



**HAL**  
open science

## Our research for lost route to root

Alain Bonardi, Jérôme Barthélemy, Raffaele Ciavarella, Guillaume Boutard

### ► To cite this version:

Alain Bonardi, Jérôme Barthélemy, Raffaele Ciavarella, Guillaume Boutard. Our research for lost route to root. ICMC'08, Aug 2008, Belfast, United Kingdom. pp.1-1. <hal-01161065>

**HAL Id: hal-01161065**

**<https://hal.science/hal-01161065v1>**

Submitted on 10 Jul 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# OUR RESEARCH FOR LOST ROUTE TO ROOT

*Alain Bonardi, Jérôme Barthélemy, Raffaele Ciavarella, Guillaume Boutard*

IRCAM

{alain.bonardi, jerome.barthelemy, raffaele.ciavarella, guillaume.boutard}@ircam.fr

*Alex Mikroyannidis, Bee Ong, Kia Ng*

ICSRiM - University of Leeds

{alex, s.b.ong, kia}@icsrim.org.uk

## ABSTRACT

*In contemporary arts many non conventional instruments, devices and technologies are used, so as many new ways of man-machine interaction, to produce new artistic works. When introducing electro-acoustics, artists did not realize the risk not to be able to re-perform their works just a few years later. In the framework of CASPAR European project, the On Line Services team at IRCAM proposes first results of the research, conducted together with ICSRiM team at University of Leeds, for trying to save artistic works from obsolescence and to support future developments.*

## 1. INTRODUCTION

### 1.1. The root

During centuries music was written on paper and a little part of what needed to re-perform the author's intention was outside that paper.

Many kinds of performances, like jazz or folk, are more styles than coded music.

This lets us consider that also social and cultural environments, in which music is produced, have to be studied in order to keep safe the tie to its origin.

In this case, re-performance possibilities are not crucial for the purpose of preservation and recordings, together with social environment trackings, can fully satisfy the need for documentation [1].

At about the middle of the last century a new era for arts, and especially for music, started.



**Figure 1.** Francois Bayle performing at Acousmonium, invented in 1974.

Many experiments were done on the structure of the sound and time, and really new possibilities were offered to arts.

Due to the rapid growth of technologies, electronic devices became affordable and available not only in specialized structures but even to home end-user.

Computer and software permitted all people to have high quality virtual instruments as well as completely new ones.

The use of hardware and especially software devices introduced a great degree of freedom for composers and artists, but the price payed was the very short life time of those devices and the wide spread of used solutions.

Considering electronics as musical instruments and modeling an interaction with human in artistic works, a new paradigm of notation was also required.

While in traditional notation the meaning was written and final effect was coded in the score, for complex contemporary works there is no adequate notation and the only saved data are the specific way used to obtain the result (the process itself) but not the intentions.

Because of the very quick changes in technology and market devices, now we live also a paradoxical situation: electro-acoustic works, that are very recent, are facing serious problems [2] not present in traditional works.

The core problem seems to be time dependency: the most difficult to be solved.

### 1.2. The route

Preservation of music has been studied for many years on a wide range of manuscripts and instruments, while the sustainability of live electronics is a recent interest, more complex and with a lot of missing informations: more and more difficult to obtain.

Because of this complexity and of many scientific underlying issues, adequate new methodologies have to be applied.

A European project named CASPAR<sup>1</sup> (Cultural, Artistic, and Scientific Knowledge for Preservation, Access and Retrieval) has been launched in 2006, bringing together 17 partners, including IRCAM, on the general topic of preservation of digital data.

This project intends to address three different communities, by developing three different *testbeds*: one for scientific knowledge, one for cultural heritage, and

<sup>1</sup> <http://www.casparpreserves.eu>

one for performing arts.

Inside the performing arts scenario we developed a base architecture on which to build stand alone open tools, specialized for analysis, feature extraction, documentation and monitoring

The target of our research are software modules, used for various activities as signal processing or symbolic calculation, for production and performance common to all arts (music, dance, theatre, video, interactive installations, etc).

## 2. STRATEGIES FOR SUSTAINABILITY

Either hardware or software, active modules provide control and signal processing functions.

Nowadays musicians use intensively graphical languages, like Max/MSP, PureData [3] or Reaktor, to implement almost any required process.

Several strategies have been evaluated to face sustainability and re-performance requirements of works based on those environments.

From one hand, the simplest way to make an artistic work re-performable is to keep safe all what was used in the first performance, including the knowledge to do that.

This strategy leads to the institution of a kind of “museum of artistic works” and needs all the related cares (including maintenance, periodic test of devices, periodic retraining of technical personnel, etc.).

On the other hand, the best solution would be to save only the meaning that is inside an artistic work, so that it could be re-performed at any time with the technology of the moment.

This approach introduces to the use of new languages needed to describe artistic works [4] and doesn't solve the problem of the already produced works.

In the same perspective, some researchers have explored mathematical formalization of signal processing [5].

Often, especially for the early works, people involved in the performance were specifically trained and all the informations were transmitted orally.

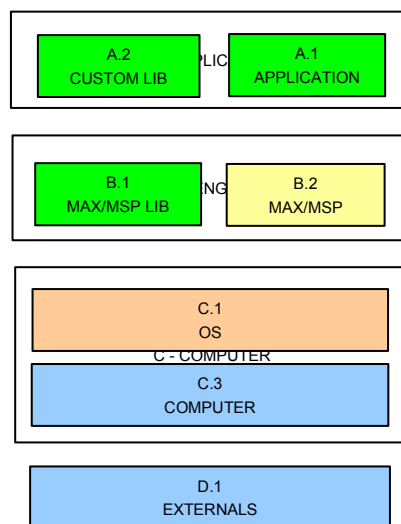
Nowadays some authors, performers and technicians are no longer available to reconstruct the original environment.

This lack of contents makes it impossible to “translate” intentions automatically into a virtual language.

So the strategy is, maybe, the best one for future productions, but only a partial solution for the whole problem.

Part of the effort of our research is in the direction of defining significant indicators and developing tools to extract them from existing works.

Many of the instruments for this purpose are still to be invented.



**Figure 2.** The simplified schema of the digital implementation for an artistic work.

Between the first strategy, conservative, and the second, innovative, some other ways can be followed.

One way may be to virtualize the hardware, using emulators, and leaving the original software untouched.

This is a good result because the re-performance will use the most recent available hardware, but needs to develop an emulator for every used device of the past and does not solve the problem of knowledge about works.

Another way is to port the artistic work on more recent environments every time the last used is near to become obsolete.

This solution only delays the problem but does not solve it, in fact, the author must be part of every new porting process.

Generally he introduces changes into the new work. So, a new artistic work is generated and it has to be preserved as well.

Finally, is to be considered the so called “problem of authenticity”.

It means that every time we modify something into an existing artistic work, we have to prove that the new product is *equivalent* to the original.

This can be achieved using specific authentication processes based on standard methodology, with absolute physical measures and throw the intervention of reference *validators* (authors, performers, musical assistants, etc.) [6].

## 3. TOOLS TO SUSTAINABILITY

### 3.1. Structure tools

We have developed a set of tools, named Patcher Tools, based on a parser of Max/MSP modules that make available the whole structure of a patch and its sub-patches.

As an instance, let us consider the Max/MSP patch of *Jupiter*, composed by Philippe Manoury in 1987.

This work, for flute and live electronics, is regularly performed since its creation.

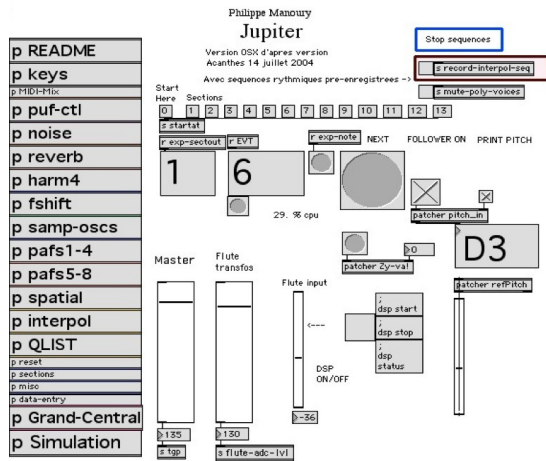


Figure 3. A screenshot of the main patch of *Jupiter*

Looking at figure 4, that represents the top patch, the boxes in the left column are sub-patches.

All together they make a sort of electronic orchestra: *reverb* is the module of reverberation, *fshift* is the frequency shifter (a signal process that adds and subtracts a certain frequency to the fundamental of the input signal), and so on.

The horizontal series of buttons, from 0 to 13, at the top of the patch, enable the user to trigger the beginning of the corresponding sections into the score.

Other boxes are for instance faders to adjust sound levels (input and output) or start/stop buttons.

Within the development program is not possible to have an overview of the whole hierarchy of patches.

That is the first reason for which we developed the PatcherMap tool.

Other important features of this tool is the ability to check the set of resources required inside the patch.

Figure 4 shows various lists: on the left, the list of abstractions (all the required files), the list of externals (third-party objects), the list of missing references (referenced but not found patches), the list of script files used, and the list of data files.

In the centre of the window, the hierarchy is displayed as a tree, with the possibility of showing only a part of it (limiting the depth or the width to be displayed).

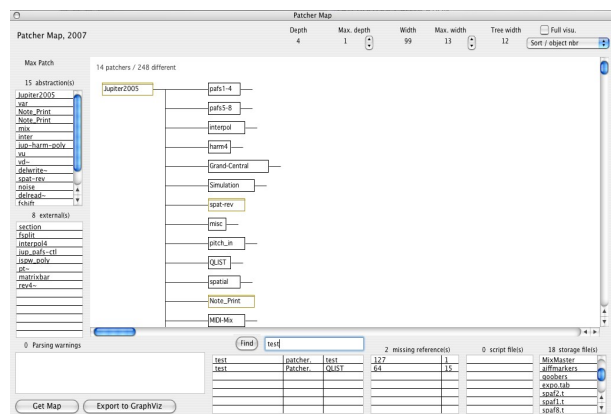


Figure 4. A screenshot of the *PatcherMap* application showing the hierarchy of modules of *Jupiter*.

### 3.2. Repository tools

Some of the artistic contemporary works are stored into existent repositories [7], some others are still resident in private archives, for some we don't know references.

Externals components (the libraries used in several artistic works) need to be documented adding some further features (version, date of last update, etc.), and informations on their behaviour.

For the purpose of bringing all together we have developed a specific repository that is based on a portal architecture and we have provided it with several specialized portlets.

The repository would be a media to store objective informations, as the features stated before, to collect external sources and to grab some subjective informations from users community.

The last ones ideally should be handled by musical assistants, through one or several wiki portlets, that will store informations such as migration informations (heuristics...), comparisons, comments and so on.

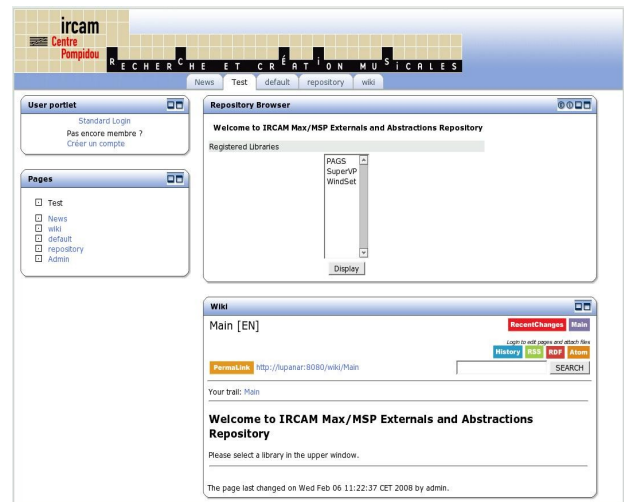


Figure 5. A screenshot of the portal for components

In addition to the information provided, queries on the contents will be possible for the purpose of localization of newer versions of a component, or versions of the same component on a different operating system, as an example.

Hash values will be provided in order to enable precise identification of the required components.

The repository is thought to make it possible to analyse texts stored, collected now as comments, in order to build thesauruses, dictionaries of terms, and relationships between entities.

These elements will enable to build specialized ontology of digital instruments.

### 3.3. Display tools

The hierarchy provided by structure tools can be displayed using any standard tool.

The purpose is to express informations in the most convenient form for users understanding.

As an example, figure 7 shows a different way to present the structure of *Jupiter*.

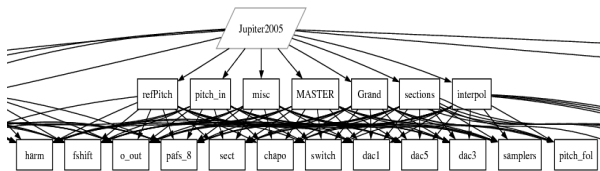


Figure 6. A screenshot of GraphViz, a standard open source tool.

These tools may be used to have an overview of patches during their reimplementaion but also for the evaluation of scenarios complexity, as an example.

### 3.4. Software engineering tools

Among the tools developed we can distinguish two different classes: the ones for static code analysis and the others for run-time performance diagnosis.

The static code analysis class of tools is intended to provide, in structured schemes, further informations about the inside of an artistic work, including environment description, project structure, file structure and catalogs for libraries (see figure 8).

We developed an “intelligent” code coverage analysis technique, we are using it to identify code holes, lacks of consistency, complexity and styles of writing.

All the collected informations will be fit into the repository and will be available as meta-data of the work.

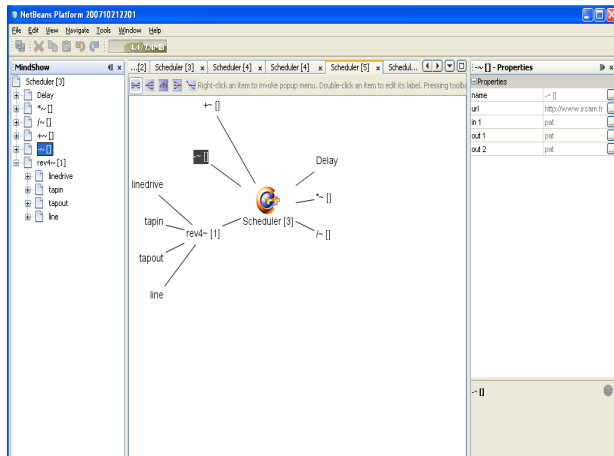


Figure 7. A screenshot of static analysis IDE.

The run-time class of tools will provide the way for both adequate dimensioning of hardware and monitoring all aspects of performances.

Based on professional profilers technology, they monitor global parameters like memory and CPU, but can furnish detailed values on every function running inside the project (number of instances, parameters passing overhead, memory usage, CPU usage, etc. as shown in figure 9).

Tools are specially oriented to musical or graphical processing, so directly provide control over complex structures like buses, pipe-lines and special data formats.

Class Name	Allocated Objects	Bytes Allocated	Objects Allocated
java.lang.String	34,448 (17.1%)	21,495 (14.1%)	190 (0.1%)
java.util.HashMap	4,144 (2.1%)	2,308 (1.5%)	2,302 (0.1%)
java.io.InputStream	2,088 (1.0%)	196 (0.1%)	292 (0.1%)
java.util.HashMap\$Entry	1,948 (1.0%)	196 (0.1%)	302 (0.1%)
java.util.HashMap\$Key	1,948 (1.0%)	196 (0.1%)	302 (0.1%)
java.util.HashMap\$Value	1,948 (1.0%)	196 (0.1%)	302 (0.1%)
java.util.HashMap\$Node	1,948 (1.0%)	196 (0.1%)	302 (0.1%)
java.util.HashMap\$EntrySet	1,948 (1.0%)	196 (0.1%)	302 (0.1%)
java.util.HashMap\$KeySet	1,948 (1.0%)	196 (0.1%)	302 (0.1%)
java.util.HashMap\$ValueCollection	1,948 (1.0%)	196 (0.1%)	302 (0.1%)
java.util.HashMap\$KeyCollection	1,948 (1.0%)	196 (0.1%)	302 (0.1%)
java.util.HashMap\$EntryCollection	1,948 (1.0%)	196 (0.1%)	302 (0.1%)
java.util.HashMap\$EntryCollection\$EntryIterator	1,948 (1.0%)	196 (0.1%)	302 (0.1%)
java.util.HashMap\$EntryCollection\$EntryIterator\$EntryIteratorEntry	1,948 (1.0%)	196 (0.1%)	302 (0.1%)
java.util.HashMap\$EntryCollection\$EntryIterator\$EntryIteratorEntry\$EntryIteratorEntryEntry	1,948 (1.0%)	196 (0.1%)	302 (0.1%)
java.util.HashMap\$EntryCollection\$EntryIterator\$EntryIteratorEntry\$EntryIteratorEntryEntry\$EntryIteratorEntryEntryEntry	1,948 (1.0%)	196 (0.1%)	302 (0.1%)
java.util.HashMap\$EntryCollection\$EntryIterator\$EntryIteratorEntry\$EntryIteratorEntryEntry\$EntryIteratorEntryEntryEntry\$EntryIteratorEntryEntryEntryEntry	1,948 (1.0%)	196 (0.1%)	302 (0.1%)
java.util.HashMap\$EntryCollection\$EntryIterator\$EntryIteratorEntry\$EntryIteratorEntryEntry\$EntryIteratorEntryEntryEntry\$EntryIteratorEntryEntryEntryEntry\$EntryIteratorEntryEntryEntryEntryEntry	1,948 (1.0%)	196 (0.1%)	302 (0.1%)

Figure 8. A screenshot of profiler, a dynamic analysis tool.

## 4. CONCLUSIONS

Thanks to the open architecture at the base of our solution, many other tools can be developed for cooperation in the future.

Sharing the large amount of meta-data available for the archived works and enriching them with the important experience of users, we will be able to make contemporary arts safe from obsolescence and usable, so really still alive.

Only the first part of the job has been done and we have to make a large amount of further research to let informations «friendly» and their retrieval as simple as possible.

Hopefully we will also be able to supply future developments with usefull and efficient guidelines: this is the way we have thought to have always straight routes to our roots.

## 5. ACKNOWLEDGEMENT

Work partially supported by European Community under the Information Society Technologies (IST) programme of the 6th FP for RTD - project CASPAR contract IST-033572.

The authors are solely responsible for the content of this paper.

It does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of data appearing therein.

## 6. REFERENCES

- [1] Longton, M. 2004. Record Keeping Practices of Composers, a survey (revised in 2004). InterPares 2 Website, at <http://www.interpares.org>, accessed October 2006.
- [2] Bernardini, N., and Vidolin, A. 2005. Sustainable Live Electro-acoustic Music. In *Proceedings of the International Sound and Music Computing Conference*, Salerno, Italy, 2005.
- [3] Puckette, M. 2004. *New Public-Domain Realizations of Standard Pieces for Instruments and Live Electronics*. In *Proceedings of the International Computer Music Conference*, Miami, 2004.
- [4] Music Notation Journal, Fall, 1986, 4(2) pp. 47-48.
- [5] Orlarey, Y., Foer, D., Letz, S. 2002. An Algebra for Block Diagram Languages. In *Proceedings of International Computer Music Conference ICMA 2002*, Göteborg, Sweden, 2002.
- [6] Roeder, J. 2006. Preserving Authentic Electroacoustic Music: the InterPARES Project. In *Proceedings of the IAML-IASA Congress 2006*, Oslo, Norway, 2006.
- [7] Bachimont, B., Blanchette, J.-F., Gerszo, A., Swetland, A., Lescurieux, O., Morizet-Mahoudeaux, P., Donin, N., Teasley, J. 2003. Preserving Interactive Digital Music: A Report on the MUSTICA Research Initiative. In *Proceedings of the Third International Conference on WEB Delivering of Music (WEB'03)*, Leeds, England, 2003.