A multimedia environment for interactive music performance

Bresin, R. and Friberg, A.

journal: TMH-QPSR
volume: 38
number: 2-3
year: 1997
pages: 029-032

A multimedia environment for interactive music performance

Roberto Bresin and Anders Friberg

Abstract

We propose a music performance tool based on the Java programming language. This software runs in any Java applet viewer (i.e. a WWW browser) and interacts with the local Midi equipment by mean of a multi-task software module for Midi applications (MidiShare).

Two main ideas are at the base of our project: one is to realise an easy, intuitive, hardware and software independent tool for performance, and the other is to achieve an easier development of the tool itself.

At the moment there are two projects under development: a system based only on a Java applet, called Japer (Java performer), and a hybrid system based on a Java user interface and a Lisp kernel for the development of the performance tools. In this paper, the first of the two projects is presented.

Introduction

After many years of research in computer generated performance, there is the need to collect all the most promising results in a hardware/software environment that allows musicians (both professional and not) to use and test them.

Such an environment would be useful for studying music performance, and to perform music otherwise impossible to be performed in a different way. In the first case, music students could take advantage of such a system: they can use it as a tool to better understand the mechanisms of music performance, and/or to build their own new performance rules. In the second case, if we look at computer-music composers, who want to perform their compositions, they would benefit from our system since it let them concentrate more on composition then on performance, and also a new role for the interactive computer-music performer could arise.

Previous software systems

In the last years, two different software packages were developed separately at the Department of Speech, Music and Hearing, KTH, Stockholm (TMH-KTH), and at Centro di Sonologia Computazionale (CSC), Padova University.

The first one, Director Musices, is a stand-alone program written in Common Lisp for Macintosh computers (a porting to the Allegro Common Lisp environment for Windows 95 is 1995a) as well as the more recent rules the basic under development). Director Musices contains set of rules described in Friberg (1991; Punctuation (Friberg et al., 1997) and Phrase-arch (Friberg, 1995b). It is also used for the development of rules; new rules can easily be added using the rule definition utilities. It features Midi/Midi file input/output and parameters, such as duration and sound level, represented in terms of physical units.

The software developed at CSC is called Melodia: it runs under Windows, and allows the user to perform files of different formats (Midi, Csound, Adagio, Melodia). This software can perform music in two different ways: using a rule system based on that developed at TMH-KTH, with some added feature and differences (Battel & Bresin, 1993), or using a Neural Network (NN) based system that uses already trained NNs (Battel et al., 1994; Bresin et al., 1992, 1993, 1995).

Both programs are continuously under development and modification.

A machine independent environment

The problem of portability of software and of its continuos update lend us to think to re-design our systems, so to separate the research on performance from the user interface design and evolution. The basic idea is to have a Lisp kernel for the performance tools (rules, NNs, and other) communicating with the graphic user interface (GUI). The latter is written in Java language and it takes advantage of the Midi share operating system developed at GRAME (Orlarey & Lequay, 1989; Fober, 1994; Orlarey, 1994).

We chose Java mainly for the following reasons:

- *Same software for every operating system.* Users can work with any kind of operating system (Mac OS, Windows, Unix, Linux, ...) and they will be able to use the same piece of software with a familiar user interface (since a Java program maintain the graphics characteristics of the operating system in which is running.)
- *An applet runs in a WWW browser.* Easy interaction with databases and other WWW based services.
- *Small dimension of the code.* The Java classes implementing an applet are very small in memory dimension if compared with compiled software: in this way the download from Internet is very fast.

Midishare allows a Java applet to exchange real-time Midi messages with any Midi device attached to the client machine.

The system can run both on a computer network or a stand alone machine. The Java program will send/receive parameters to/from the Lisp program: in this way it is possible to have different developments both for the Lisp and the Java code. In particular, we expect users of the system to suggest us both new improvements for the GUI, and further adjustments of the performance rules and NNs. In this way, the whole system is easy to upgrade and develop.

Japer: Java Performer

In this paper, we present JAPER (Java Performer): it is a complete Java version of the "merge" between Director Musices and Melodia. The reason of such a choice is to distribute a complete version of the performance program also to people who do not have a Lisp environment and want to use the software under different operating systems.

Figure 1 shows a general scheme of the global system. In the present work, "Host" stands both for a WWW server or for a local host (it is the place where the applets are hosted). The box "Java applets" is represented by Japer; "MidiShare" is the real time multi-tasks software module for Midi applications. It acts as a Midi interface and communicates with other compatible applications as well with any Midi devices of the local host.

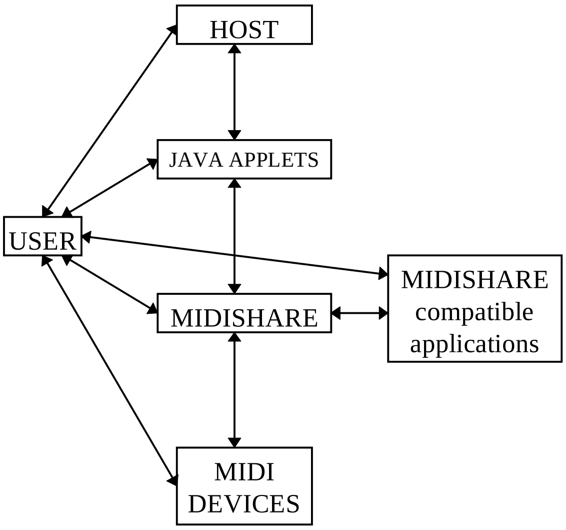

Figure 1. General scheme of the system.

In the following lines, a description of the system running is given.

The user can run Japer in a WWW page or in an applet viewer, and do the following actions:

1. *Load a score.* The input score can be written in different formats (Midi, Csound, Melodia) and can be loaded both from a server on the Internet or from the user's computer. The Melodia format provides a meta score in which information related to performance actions and score description are inserted (i.e. phrase limits, bar boundaries, tonality changes).
2. *Set the performance parameters/weights (cf. the k values in Friberg, 1991).* Each performance command (rule, NN, or other) has one or more sliders to set the values of some parameters in order to achieve the desired performance.
3. *Process the score.* The score is performed according to the settings at point 2.
4. *Play the score (both in its dead pan version and in its precessed version).* The user can listen to the performance using any Midi device connected at her/his machine thanks to the MidiShare system.
5. *Save the performance and/or interact with other MidiShare applications.*
6. *Go back to point 1, 2 or 3.*

Applications

The system has an important application in the pedagogical field, since it can be used as a mean to study the different behaviours of performance rules and performance actions.

JAPER will be part of the "*Garden of Knowledge*", a project under development at CID-NADA, KTH, Stockholm. The Garden of Knowledge is a knowledge tool developed in order to experience, explore and experiment with connections between mathematics, music and art.

Furthermore, a Java applet implementing performance capabilities is useful to perform scores stored in music databases, i.e., in Internet it is possible to find very large music file databases like the Classical Midi Archive, http://www.prs.net/midi.html, which contains over 4290 classical music files in Midi format. A performance tool would be very useful since most of the files in these databases are not performed or are poorly performed. (Otherwise they would not be printable in a useful form using a commercial score editor, and it would also take a lot of time to obtain a good performance adding nuances by hand!).

Conclusion

We proposed a music performance tool based on the Java programming language, running within a Web page and interacting with the local Midi equipment by mean of a multi-task software module for Midi applications (MidiShare).

The system is hardware and software independent and provides an intuitive tool for music performance.

The modularity of the architecture as shown in Figure 1 provided an easier development of the software tool.

Future versions of the system will include a Java score viewer and editor as well as the interaction with new Java modules that will be developed; and a Lisp kernel for the formulation of new performance commands.

Resources available on the Internet

Java applet for rules testing:
http://www.speech.kth.se/~roberto/japer

KTH performance rules description:
http://www.speech.kth.se/music/performance/

Melodia sw (Windows OS):
ftp://ftp.iam.it/pub/music/melowin.zip

Director Musices sw (Mac OS):
http://www.speech.kth.se/music/performance/download

MidiShare:
http://www.grame.fr/english/MidiShare.html

Garden of Knowledge:
http://www.nada.kth.se/cid/projects/garden.html

Acknowledgements

This work was supported by EU TMR program (Training and Mobility of Researchers), by the Swedish National Council for Research in the Humanities and Social Sciences, and by the Bank of Sweden Tercentenary Foundation.

References

Battel GU & Bresin R (1993). Analysis by synthesis in piano performance: a study on the theme of the Brahms' "Variations on a Theme of Paganini", op. 35. In: *Proc of SMAC 93 (Stockholm Music Acoustic Conference),* Sweden, 69-73.

Battel GU, Bresin R, De Poli G & Vidolin A (1994). Neural networks vs rules system: evaluation tests of automatic performance of musical scores. In: *Proc of 1994 ICMC*, ICMA, San Francisco, 109-113.

Bresin R, De Poli G & Vidolin A (1992). Symbolic and sub-symbolic rules system for real time score performance. *Proc of 1992 ICMC,* ICMA, San Francisco, 211-214.

Bresin R, De Poli G & Vidolin A (1993). A neural network based system for automatic performance of musical scores. In: *Proc of SMAC 93 (Stockholm Music Acoustic Conference),* 74-78.

Bresin R, De Poli G & Vidolin A (1995). A neural network based system for automatic performance of musical scores. In: *Proc of Conference Das Hör als Interpreter, Berlin.*

Fober D (1994). Real-time Midi data flow on Ethernet and the software architecture of Midishare. In: *Proc of 1994 ICMC,* ICMA, San Francisco, 447-450.

Friberg A (1991). Generative rules for music performance: a formal description of a rule system. *Computer Music Journal,* 15/2: 56-71.

Friberg A (1995a). A quantitative rule system for musical expression. *Doctoral dissertation,* Royal Institute of Technology, Stockholm, Sweden.

Friberg A (1995b). Matching the rule parameters of Phrase arch to performances of "Träumerei": A preliminary study. In: Friberg A & Sundberg J (eds), *Proc of the KTH symposium on Grammars for music performance,* May 27, 1995, Stockholm, Sweden, 37-44.

Friberg A, Frydén L & Sundberg J (1997). A rule for automatic musical punctuation of melodies. In: *Proc of the third triennial ESCOM conference,* Uppsala, Sweden, 719-723.

Orlarey Y (1994). Hierarchical real time inter-application communications. In: *Proc of 1991 ICMC,* ICMA, San Francisco, 408-415.

Orlarey Y & Lequay H (1989). *MidiShare: a real time multi-task software for Midi applications, Proc of 1989 ICMC,* ICMA, San Francisco, 234-237.