Perspectives in the development of audio databases at SEL

1.Introduction

This document is intended to describe perspectives in the development of audio databases at SEL.

We will start by analyzing user scenarios for audio databases at IRCAM. Some of these scenarios are only partially covered by the current state of the Sound Palette, some aren't covered at all.

Two evolutions are foreseen and described hereafter:

- First, a short term evolution, in order to make the SoundPalette functions available for scientific teams, and to become operational for projects like Orchestration, but with some limitations due to the actual architecture of the database (MPEG7).
- Second, a long term evolution, which will extend the functionalities of the database to non- MPEG7 schemes, and will allow a distribution of resources and databases.

2.User scenarios for audio databases at IRCAM

Music producer:

M1 - I need to find a set of sounds for a work

M2 - I want to apply the SP functions on my own sound collection

M3 - I have a sound collection and want to publish it to others

Researcher:

R1 - I need a set of sounds for a work

R2 - I want to add my own descriptors to the database and use them locally

R3 - I want to add new descriptors to the database and make them available to other users

R4 - I want to publish a new description algorithm (Matlab based)

R5 - I want to apply the SP functions on my own sound collection

Valorisation:

V1 – I want to put a set of sounds on DVD and distribute it

SEL:

S1 – Import a set of sounds into the database.

S2 – Integrate a new extractor.

S3 – Export a sound collection with a management tool for a DVD.

The next paragraphs contain an analysis on how the current Sound Palette 2 responds to these user

scenarios, what are the problems and what we plan to do.

M1 I need to find a set of sounds for a work

Current state

Option 1) The user can connect to the Sound Palette and explore how sounds are organized using the Organize tab. He can search sounds by full-text search, by pitch, by values on numerical descriptors and by similarity to a target sound. If the target sound is not on the database, he has to upload it before and, if he has the permission, launch the extractors on it. After the sounds have been found, the user can put them in his shopping cart and download them.

Opt. 2) The user can use the SPMatlab – or SP2Max - tool to connect to the Sound Palette and find sounds by full-text or descriptor values. Once the sounds have been found, he can use them directly from the mounted SDB hard disk.

Evaluation

Users are generally satisfied with the search features, except those involving numerical descriptors since they are poorly explained and since you must have special permissions in order to extract them on your sounds. This has an impact especially on the usability of the search by similarity feature.

Other problems have been raised regarding the sound quality and channel structure, but this does not really depend on the system architecture.

Some users still feel that Mac OS' finder is a far better way to find sounds in a large collection, since it is more handy and you can drag-drop your sounds directly to your hard disk or application.

Perspectives

1) Simplify the descriptor structure. By now the descriptors are Mpeg7 compliant (with some extension). This is a limit in two senses: a) they are pretty complex and hard to manage b) there are only two descriptor types (scalar and vector) you can generically apply to a sound, and this is a limit in the sense that all others (title, instrument etc.) are not really extensible, nor do they share the same logic.
2) Conceive more specific descriptor types for pitch, float and int vectors, dynamic and so on with specific search patterns. This will add power and ease of use to the search features.
3) Document the descriptors by giving a meaningful label and a help text. This solution together with point (1) and (2) allows to provide a much cleaner user interface, a simpler, yet more powerful, search logic and far better performances.
4) Let the user have a personal database where he can extract the numerical descriptors without the need of a special permission. The permission is needed on the centralized database for both server charge problems and data quality. On his own data and with his own computer, the user can do whatever he wants.
5) Integrate the system in the musicians work environment as much as possible. There is already a Max/MSP client for the current Sound Palette. This should be maintained and upgraded together with the other system components. An Open Music plug-in is also foreseen.
6) Switch definitely to an external audio file linking technique rather than to store them in system folders. This makes the system less monolithic and more inconsistency-prone, but the users can access the files directly on a mounted disk or even on DVDs without having to be on-line. This is very important since the file based access is still the most used technique to manage audio files and people don't want to drop it.

M2 I want to use the functions offered by the SP for my own sounds

Current state

The user has to upload all his sounds in order to benefit from having them stored in a database (he can search by full-text and, if he has the right to do it, launch the extractors and find by descriptor or similarity).

Evaluation

Users seem not to be satisfied by this solution, since nobody has ever used it. This is also quite understandable, since you need special permissions to launch the extractors and since, anyway, you have few usable features after. For example you can not use the extracted descriptors in your environment directly for re-synthesis or other processing. When you change your sound you have to re-upload it and re-extract the features etc.

Perspectives

1) Allow the user to have his own system locally (the local database is already foreseen by M1.4). This allows to easily use the sound database features on his own computer, on his own sounds and on his own database.
2) Allow the user to access the on-line database and his own database at the same time. This allows the user not to drop one big feature of the centralized server that is its sound content. The user can compare his own sounds to those on the centralized server and to make queries involving both.

M3 I have a sound collection and want to publish it

Current state

The user must contact the SEL team and ask them to import the sounds in the database.

Evaluation

This solution seems to be satisfactory, a least for the moment.

R1 I need a set of sounds for a work

Current state

Opt. 1) The user can connect to the Sound Palette and explore how sounds are organized using the Organize tab. He can search sounds by full-text search, by pitch, by values on numerical descriptors and by similarity to a target sound. If the target sound is not on the database, he has to upload it before and, if he has the permission, launch the extractors on it. After the sounds have been found, the user can put them in his shopping cart and download them.

Opt. 2) The user can use the SPMatlab module to connect to the Sound Palette and find sounds by full-text or descriptor values.

Evaluation

Same problems seen on M1.

Perspectives

The solutions foreseen for M1 already address most problems raised here. In addition we foresee to:

1) Maintain the SPMatLab API in order to reflect the new system features.

R2 - I want to add my own descriptors to the database and use them locally for my own work in conjunction with descriptors already in the database

Current state

User shall implement its own solution – nothing is implemented in the SoundPalette to this end.

Perspectives

1) Allow the user to create new descriptors on his local database. Since this happens on the private user database, it doesn't pose any problem of server charge by heavily create, remove, alter descriptors. Nor does it pose problems of managing the descriptor visibility.
2) The simplified descriptor structure (M1.1) will also be of ease in defining new descriptors.
3) Since the user can access both local and centralized databases (M2.2), he can compare the meta-data stored on both.
4) Allow the user to import meta-data from the remote database. This is useful if the researcher wants to work off-line (no network connection, better performance, no server charge).

R3 - I want to add new descriptors to the database and make them available to other users

Current state

I shall contact the SEL team, and ask them to integrate my descriptors in the database by extending the schema and importing the descriptors.

Evaluation

From the researchers point of view (SEL's will be examined later) this still involves some work, since probably there are some adaptations to do on the code, the output format etc.

Perspectives

1) Since the researcher can develop his descriptors directly in conjunction with the system (R2), they are inherently compliant with the meta-model.
2) The local system can store descriptors pointing to the sounds on the centralized database (M2.2).
3) The user can export the descriptors from his local database to the centralized database, or (see R4) add the algorithm to the centralized database and let it run.

R4 - I want to publish a new description algorithm (Matlab based)

Current state

The user must contact the SEL team and ask to integrate the new extractor (see scenario S2).

Perspectives

1) The system will be able to integrate extractors running locally as executables or remotely as web services and outputting a compliant XML file. Once an algorithm has been developed, it is sufficient to make an executable (compiled Matlab) out of it and it will be ready for integration.

R5 - I want to apply the SP functionalities to my own sound database.

Current state

Not feasible.

Perspectives

This scenario is very close to M2, and will be implemented the same way, maintaining the SPMatlab API (R1.1).

V1 – I want to put a set of sounds on DVD and distribute it

Current state

Opt. 1) The user can use the on line Sound Palette to collect the sounds and download them.

Opt. 2) The user has some more need (like putting on the DVD also a basic management tool for the sound files, for example) and must thus ask the SEL team to do the job.

Evaluation

The first option has never been used, the second one is acceptable (see S3 for details).

S1 – Import a set of sounds into the database

Current state

We have to generate an Mpeg7 file describing the sounds and a simple SQL file describing the rights on these sounds. This might be simple or difficult, depending on what information we have and on what information is expected to be stored in the database. Once the files are generated, they are imported into the database. Depending on the size of the collection, this importing process can take from few hours to almost a week. After we have to launch a batch file if we want to extract the descriptors.

Evaluation

This step is quite difficult each time, due mainly to the complexity and limits of Mpeg7.

Perspectives

1) The simplified descriptor structure (M1.1) helps a lot
2) We foresee a tool that browses a directory and imports the sounds directly in a database.

S2 Integrate a new extractor.

Current state

We have to dive into the code of the current extractor module (which was not written by us), and add the extractor to it. We would also have to run manually the extractor on the audio files already present in the database, in order to avoid rerunning all extractors on them just to have the new one.

Note: extractors that can be integrated are only extractors outputting something compatible with the mpeg7:AudioLLDScalarType or mpeg7:AudioLLDVectorType, which is limited to global numerical descriptors or to (scaled) instantaneous ones. For other descriptors we would have to upgrade the whole application.

Perspectives

1) In the eventuality of completion of user scenarios R3 and R4, we just have to reference the extractor in the database.
2) It will be possible to run a single extractor on all audio files, as foreseen by R3.3.

S3 Export a sound collection with a management tool for a DVD.

Current state

We have to do some manual work, whose complexity depends on the requested features. It would be quite easy to produce an HTML page listing all the sounds on the DVD, for example.

Perspectives

1) Since R3.3, the system can export the descriptors for a sound collection as a new database.
2) The system can connect to several local databases at the same time, so a DVD will be automatically readable and usable by any client application (SPMatlab, SPMaxMSP, SPOpenMusic).

3.Short term evolution

In a short term view, we intend to finalize the different tools for accessing the SoundPalette from the scientific environments. In this view, several tools are developed which allow access to the SoundPalette from Matlab, Max, and OpenMusic.

The actual state of these tools is as follows:

- Access from Matlab (SPMatlab) : available.
- Access from Max (SP2Max) : available
- Access from OpenMusic (SP2OpenMusic) : currently developed.

The main functions to be added to these tools in order to make them fully functional for projects like Orchestration are:

- Creation of descriptors. A limitation is that the descriptors shall be MPEG 7 compliant.
- Controlling access to newly created descriptors (and particularly, hiding these in the SoundPalette online).
- In addition, it is foreseen that some evolutions of the query language will be asked by the scientific teams.

The scenarios covered by this evolution will be scenarios R1, R2 and R3.

Regarding the scenario R4, we envision to cover this by extending the metadata scheme, attaching to newly published descriptor the adress of a service (probably in the form of a Web Service).

The service will accept in input the adress of the sound to be analyzed, and return the corresponding MPEG7 compliant description.

All these evolutions – with the exception of evolutions of the query language - are envisioned to be available in short term, that is, for October 2005.

4.Long term evolution

The current architecture has two major drawbacks:

- The use of Mpeg7 which makes it complex to use the system, to extend it and that has also some impact of the performance.
- Everything happens on a centralized database. We have to carefully check permissions, and using the system features for tests or local sound files is quite a pain.

On long term we plan to change both of these points, adding a local database in Sqlite3. Sqlite3 runs on all platforms with almost zero-installation, it is robust and well performing, as stated also by the adoption of it as part of the next Mac OS release. This database will be compliant with the centralized server and can be used by researchers as well as by musicians in order to build a full featured local system that can be used as stand alone or in conjunction with the centralized database. A very interesting feature of Sqlite3 is that it can link several databases together. This allows the user to search at the same time on his working database as well as on a database on a DVD etc. without any development overhead for us.

The switch to audio file references based on their physical location on the file system allows to reference on a local database files that are referenced also by the centralized database. This leads to the possibility of storing new descriptors, or even editing existing descriptors about the sound files of the centralized database without affecting the database itself.

The new database schema will be a user scenario driven, instead of being an implementation of Mpeg7. We strongly believe this will improve both usability and performance.

Implementation plan

There are two main steps in this evolution :

1) Developing local databases
2) Ensuring interoperability between local database and the centralized database.

The RT team is interested in testing the system and developing features on top of it in Max/MSP. For this reason we foresee Max/MSP as our test platform. All implemented features will be first of all accessible in Max/MPS.

Developing local databases

Step 1: SOL on Sqlite3

The first step for us is to implement the read-only features on a Sqlite3 database containing the recently up-cleaned version of the SOL audio files. These features should be available in Max/MSP. The database file would be accessible in read only on the SDB server, as well as all the audio files.

We believe that this is already an interesting step, allowing the users to better and easier access the SOL files inside a musical environment. This tool can be used by music assistants as well as by the students searching for instrumental sounds.

The new SOL collection has anyway to be imported into the current Sound Palette, and we can, with almost no overhead, use the same process for generating both, an Mpeg7 file and an Sqlite3 file ready to be used by this system.

For this step we foresee one MM of work.

Step 2: RW features on a local database

A second step would be to implement read/write features on a local, user maintained database, integrated with the SOL tool of step 1 (this would be the version of SPMaxMSP).

This should allow users to manage SOL files and possibly also files coming from other sources, re-editing the descriptors if he wants, adding new ones he needs for the current work etc.

This step is not too big, since the database already embeds somehow the low level functions for it.

For this step we foresee one MM of work.

Step 3: Test, bug-fixes, user feed-backs

At this point, and without too much effort, we have many elements for the evaluation of the new system. The next steps are broader and have no time indication, because they depend a lot on what emerges during this phase.

Ensuring interoperability

Step 4: Upgrade of the centralized server

Let the centralized server communicate with the local databases. This is definitely a big step.

This will allow users to access not only SOL, but the whole corpus stored on the centralized server.

Step 5: Tools for import/export of sound collections

Create tools that import a sound collection from a local user database and to generate DVDs including sounds and an Sqlite3 file.

This feature could also be integrated on the local system, allowing users to easily make back-ups of their work, having a database file on the CD or DVD that contains all information about the sounds, and that can be simply loaded and queried later.

Step 6: Upgrade of the extractor mechanism

This will allow to manage executables and web services on both server and client side containing extraction algorithms whose results are stored on the database.

Step 7: Upgrade the SPMatlab and SPOpenMusic modules