The new MARS workstation

Andrenacci Paolo, Armani Fabio, Bessegato Renato, Paladin Andrea,
Pisani Patrizio, Prestigiacomo Angelo, Rosati Claudio, Sapir Sylviane, Vetuschi Mauro
IRIS srl, Parco La Selva 151, I-03018 Italy
mc2842@mclink.it, http://aimi.dist.unige.it/IRIS/index.html

Abstract

The Musical Audio Research Station (*MARS*) is a programmable specialized digital machine for real time audio applications which has been entirely developed by the Italian Bontempi-Farfisa research institute IRIS. MARS has been conceived as an integrated environment in which a graphical user interface, an embedded real-time operating system and two IRIS digital audio processors are linked together to create a flexible and an interactive workstation for audio research, musical production and computer music pedagogy.

MARS has been used in a wide range of research, musical and teaching contexts such as universities, music conservatories, computer music research centers, music studios, interactive museums, and in the musical instruments and audio industry.

The first version of MARS, which has been presented at the ICMC92 was running on Atari computers. In this paper IRIS will present the new version of the MARS workstation which has been entirely redesigned for the PC world.

1 MARS overview

MARS derives from the *MUSICn* philosophy, as it maintains the basic approach developed by M. Mathews [5]. Synthesis algorithms are specified as combinations of building blocks; the activation of instruments is controlled by note events. Its historical background can also be found in the main features of the *4X system* [4] (high performance real-time sound processing and interactivity) and in *GROOVE* general concepts for controlling note parameters through functions of time during performance [6].

It emphasizes symbolic representations of data in order to offer a user interface which is independent of hardware architecture and which makes musical applications portable.

1.1 Architecture

MARS is a traditional three level based (DSP, controller and host) system [1]. The IRIS *X20* Digital Signal Processor is dedicated to real-time sound processing. The Motorola MC68302 microcontroller supports real-time operating system *RT20M*, developed for sound generation board *NERGAL*. The host computer supports graphical development environment *ARES*. This architecture guarantees high performance and gives the opportunity to distinguish between sound processing and performance controls, also allowing musicians to use MARS as a MIDI instrument.

1.2 Objects

MARS has a predefined set of primitive objects which can be edited using *ARES* graphical environment to build more complex objects such as algorithms, tones, orchestras and MIDI performance setups. Following are definitions of the main objects managed by the system.

The MARS basic audio objects are user patchable *algorithms* which correspond to specific signal processing operations (microcode for the X20s). Algorithms are built graphically and are immediately implemented on the sound generation board avoiding compilation time.

For each algorithm, the user can configure a number of *tones*. A tone defines for all algorithm parameters the rules for calculating their values and their relationship with MIDI controls and events.

An *orchestra* is a collection of polyphonic algorithms (*clones*) and a map of their connection with audio busses, DACs and ADCs. For each orchestra, the user can create a number of *tone maps* which define the collection of tones to be played and the overall orchestral MIDI environment, also called *performance environment*.

The creation and editing phases of these objects involve different contexts which are managed by different processors. For example, the ouput data of an algorithm or an orchestra editing session involves the DSP level (X20 microprograms, data memory values, sample memory, ...), while the ouput data of a tone or a tone map editing session involve the control level (i.e. the operating system running on the microcontroller) which manages the MIDI messages and the performance environment interactions (see *Figure 1*).

2. The new hardware

The minimum MARS hardware requires a sound generation board NERGAL with 8 megabytes of sample RAM and a low-cost audio unit.

Hardware expansion allows to plug up to 8 sound generation boards together into the ISA bus extensions of the host PC. On each board, sample RAM may be expanded up to 64 megabytes. Furthermore, IRIS provides a configurable professional audio unit which allows up to 8 analog inputs and outputs, and 8 AES-EBU inputs and 8 outputs or another digital audio interface.

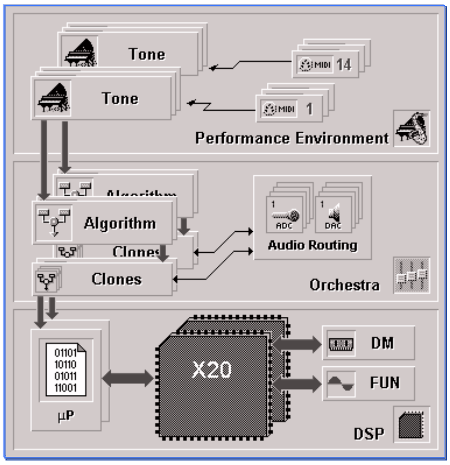


Figure 1: MARS objects hierarchy

2.1 NERGAL, the new MARS sound board

NERGAL derives from the SM1000 sound generation board [1]. It is a full size standard IBM ISA card running at a sampling rate of 44.1 kHz. NERGAL is divided into three main sections: DSP, control and mixer sections.

The DSP section contains two IRIS X20 DSPs and sample memories.

The control section include a MOTOROLA MC68302 microcontroller with the IRIS RT20M operating system. RT20M manages all the MARS objects described previously, and all their interactions with the DSP section, and any communication through MIDI, RS232, and ISA ports.

The mixer section controls a set of predefined audio mixing operations among boards and audio units.

Each board has one connector to/from the audio-unit, and four connectors (two inputs and two outputs) which allow multi-board links. This last feature allows to create a MARS boards network with a parallel, a serial or a mixed configuration using one or more audio units.

2.2 MARS Audio units

Two types of IRIS audio units are available for the NERGAL board: low-cost and professional.

The low-cost unit is a small external box (1/3U) with two monophonic input lines, four monophonic output lines, and one stereophonic headphone output.

The professional audio unit is a 2U standard rack module. It is modular and may pilot up to 8 input/output analog channels and 8 input/output digital channels. Each unit can be shared between two different NERGAL boards.

3. ARES

The MARS music workstation is an open system that needs to be programmed and configured. Such operations are all managed by the proprietary development system, the IRIS *Audio Resource Editing System (ARES)* running on Windows 3.x and Windows 95. ARES hides all the low-level feature of the system, so users don’t have to know anything about the X20 architecture, or about the RT20M details. Furthermore, ARES provides a set of default behaviours which help users to quickly program the system. Finally, ARES provides any editing in a real-time interactive context.

3.1 Use and programming

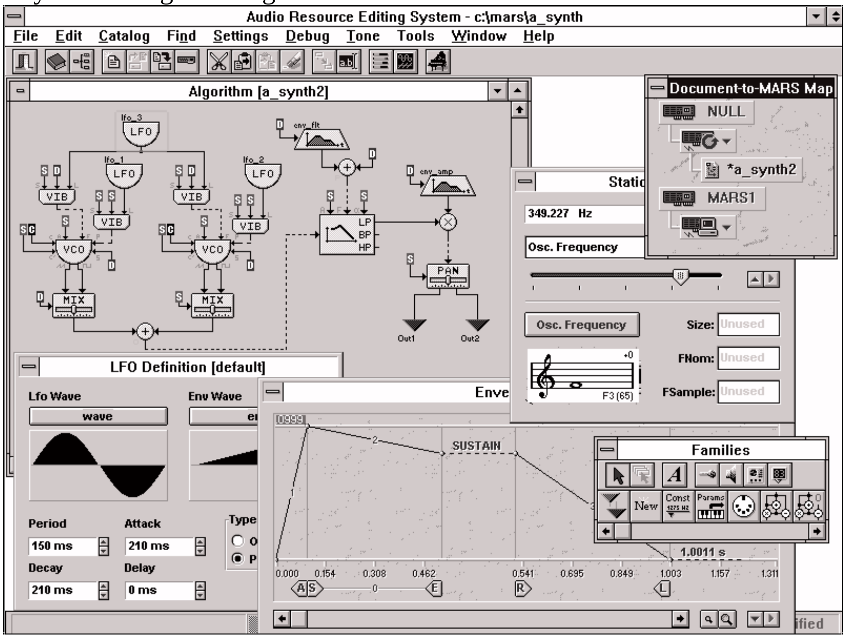
The work on MARS usually develops in three steps. The steps order described in the following paragraphs is not a strict necessity, on the contrary many editing steps can be done in parallel thus increasing the system flexibility.

The first step is to create and debug algorithms with the algorithm editor [2] for the processing of signals: synthesis, analysis and effects. This system provides the user with a rich and expandible library of patchable modules for signal processing and with a series of debugging tools (oscilloscope, VU-meters, etc.) which simplifies the creation and debugging of algorithms. Furthermore, the system implements the modules in real-time, as soon as they are installed by the user. This operation requires no pre-processing or compilation of the algorithms, thus allowing one to instantly hear the resulting sound.

The second step is to create and debug timbres [7] with the tone editor: define their values and the rules for calculating parameters of previously defined algorithms. Any algorithm can be configured with a number of tones (defining its parameters values, its envelopes and its waveforms, and assigning and defining MIDI control parameters, etc.) by means of a series of secondary editors and graphically interactive tools that are all part of the ARES package.

The third step is to create and debug the orchestra with the orchestra and the tone map editors: listing the algorithms and timbres which will be used and defining the MIDI performance environment associated with the orchestra. While developing new algorithms and timbres, the user creates a sort of database of sound resources that can be later used inside one or more orchestras for the execution of musical pieces. These orchestras can be considered true perfomance environments once the user specifies the polyphony of various algorithms used, the list of timbres to be played, the assignment of MIDI channels, and finally the configuration of the audio bus.

Once these three phases are completed, MARS can be used for live music by connecting it to a gesture MIDI device or some MAX application [9], by playing a MIDI score executed by a sequencer, or by combining these techniques.



349.227 Hz

Period 150 ms

Attack 210 ms

Decay 210 ms

Delay 0 ms

Figure 2 : ARES windows examples

3.2 Friendly high-level user interface

ARES hides from the user all low-level implementation and communication aspects related to MARS sound generation board. Multi document metaphor for all MARS objects editors, and real multi-window management makes editing an easy task (see some windows examples on *Figure 2*).

Design, enhanced graphics, intensive use of short-cut keys, tool bar buttons, and information displayed in the active document status bar help users to interact with ARES. In general you can act on any kind of object by following these two steps: selecting the object by clicking on it, and carrying out an edit operation by using, the mouse, a short-cut or a menu command.

ARES On-line Help allows user to reach any kind of information on patchable modules, glossary terms, editors layouts and commands, procedures and so on. Furthermore a complete tutorial and a documented set of demo are available to enable beginners to easily start to work with the MARS workstation.

3.3 A browser and other tools

While developing MARS objects, the user creates a sort of database, the *catalog* of sound resources.

In ARES, objects can be accessed by using the *Catalog Browser*. The Catalog Browser hides all ARES subsidiary files focusing only on ARES object names.

ARES also provides a set of software tools which improves MARS overall performance.

The *Oscilloscope* allows users to display audio signals on four independent tracks, picking it up at any point of patch data flow.

The *MIDI Keyboard* allows users to send standard channel MIDI messages through one of the available communication ports.

The *Wave Table Manager* allows users to manage MARS sample memory. ARES supports all major file formats including binary Intel, and binary Motorola.

The *Juke Box* allows users to send to a board one or more performance environments in a batch session without running ARES. Furthermore, the juke box allows playing MIDI files which makes it very useful for stage use of MARS.

3.3 A rich set of proprietary modules

MARS is a development system for interactive design, implementation and test of digital audio algorithms particularly oriented to musical applications. Any kind of real-time digital signal processing techniques such as analysis, synthesis, any kind of filters, and sound effects may be realized.

ARES provides about 200 modules to be used in the algorithm editor environment. This library covers low and high level needs for DSP applications such as arithmetic and logical operations, many types of oscillators, complex sound generators (random, 1/f noise, noise exciter, pulse, additive, FM, AM, ring modulation,..), physical modeling building blocks (strings, reeds, tubes, masses, spring,..), and audio effects (delay lines, harmonizer, pitch shifter, doppler effect, echo, flanger, chorus, reverb, exciter, mixer, panpot effect, envelope follower,..).

ARES modules are logically grouped in families. They may be easily picked-up from a Families Palette and patched in the ARES algorithm editor. In the future, while this modules list will constantly be updated and powered by IRIS, users will have the means of easily organizing their own lists, building their own modules set and editing their families palette.

On the left, *Figure 3* shows an algorithm which uses physical modeling modules to simulate a clarinet. The clarinet model uses the classic excitation/resonator model: a static reed followed by a pipe. The pipe is simple, it is composed by a piece of tube, a hole and a bell. The output is given by mixing the signals produced by the four elements.

On the right, *Figure 3* shows an orchestra with 4 algorithm families. The communication between algorithms and between audio I/O is achieved by the orchestra busses. These busses allow the accumulation of audio signals, but also feedback effects. For example, a feedback effect is obtained by sending the ouput of the string algorithm family to the input of a distortion effect. Then the output of the distortion is sent back to the string in order to be added to the plucked excitation of the string model.

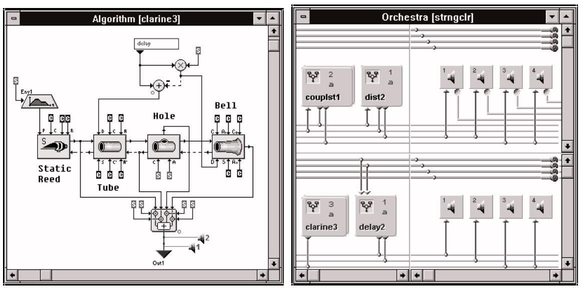


Figure 3 : Algorithm and orchestra examples

4. Extending MARS

MARS remains an open system. It is possible, as for the Atari version, to expand MARS possibilities, or to specialize it for particular needs, by programming dedicated new applications [3, 8].

A new development toolkit consisting of C/C++ functions and class libraries allows programmers to develop their own MARS applications without using ARES. However, these applications will be able to exchange catalogs of sound resources with ARES and to share a common NERGAL network.

5. Conclusions

The new release of the MARS workstation presents a completely new design of the sound board NERGAL and of development environment ARES.

The MARS hardware is now directly pluggable into the ISA bus extension of a PC, and allows multi-board configuration to improve the computing power of the workstation. Furthermore, a new family of audio units allows a more flexible configuration of the audio I/O.

ARES has an improved user interface, a new design of all editors and tools, and it allows real multi-task operations. It contains a browser which makes MARS objects database navigation easier. A richer proprietary modules library and the possibility of expanding this library increase the DSP functionalities of the system.

ARES replaces the old EDIT20 package running on the Atari. It is an improved environment with multi-board and multi-task facilities, but the original “philosophy”, and most of the EDIT20 features have remained unchanged to maintain the compatibility and the continuity with old applications.

A development toolkit offers programmers the chance to create their own MARS applications and to extend the functionalities of the workstation.

MARS is a flexible musical workstation which is used in a wide range of research, musical and teaching contexts such as universities, music conservatories, computer music research centers, music studios, interactive museums, and in the musical instruments and audio industry.

References

[1] P. Andrenacci, E. Favreau, N. Larosa, A. Prestigiacomo, C. Rosati, S. Sapir, “MARS: RT20M/EDIT20. Development tools and graphical user interface for a sound generation board”, ICMC proc., pp. 340-343, San Jose 1992.

[2] F. Armani, L. Bizzarri, E. Favreau, A. Paladin, “MARS: DSP environment and applications”, ICMC proc., pp. 344-347, San Jose 1992.

[3] F. Armani, A. Paladin, C. Rosati, “MARS Applications Using APPLI20 Development Tools: a Case of Study”, ICMC proc., pp. 230-236, Aarhus 1994.

[4] E. Favreau, , “Software development for the 4X real-time system”, ICMC proc., pp. 369-373, S. Francisco 1986.

[5] M. V. Mathews, J. E. Miller, F. R. Moore, J. R. Pierce, J. C. Risset, *The technology of computer music*, MIT Press, Cambridge Mass., 1969.

[6] M. V. Mathews, F. R. Moore, “GROOVE: A program to compose, store and edit functions of time”, Comunications of the ACM 13(2), 1970.

[7] G. Palmieri, S. Sapir., “MARS: Musical applications”, ICMC proc., pp. 352-353, San Jose 1992.

[8] P. Pierucci, “Singing Voice Analysis and Synthesis System through Glottal Excited Formant Resonators”, ICMC proc., Thessaloniki, 1997.

[9] M. Puckette, “Combining Event and Signal Processing in the MAX Graphical Programming Environment”, Computer Music Journal, 15(3), pp. 68-77, MIT Press, 1991.