

Recent Developments of MidiShare

Dominique Fober, Stéphane Letz, Yann Orlarey

GRAME Research Laboratory, 6 quai Jean Moulin, BP 1185, 69202 LYON Cedex 01, France

Email: [fober,letz,orlarey]@rd.grame.fr

Abstract

MidiShare is a real-time multi-tasks musical operating system first presented at the ICMC 89. Based on a client/server model, MidiShare offers fast and accurate real-time performances, multi-platform support, powerful inter-application communications, while considerably simplifying the development of complex real-time musical applications. MidiShare has been used by our institution for internal developments for more than 7 years. It was awarded by the Apple Trophy 89 and the Paris-Cit 90 prize. Today, it is evolving toward a distributed multi-platform environment. The poster session will present and demonstrate its recent developments: the MidiShare environment, the developer's kit and documentation, multi-platform development issues, libraries, Ethernet support, compatibility with other systems and third parties applications.

1. Introduction

MidiShare is a real-time multi-tasks musical operating system originally developed for the Macintosh by Y. Orlarey and H. Lequay [Orlarey, Lequay, 1989] and recently ported on PC/Windows by D. Fober [Fober et al. 1995].

MidiShare has been designed in response to commonly met problems in the development of real-time musical applications. Particularly, standard operating systems on micro-computers don't take account of real-time memory management or critical resources sharing. Time management is often expensive and is generally not suitable for the complex real-time operations required by musical applications. Moreover inter-application communications, when they exist apart from side effects communication, don't take account of time and can't be used for real-time communication. MidiShare provides its client applications with the appropriate solutions and constitutes an efficient toolbox for the developers.

2. MidiShare environment

2.1 Data representation.

MidiShare offers high level representation for musical data. Instead of using MIDI byte packets, it provides its client applications with fully structured events, time stamped with a millisecond resolution. These events include both the MIDI and the MIDIFile typologies. They also include various internal representations, from *Notes* (shortcut for a *Key On* and its associated *Key Off*) to tasks (client applications functions that need to be called at precise dates).

Their semantic can be easily extended, they constitute the base of inter-application communications. To allocate, copy and dispose these events, MidiShare includes a real-time memory manager able to operate at interrupt level.

2.2 Time management

MidiShare maintains the current date of the system with a 1ms resolution and supports accurate and transparent SMPTE synchronization. It includes a scheduler in charge of delivering events at their falling dates to the communication manager. The scheduling algorithm [Orlarey, 1990] ensures a very low and constant time overhead per event, even when the scheduler is heavily loaded. Client applications needn't worry about time: they can schedule events in the future (including tasks, messages for itself or other applications), MidiShare ensures by means of its communication manager, that these events will be rendered at the right date, according to their semantic.

2.3 The communication

The MidiShare communication manager routes the events received from the scheduler and from the input drivers to the client applications, the output drivers or the tasks manager, according to the events type and the connection set between applications. This connection set can be dynamically configured by the users. Inter-application communications are transparent for the developer but provide users with a simple and efficient way to freely assemble applications in a collaborative network.

2.4 Real-time tasks management

MidiShare offers several mechanisms to control the real-time behavior of an application. Function calls can be scheduled in the future and are automatically achieved by MidiShare at interrupt level as soon as the specified date is

due. Incoming MidiShare events as well as global context changes (new clients starting a session, modifications of the connections between applications, SMPTE synchronization changes, etc.) can be processed in real-time by user defined *Alarms*

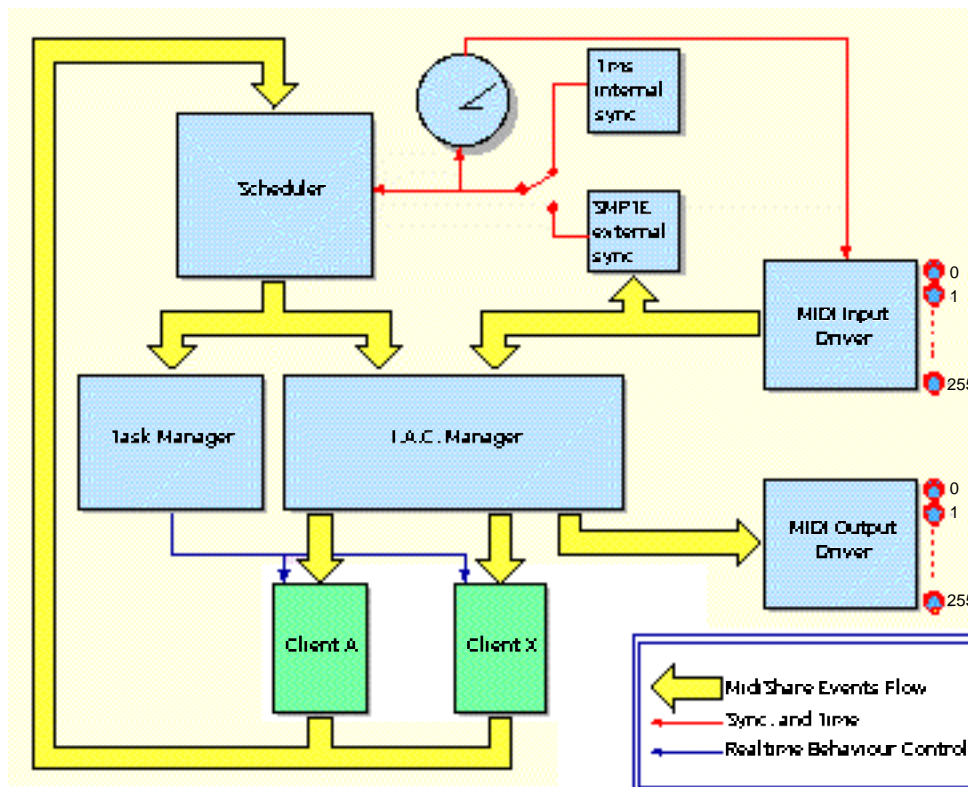


figure 1: MidiShare internal structure overview

3. Recent Developments

3.1 Ethernet support

MidiShare communication scheme has been extended to real-time communication over Ethernet [Fober, 1994] in order to enlarge inter-applications collaboration to inter-stations collaboration. Our implementation respects the traditional services of the network while allowing at least 25 stations to transmit a full MIDI data flow. Users can see the remote stations like any other MidiShare applications and can include them within their communication scheme using the same way and the same tools.

3.2 OpenShare

Since MidiShare first version, several other musical operating systems appeared (and sometimes disappeared). On the Macintosh, the main ones are Apple Midi Manager [Apple Computer Inc., 1990] and Opcode OMS [Opcode, 1993].

Open Share has been developed to allow applications from these different systems to run simultaneously on the same machine and to

collaborate using the MidiShare inter-application communication manager. Via Open Share, Midi Manager and OMS applications can also benefit of the Ethernet support.

3.3 Internet

Several new developments are related to Internet:

- ¥ msNetPlayer, a MidiShare based MIDIfile player Plug-in for Netscape 2.0.
- ¥ a set of MidiShare-JAVA classes for Macintosh and PC/Windows platforms.
- ¥ an extension of MidiShare inter-application communications over TCP/IP.

3.4 Third parties applications

Various MidiShare compatible applications are provided by third parties developers. Among them :

- ¥ *Inter-Actors*, by TimeTech, an innovative software for controlling MIDI devices.

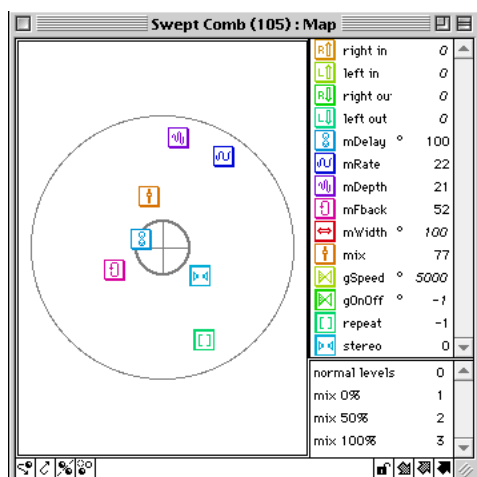


figure 2 : Inter-Actors

- ¥ *Symbolic Composer*, by Peter Stone (<http://www.xs4all.nl/~psto/>), a Common Lisp object-oriented music composition language.
- ¥ *Phonogramme*, by Vincent Lesbros (<http://www.univ-paris8.fr/~gaiv/about-phono.html>) a program to create, transform or compose sounds based on time/frequency representations.
- ¥ *Dmix*, by Daniel Oppenheim (<ftp://ftp-ccrma.stanford.edu/pub/dmix/>), a state of the art environment for music composition and performance based on ParcPlace Smalltalk VisualWorks 2.5.

4. The developer's kit

The MidiShare developer's kit is freely available on the Internet via our ftp server : <ftp://ftp.grame.fr/pub/MidiShare>, or via the MidiShare web page : <http://www.grame.fr/english/MidiShare.html>. It includes a full programmer documentation, a C tutorial and full source code of several applications. Interested developers can also subscribe to the MidiShare mailing list to be periodically informed.

4.1 Multi-platform support

MidiShare currently runs on Macintosh, Windows (from 3.1 to Windows 95) and Atari. The API is the same on the three platforms and source code can be easily ported between them.

Interface files exist for several languages:

- ¥ C, Common Lisp, Pascal, SmallTalk for the Macintosh;
- ¥ C, Delphi, Common Lisp for Windows;
- ¥ Lattice C, Turbo C and Mark Williams C for the Atari.

Several of them come from users' contributions. Inline help is available for the MPW environment on the Macintosh.

4.2 Additional libraries

Two additional libraries are also provided: a MIDI files management library and a Player library.

The MIDI files library is intended to read and write MIDI format files and to convert them into MidiShare events sequences.

The Player library is intended to facilitate the development of a complete multi-tracks MidiShare sequencer: each Player is a MidiShare application, can manage up to 256 tracks, supports three different synchronization modes, transparent events chase and loop markers.

The MIDI files library is compatible with all the supported platforms. The Player library is provided for the Macintosh and for Windows.

5. Acknowledgments

The authors would like to thank all the persons who contribute to MidiShare by providing interface files and applications. Olivier Carbonnel, Mark Bilton, Vassili Lygeros, Michel Rousset, Gil Muller, Nicolas Roux have directly contributed to the kernel development. Colin Whyles and Lorien Dunn offered to support the Atari version on the Falcon. Thanks also to J.P. Smets, Georges Bloch, Laurent Veliscek, Olivier Delerue, Thierry Carron whose help has been very useful.

References

- [Apple Computer Inc., 1990] *MIDI Management tools 2.0*. Apple Programmers Developers Association.
- [Fober, 1994] Fober D. *Real time, musical data flow on Ethernet and MidiShare software architecture*. Proceedings of the ICMC, 1994, ICMA, San Francisco.
- [Fober, Letz, Orlarey 1995] *MidiShare, un syst me d'exploitation musical pour la communication et la collaboration* - Proceedings of the Journ es d'Informatique Musicale JIM95, Paris
- [Grame, 1990] *MidiShare Developer Documentation*. Grame 1990, Lyon
- [Opcode, 1993] Doug Wyatt, *Opcode MIDI System, Application and Driver Interface*, Opcode Systems, Inc., 1993
- [Orlarey, 1990] Orlarey Y. *An Efficient Scheduling Algorithm for Real-Time Musical Systems*. Proceedings of the ICMC, 1990, ICMA, San Francisco.
- [Orlarey, Lequay, 1989] Orlarey Y., Lequay H. *MidiShare : a Real Time multi-tasks software module for Midi applications*. Proceedings of the ICMC, 1989, ICMA, San Francisco.