronization is done by applying interpolation and re-sampling to the data as needed in order to correct drifts in the data. Once all data streams are synchronized, they are stored in a common format before being uploaded to the server.

For all continuous one-dimensional data streams, as a consensus between all the SIEMPRE project partners, a binary format called BWF (Broadcast Wave Format) has been adopted. BWF is basically an extension of the popular Microsoft WAVE audio format with the addition of an XML header chunk. The XML chunk is used for storing metadata associated with each file like: the recording device that

was used, its sampling rate or the range of values corresponding with the maximum and minimum values of each particular device. The advantage of using a binary format like BWF is that it allows to be opened by many audio editors (although only for visualization because data streams with low sampling rates won't be playable) and also by numerical computing softwares like PyLab, Octave or Matlab. Some examples of one-dimensional data streams are: audio streams, physiological sensor data, MoCap marker x, y & z positions, synchronization SMPTE streams, etc.

Video files are stored using the Motion JPEG format that uses intraframe-only compression. This method produces larger files than other video formats like H.264 or MPEG-4 but allows cutting video files at any starting and ending frame avoiding the process of decode and encode again losing quality, as it happens with other interframe compression formats. This is crucial because at a later stage (when sharing the data in the server repository) we will allow the user to cut and download a specific segment from the server and we want to preserve the original quality.

Apart from the above examples, some other data associated with a multimodal recording may not allow to be stored in any of the previous formats because the nature of the data. For these particular cases other data formats are used.

The repoVizz framework is not limited to the above formats. The framework is open and extensible enough to allow any other standard data format. The use of the above has been adopted for convenience for the deployment of the repoVizz as a supporting tool for the SIEMPRE repository, but for any other uses a different standardized data format may be used.

As an example of other formats for storing data, in the context of the SIEMPRE project, we are dealing with musical scores. For this particular case, the MusicXML standard is used, providing an open XML-based music notation file format. Other data like the user segmentations and annotations, and MoCap translation matrices and models are stored using customized XML files.

3.3.3 The Datapack

A *datapack* is a compressed file that contains the XML skeleton file and all data streams and extra information that has been linked to the file using the Datapack Designer Tool.

The process of creating a *datapack* is as follows: (1) the skeleton file is created using the Datapack designer Tool, (2) then the filenames of the data files contained in the XML skeleton are defined manually or automatically depending on the user preferences, (3) from the multimodal recording, all data files are processed and stored in the XML skeleton folder, and (4) using the Designer Tool the user generates the *datapack*, just pressing a button.

One of the most tedious tasks when creating a *datapack* is the linking between the recorded data files and the XML skeleton data nodes. Doing it manually is an enormous amount of work, especially for *datapacks* with hundreds of data nodes. For each node, the user has to manually assign a file to it, browsing into the local user's hard disk. To avoid this, an automatic file naming method has been implemented. The filename for each data node is automatically generated from a unique ID that identifies each of the nodes in the skeleton XML file. The same unique IDs (extracted from the XML skeleton file) are then used by the synchronization and storing tools explained in the previous section to generate the appropriate filenames for data files.

As said before, the *datapack* will contain all necessary data and information from a multimodal recording to be uploaded to the repoVizz repository for sharing it with the research community. The upload to the server is restricted to some research groups that have adopted the repoVizz protocol for defining the recording setup and for converting all recorded data to make it compatible with the agreed standard file formats described before. After the upload, all data streams of a multimodal recording are stored in the server, in a folder's structure defined by the user and are accessed from the client through a web based server's front-end interface that connects to a structured MySQL database which contains links to the actual recording data files.



3.4 Back-end server

Once the *datapack* is uploaded to the server, then some processing tasks are executed automatically in the server to prepare the *datapack* in order to be ready to be visualized and shared among other users of the scientific community. The back-end server is the responsible for the database management (user rights and storage/retrieval of *datapacks* data) and to execute *processing data services* over the *datapack*'s multimodal data.

3.4.1 Database structure

The multimodal recordings database consists of a repository of *datapacks*, organized in several used defined folders containing a large number of data files plus the original Skeleton XML file. For all *datapacks* uploaded to the server, an entry in a mySQL structured database is created for managing all the streams of data. The mySQL database contains (in a set of tables) a duplicate of all metadata extracted from the XML Skeleton file and a link to the local path of each file in the server storage. Implementing the repository as a mySQL database makes much easier for server management and access to the multimodal data files taking the advantage of using SQL queries.

The *datapacks* are structured in folders and each folder and *datapack* has associated the user that created and uploaded the multimodal recording. Having all the data in a structured database makes it easy to build a client interface that connects to this database allowing to navigate through the recordings in a user friendly way, featuring sorting, filtering, selecting by type and search capabilities over the multimodal recordings.

The server management of the *datapacks* database and the processing of the client queries are implemented in JAVA on top of an Apache Tomcat installation over a Linux Platform. For a user's retrieval of data, through an HTML5 web interface, the server receives asynchronous queries from the client using AJAX and a WebSockets connection, gathers the corresponding data from the database and responds back to the client with the demanded data.

The user rights management is also done using a structured database containing all usernames and passwords linked with the rights and ownership of each user over the server *datapacks* and folders. Each *datapack* is owned by the user that uploaded it. The owner of a *datapack* can set up who can access, view or modify each *datapack*.



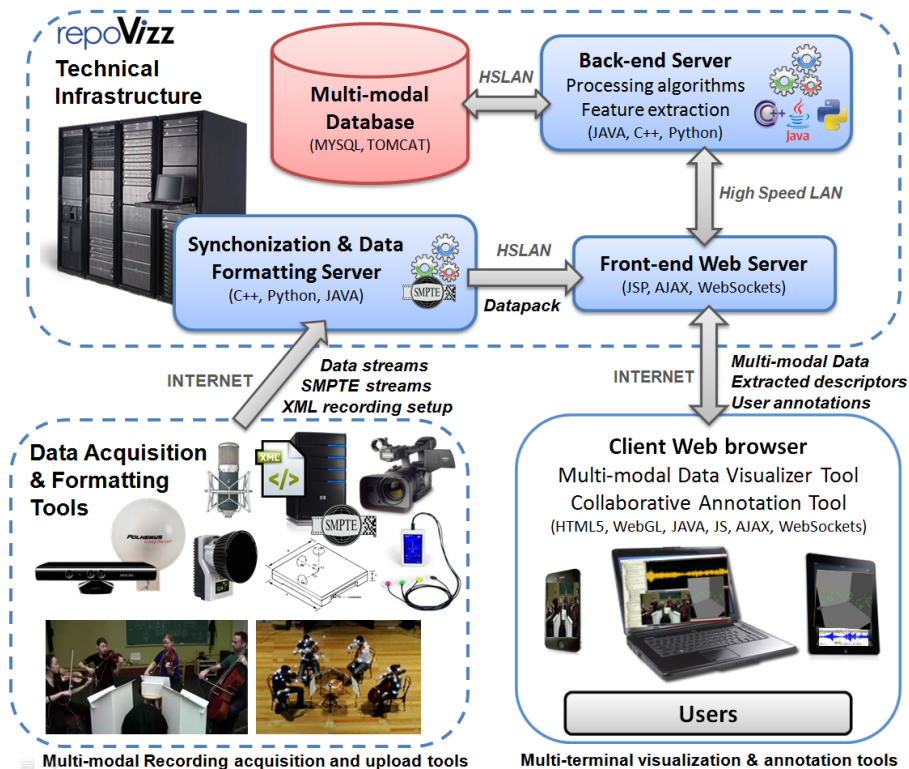


Figure 9. Diagram of the repoVizz Framework modules

3.4.2 Back-end data processing

Once a *datapack* is uploaded to the repository, the back-end server stores the data in the structured database. There is also some processing of data that happens in the server. This processing is implemented as processing algorithms that run in the server for certain types of streams. Algorithms are processed over specific streams and the results of each algorithm are added to the structured database (and to the XML skeleton file) as if these streams were additional streams of the original multimodal recording. The newly generated data can also be shared and visualized (in the same way as any other stream) using the web client interface explained in next section.

The idea of server data processing is a service that the repoVizz framework offers to the research community, where any user can upload customized algorithms that perform a processing over some specific data streams and generate as output another stream that will be added to the multimodal recording. Users may upload processing scripts or binary executables to the server and specify using a menu which kind of data files (from a *datapack*) are eligible for this kind of processing and how the output files should be added to the xml skeleton file. The current implementation does not yet allow users to create new processing algorithms to be executed in the server over particular data streams. Nevertheless, as a provisional solution, some algorithms have been implemented already in the server for the specific deployment of repoVizz as the main framework for the SIEMPRE repository.

3.5 Web-based front-end interface

One of the most interesting features of repoVizz is its web-based front-end interface. Users of repoVizz can easily navigate through the repository of multimodal data, browse datapacks, interactively visualize desired streams, download data segments, add/edit annotations, call data processing services, customize the data views, etc. All of the above is achieved using a web-based interface developed using state-of-the-art HTML5. Two main components empower meaningful navigation and retrieval of data: the repository navigator and the datapack visualizer.

3.5.1 The Repository Navigator

Within the repoVizz portal, the repository navigator manages user access, data browsing and uploading of datapacks. Once users are authorized (via secure user password management) to use the repository, two main tabs respectively offer (i) uploading of datapacks, and browsing of existing data (see Figure 10).

Datapack uploading is carried out by selecting the *datapack* compressed file (previously created with the Datapack Designer Tool) from the user local machine. The user can provide a descriptive text for the *datapack*, and a logical folder (in the server) where the multimodal recording will be accessible. Once the *datapack* is uploaded by clicking the “upload” button, the compressed file is automatically processed by the server, making it available for browsing, visualizing, editing, or downloading.

When a user logs in repoVizz and goes to the “Browse” tab in the main menu, an intuitive interface enables navigation through existing *datapacks* (see Figure 10). For each existing folder/*datapack* (or folders/*datapacks* to which the user has rights to access), some details (including name, owner, description, last access, etc.) are displayed as to ease navigation.

While navigating through the “visible” *datapacks*, the user can select one of them to (i) retrieve more detailed information about the data streams contained, (ii) delete it (if the user has enough rights), or (iii) to go ahead and start visualizing the *datapack* through the visualizer (see next Subsection). User-defined custom views (will be explained below) can be selected, so that the data visualization layout accommodates the user when exploring, annotating, etc.

Name	Description	Author	Folder	Date	Views	Actions
testDP	no description	unige	default	19 Mar 2012 18:14:10 GMT	default	View 10 secs 8 Delete Show Info
sarc_audio_test	no description	mtg	default	19 Mar 2012 18:14:10 GMT	default	View 10 secs 8 Delete Show Info
EUE1_ensemble	no description	mtg	default	19 Mar 2012 18:14:10 GMT	default	View 10 secs 8 Delete Show Info
pablo2	no description	mtg	default	19 Mar 2012 18:14:09 GMT	default	View 10 secs 8 Delete Show Info
UE2_ensemble_audiodesc	no description	mtg	default	19 Mar 2012 18:14:09 GMT	default	View 10 secs 8 Delete Show Info
UE2_Solo_violin1	no description	mtg	default	19 Mar 2012 18:14:08 GMT	default	View 10 secs 8 Delete Show Info

Figure 10. Repository Navigator showing available datapacks.

3.5.2 The Datapack Visualizer

One of the most interesting and powerful features of the repoVizz framework is the web based datapack visualizer. The visualizer offers a fully fledged web application using state-of-the-art HTML5/Canvas features for visualizing all data streams contained in a multimodal recording,

featuring a graphical representation of the datapack's Skeleton XML with the setup of the recording (on the left side), and several panel views for visualizing the data streams and all other associated information.

Viewer panels are used for visualizing almost any data file contained in a *datapack*. Some examples of data streams that can be visualized in a data stream panel are audio signals, data streams from position sensors (x, y, z), physiological sensor data (EMG, EKG, EMG, EEG), accelerometer streams, instrumental gesture descriptors, audio frame descriptors, etc. These panels are very similar to the panels of any audio analysis software (like sonic visualizer, audacity, audition or sound forge). In these panels the user is able to zoom in the time and value axis and navigate forward and backward through the time axis. A tooltip shows the individual units and range of values for each stream when the user moves the mouse. Each panel can hold multiple streams with multiple colors and the user is able to fully configure the view. All streams and panel views can be synchronized so the user can zoom all the signals simultaneously or go forward/backward to a certain time offset preserving that all signals are aligned. Audio streams displayed in a view panel can be listened to and multiple-audio streams can be played in sync simultaneously.

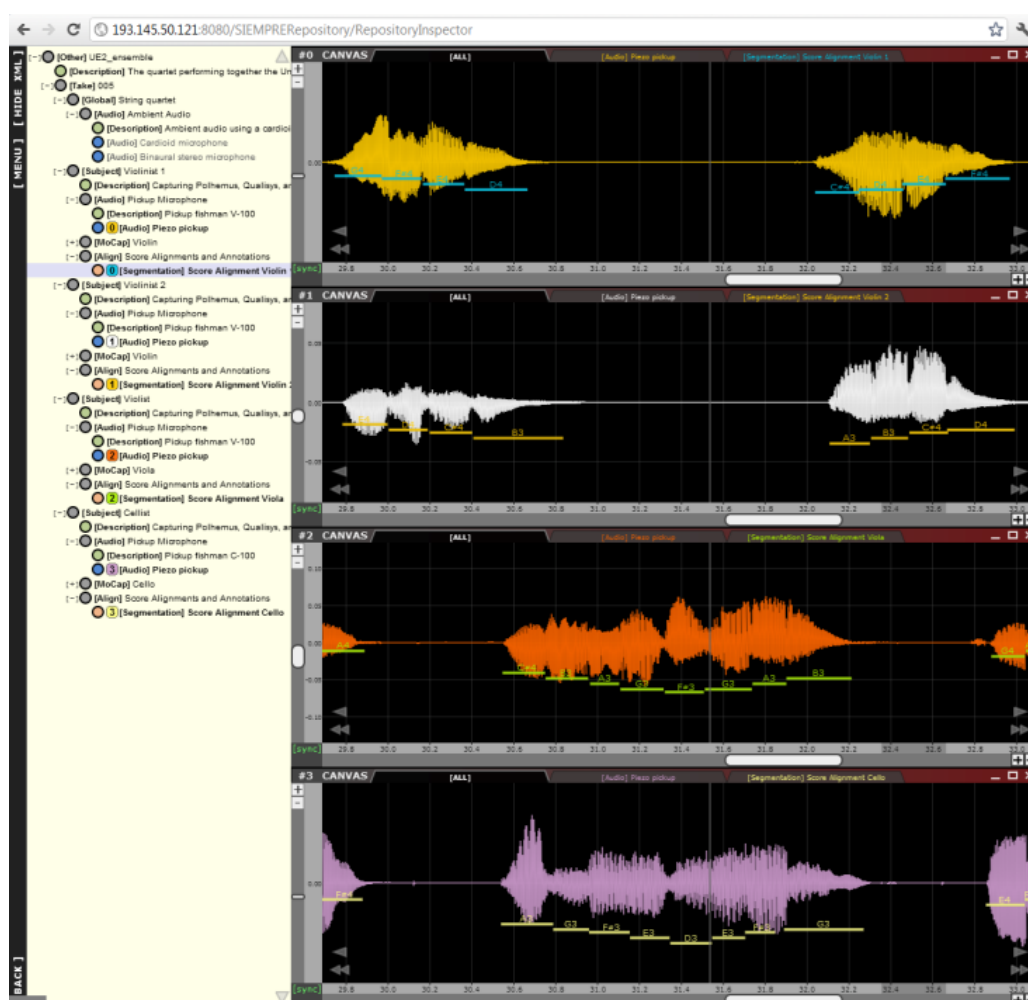


Figure 11. *repoVizz's web based multi-modal data visualizer.*

Video panels are used to display video streams contained in the multimodal recording datapack (see Figure 12). For the visualization, all the previously uploaded videos are converted on the server to a lower resolution version mp4 & webm formats to be able to be streamed to the web browser client. The user visualizes the low resolution videos but in case that he wants to download a segment from the server, then the original video is chosen in order to get the original quality. Videos are reproduced in

sync with audio and all other streams. Video panels can contain any type of video: videos of a performance, termographic camera videos, 3D renders of the moCap data, etc.

3D visualization panels allow playback and visualization of acquired motion capture data (e.g. Qualisys 3D spatial markers) allowing interactive control to change the view camera angle and zoom level using the modern browser based WebGL technology. Other visualization panels are in progress to be able to visualize document files, static images or musical scores.

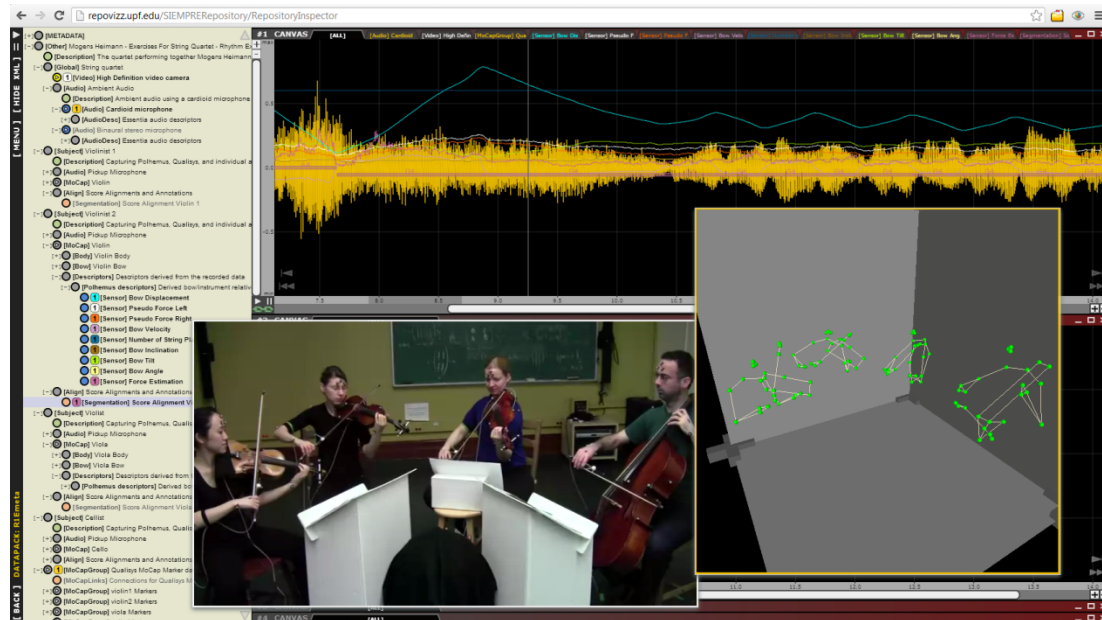


Figure 12. *repoVizz's web visualizer showing video and 3d visulization of Motion Capture data.*

3.5.3 Usage of the Visualizer

Once a multimodal recording is selected from the database using the repository navigator (Figure 5), then the *datapack* visualizer tool (Figures 6 & 7) is executed and allows viewing with detail all associated data acquired during the recording session and additional server processed data using the viewer panels detailed before. The user just drags the data nodes from the XML skeleton tree view (on the left) to the corresponding data view panel (on the right) and the stream is displayed accordingly. To avoid memory and bandwidth problems for large amounts of data, all data files remain in the server and are automatically retrieved by the user on demand depending on the segment that is viewing and the level of zoom for each data stream. The communication between the server and the client is made through asynchronous AJAX and WebSockets calls.

The user can fully customize how and where to view each data stream and change the layout and colors for each individual stream, changing the number of panels, the zoom levels, etc. All these customizable layout properties can be saved as a “view configuration” for a particular *datapack*. Once a “view configuration” is stored in the server, for future visualizations of the *datapack*, the user can select (from a drop down list) any of the saved configurations and the *datapack* is visualized with the same layout configuration that was saved.

One of the main advantages of this framework versus other existing multimodal databases is that it allows having a look at the data before downloading it, avoiding the hassle of building offline tools or parsers to read the data before knowing if it's useful or not for our research. Using the *repoVizz* framework users can graphically select a segment of the recording (or the full recording if desired) and just selecting the “download *datapack*” option from the menu, the *datapack*, containing all data streams within the chosen time segment, is downloaded to disk. The downloaded *datapack* will also

contain Matlab scripts prepared to import directly all data streams to Matlab to help the researcher in the analysis process.

