SIEMPRE XML Creator User's manual

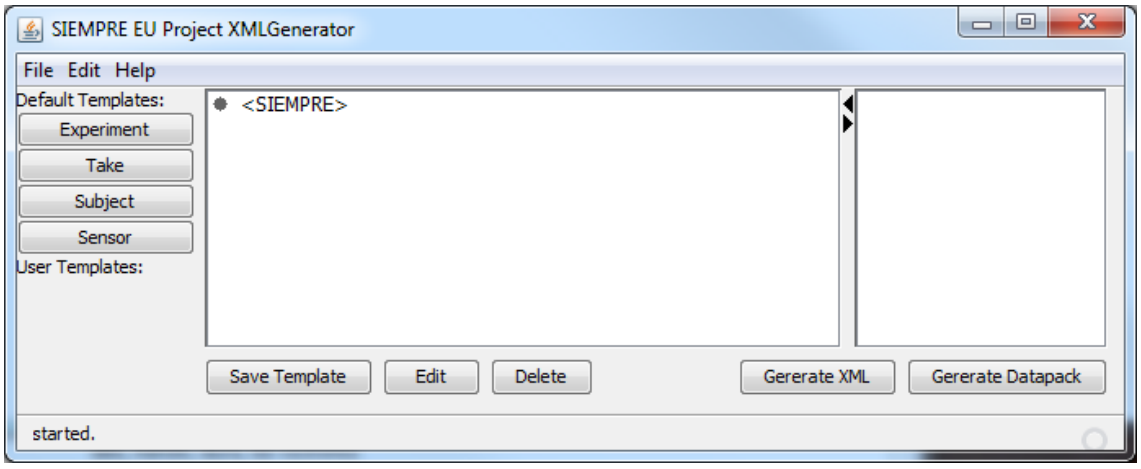
The SIEMPRE XML Creator is a Java tool for defining the setup of multimodal recordings. This tool allows any inexperienced user to create a layout for a multi-modal recording experiment, following the SIEMPRE rules and formats defined during the project. The tool is based on an intuitive GUI where the user designs the structure of the experiment following a set of node templates (Experiment, Take, Subject, Sensors, etc) and visualizes the structure of the experiment as a tree. The generated XML contains links to data files containing streams of formatted data generated during a multimodal recording session. The structured xml file together with the data files form a "Datapack" which defines a multimodal example and can be uploaded to the repoVizz server for sharing or visualizing purposes.

In order to use the XML Creator, first of all open your preferred internet browser. You will need to have java installed.

Open the address: http://repovizz.upf.edu/xmlcreator

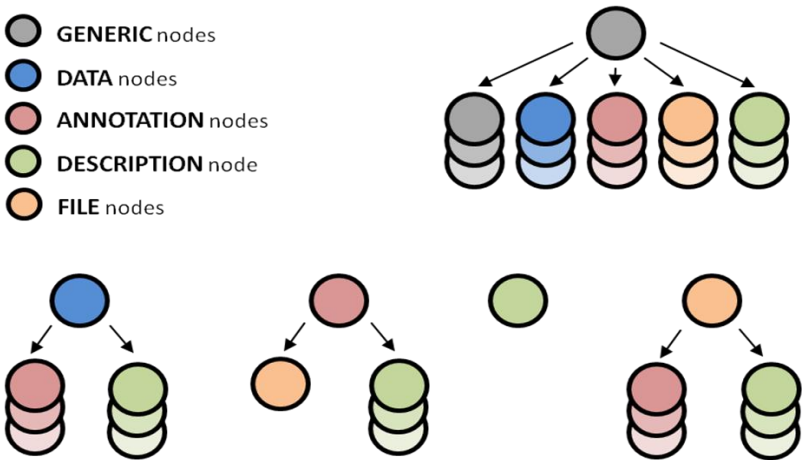
Then, open the java application by pressing the launch button. The browser will prompt with a security alert, allow it in order to grant access to your computer for the application, otherwise you will not able to run the application.

Once the application starts it will ask the user to select the project directory, you need to specify the folder where the synchronized data streams recorded during the multimodal recording actually are present or will be copied afterwards (you need to have read/write permission on that folder, please ensure this!). Once the folder is selected, the application main window will appear.



The middle window contains a visual representation (as a tree with nodes) of the xml file that defines the multimodal recording setup. When opening the application the xml is empty, to add new nodes, the user can right-click with the mouse in the nodes to add subnodes choosing the category for the node and filling its attributes.

The following figure shows the taxonomy of nodes that can be used to create the xml tree:



As you can see, each node class is represented with a different color, and some nodes can only accept as a child certain node classes and also some nodes cannot be child of any node except themselves, as it happens with generic nodes.

Generic nodes represented with the grey color are usually used for defining logical/physical groupings of data annotations and can have as children any type node. Their attributes are the Category (e.g. Take, Subject, Experiment, Audio, Body, MoCap, Violin, Bow, etc) and a text description of the node.

Data nodes represented with the blue color are used as the container for continuous (numeric) data streams (e.g. acquired signals, computed descriptors, etc). Each node can only contain one single-channel data array, for multiple channels, many data nodes can be grouped within a generic node. These nodes only accept as children description nodes or annotation nodes. Their attributes are the Category (e.g. Audio, Sensor, Descriptor, etc), a text description of the node, the path to the file containing the data stream and some attributes extracted automatically from the data file. All data files are stored with the same format, we have adopted the Broadcast Wave Format, which is a binary format that allows saving continuous data streams with different sampling rates, storing each sample as a 32bits IEEE float. This format, which is basically a wave file + and iXML with some extra data, can be read in many audio editors (e.g. audacity).

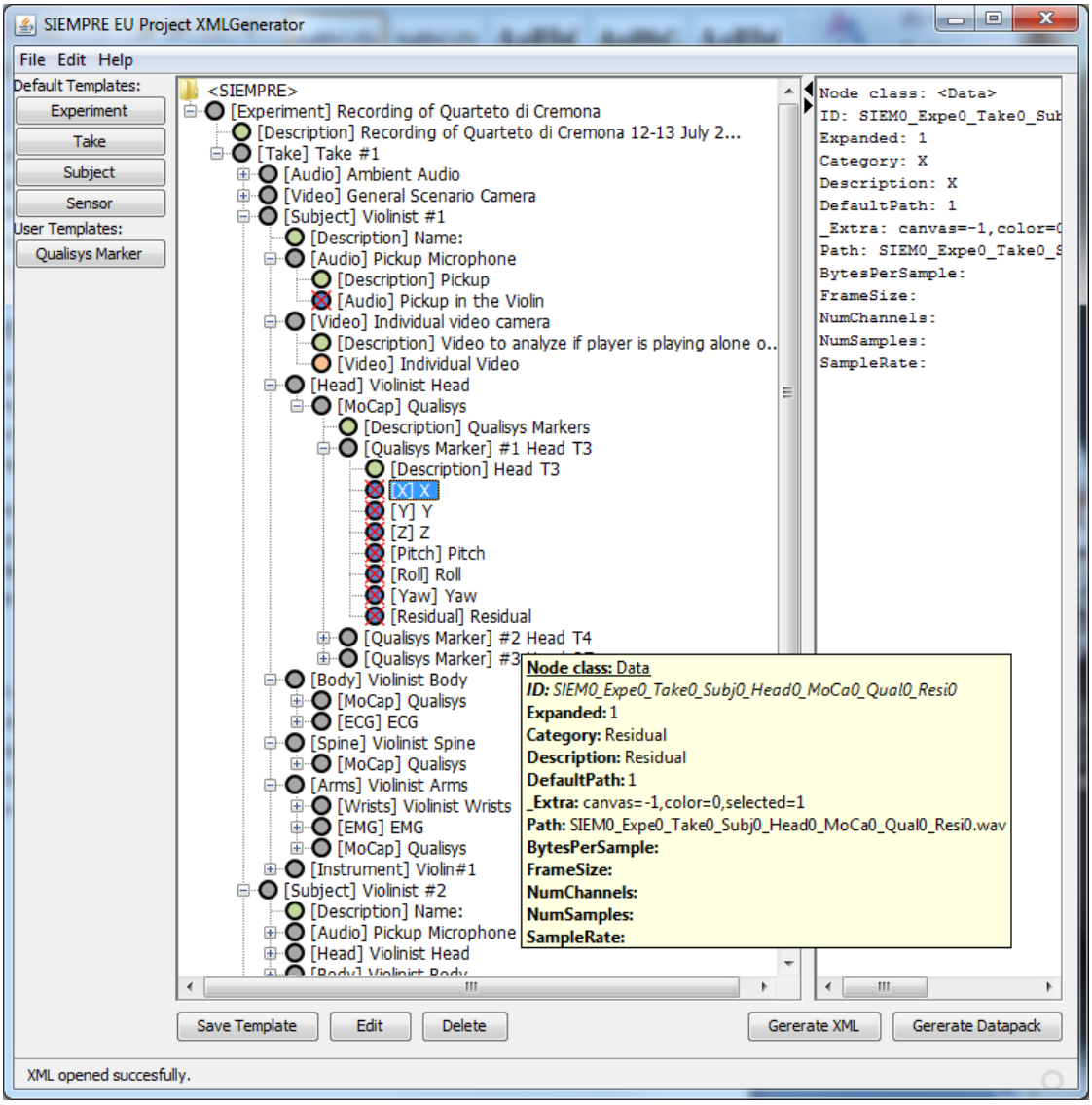
Annotation nodes represented with color red are used for holding user annotations. The user annotations can be stored in two different forms: Segment based annotations in the form <InitTime, EndTime, Label> or event based annotations in the form <Time, Label>. The attributes of the annotation nodes include the Category (e.g. Comment, Segment, Event, etc), a text description, the BeginTime, EndTime and Label. These nodes accept as children a description node or a File node containing a txt file with a bunch of annotations, one per line in the form BeginTime, EndTime, Label.

Description nodes represented with green color are used for holding text descriptions. These nodes do not accept any children. The attributes of these nodes are a Category that can explain which kind of description is stored in the node (e.g. manufacturer, location, body part, etc) and the actual description text.

File nodes represented with orange color are used for holding references to additional/relevant files with any data not covered within the rest of node classes. For instance video files, score files, images, etc, can be contained in these nodes. This node only accepts as children description nodes.

Creating the xml structure

Once that we know which is the taxonomy of nodes that we can use for defining a multimodal recording setup we can start creating a tree structure with generic nodes and its children, it's as easy as right-click with the mouse in the nodes to add subnodes choosing the category for the node and filling its attributes. To start from scratch, we can add child nodes to the <SIEMPRE> parent node. We can also use some copy/cut/paste and moving up/down nodes for easier and faster editing of the structure, especially when for instance we need to replicate many times generic nodes containing the same child nodes inside, all these options are available in the context menu that appears when right clicking in an existing node. The GUI also allows to collapse/expand the nodes that have children on them, just pressing the [+]/[-] sign at the left of the node.



The user can save templates of a particular structure of nodes for later use, just right-click with the mouse on the parent node of the structure and select the option “save as template”, the template will be saved locally in the working folder the user has selected at the beginning (when the program was started) with the name specified in the category of the parent node. All the saved templates that are stored in the current working folder will appear at the left panel below the “User Templates:” label, so the user can use them at any time just selecting the parent node where the user wants to append the template as a child and clicking in the button of the saved template.

When creating a data node, the attribute path, which refers to the name of the file containing the data stream, is automatically generated by the software, giving as name a concatenation of each parent node category name (the first four letters plus a number), to form a unique filename (.wav) that cannot be repeated within the same xml. This automatic generation of the name can be overridden by the user when selecting an already created file in the local disk. The advantage of using automatic filenames is that when doing the recording session or post synchronization of the files, the xml (generated with this tool) can be used as input to generate the data files with the correct names and then avoid the hard task of manual file renaming that can yield in many errors.

As it can be seen in the previous figure, some of the data files (blue nodes) appear with a red X covering the blue circle, this means that the software cannot find the data file specified in the path attribute of the data node, once the file is created/recorded and copied to the working folder, the red cross will disappear automatically, the cross it’s there just to alert the user that the file is not present in the current working folder.

Saving the XML/Generating the Datapack

Once the user has finished creating the xml, in the bottom right corner there are two buttons to save it. The first, Generate XML, only saves the xml file containing the nodes and its attributes to the current working folder, the Generate Datapack button, apart from saving the xml, it also generates a zip file with the xml file and all the associated data files, score files, annotation files, etc that are linked to the xml. This datapack file (.zip) will be used to upload the multimodal recording to the RepoVizz server (http://repovizz.upf.edu).