MARS: RT20M/EDIT20
Development tools and graphical user interface for a sound generation board

P. Andrenacci, E. Favreau, N. Larosa, A. Prestigiacomo, C. Rosati, S. Sapir

IRIS S.r.l.
Parco La Selva 151 - 03018 Paliano (FR) Italy
Fax. (775) 533343 - Tel. (775) 533441 - E-mail: nicola.larosa@eurokom.ie

ABSTRACT

The Musical Audio Research Station is a programmable and open system for real time digital signal processing which allows sound and musical performance environment development. The MARS interactive user interface, EDIT20, provides a means of graphic definition for audio objects, with immediate sound feedback. Its real time operating system, RT20M, is integrated into the SM1000 sound generation board and manages all the interactions with the EDIT20 and external MIDI devices. The high integration of digital audio signal processing with MIDI capabilities provides a flexible musical instrument that can be used for research in the DSP field, for computer music teaching and for live musical performance.

1. OVERVIEW

MARS is an integrated environment in which a graphical user interface, a real time operating system and two general purpose digital signal processors are linked together to create a general sound processing system, which is interactive and operates in real time.

It derives from the MUSICn philosophy, as it maintains the basic approach developed by M. Mathews (Mathews 1969). 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 (Favreau 1986) (high performance real time sound processing and interactivity) and in GROOVE general concepts for controlling note parameters through functions of time during performance (Mathews 1970).

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.

MARS is a traditional three level based (DSP, controller and host) system. The IRIS X20 Digital Signal Processor (Cavaliere 1992) is dedicated to real time sound processing. The Motorola MC68302 microcontroller supports the real time operating system RT20M, developed for the SM1000 sound generation board (ITIS 1991). The host computer is an Atari, which supports the graphical development environment EDIT20. This architecture guarantees high performance and allows the opportunity to distinguish between sound processing and performance controls, also allowing musicians to use MARS as a simple MIDI instrument.

2. OBJECTS

MARS has a predefined set of primitive objects which can be edited using the EDIT20 graphical environment to build more complex objects such as algorithms, tones, orchestras and MIDI performance setups (see Figure 1.a).

**Algorithms** are combinations of predefined unit generators whose entry points represent the interface with tones. An algorithm can support multiple tone definitions.

**Tones** are specific definitions of three classes of variable data: parameters, envelopes and LFOs. The main feature of MARS is the ability to define dynamic conversion rules for parameters and assign MIDI controllers to any algorithm variable. This allows dynamic timbral control during performance.

An **orchestra** is a collection of algorithm definitions. Each algorithm can be cloned to form an algorithm family, characterized by its polyphony level and by the routing of its I/O to the system audio buses.

The **Tone Map** defines the MIDI performance environment of the orchestra. For each algorithm family it is possible to recall predefined tones or to build new ones that will be mapped onto the standard MIDI channels. The global polyphony of the system depends not only on the number of algorithm families and clones, but also on the fact that more tones and more notes can share, at the same time, the resources of one algorithm family.

Once MARS has been programmed and the MIDI environment configured by a complete orchestra and its relative tone
map, the SM1000 board can be disconnected from the host and used as a common MIDI device.

Figure 1. (a) MARS objects and hierarchical structure  (b) Graphical user interface structure

3. GRAPHICAL ENVIRONMENT

The graphical environment EDIT20 gives the user the means to visualize and interact with MARS objects, using graphic windows, dialog boxes and menu tools. EDIT20 represents a high-level user interface (Brown 1989, Hartson 1989) for the performance environment running on the SM1000 sound generation board. It hides from the user the low-level implementation aspects related to the SM1000 processors (X20 DSPs and MC68302 controller).

The program, running on Atari computers, starts with a main shell from which the user calls one of four MARS object-specific editors (ORCH, TMAP, TONE or ALGO) (see Figure 1.b and cfr. 3.1).

The last phase in using the EDIT20 is to build a set of tones organized in a specific MIDI performance environment. There are three ways to edit a tone: via ORCH in a multi-timbral and polyphonic environment, via ALGO in a mono-timbral and monophonic environment, via TONE in a mono-timbral and polyphonic environment.

During the editing of a tone one can invoke several data-specific editors (PAR, LFO, ENV or TAB) (see Figure 1.b), handling from scalar to structured data, such as dynamic time functions, Envelopes, Low Frequency Oscillators and tables (Palmieri 1992).

3.1 Organization

The editors named above are based on a core of four libraries. Each library manages a different aspect of edited objects: graphical representations, data structures, DSP resources and communication with SM1000.

ALGO is the graphical algorithm editor, which allows one to call and connect (Rosati 1992) unit generators to build sound processing algorithms (Armmani 1992). The user interactively develops and debugs one algorithm at a time, setting its variable definitions to build tone prototypes in a one-voice context. The ALGO editor can call specific variable editors such as ENV, LFO, PAR and TAB.

ORCH is the graphical orchestra editor which allows one to call and connect different algorithms in order to define algorithm families, by duplicating them as clones, and routing them to the SM1000 sound generation board audio buses (8 outputs, 4 inputs) (Armmani 1992, Cavaliere 1992).

TMAP editor can be called from ORCH and allows the user to define the MIDI configuration of the orchestra by accessing, for each algorithm family, a set of predefined tones and mapping them onto the MIDI channels (Palmieri 1992).

TONE editor can be called either from the ALGO editor or from the TMAP editor. It displays in one window all the algorithm entry points grouped by type and allows the user to interactively define their values and performance meanings by calling the specific variable editors. When a tone is defined from ORCH and TMAP, the system provides a multi-algorithm, multi-timbral, polyphonic environment which allows the user to better set parameters in a musical context (Palmieri 1992).
EDIT20 integrates a series of tools developed by IRIS such as desktop accessories (virtual MIDI keyboard, DSP debug functions) and applications for creating optimized algorithms (ASM20 and MERGE20) (Armanii 1992, Favreau 1990). It also allows the user to call from the main shell, without leaving the EDIT20, a set of commercial software packages such as a general purpose text editor, a sample editor, a MIDI sequencer and other applications.

4. PERFORMANCE ENVIRONMENT

The RT20M runs on the Motorola microcontroller MC68302. It is an event based real time system for general control of the SM1000 sound generation board. As with most real time musical systems, it has to face the problem of ordering, in time, various parallel tasks to compute sound events and to control performances in real time.

4.1 Basic concepts

The RT20M has to deal with three fundamental components: events, resources and actions (see Figure 2.a).

![Figure 2. (a) RT20M Components (b) Communication](image)

Events are divided into three classes: physical events are produced by external device interrupts; kernel (or logical) events are produced when the status of an internal kernel resource changes; performance events are the result of all RT20M processes.

Resources are divided into two classes: queues are protected shared data areas for synchronized communication between tasks; semaphores are tools for task synchronization.

Actions can be divided into three code modules: drivers are hardware dependant routines that manage physical events (MIDI line, X20 interrupts, timer ticks, VME line); kernel manages the initialization of the board, maintains the synchronization and the communication between processes, allocates and dispatches the resources of the machine and supplies a subset of debugging tools; processes cause performance events according to the current orchestra configuration and to the occurrence of other physical and kernel events.

4.2 Processes

Processes are the only code modules which can be interrupted during execution by physical or kernel events. To manage processes, the RT20M takes into account three main process states: running, ready to run and waiting for resources.

The system provides one Communication process, which allows the setup and dynamic updating of the performance environment by means of MIDI System Exclusive messages.

The following four main predefined processes are available for the computation and production of performance events.

The Note process manages the orchestra polyphony according to the current Note On/Off message (selection of the current tone associated with the channel and selection of the designated algorithm clone which supports the tone definition). It computes and sends note parameters to the relative algorithm inputs and starts envelopes and LFOs. This process is also dedicated to the decoding of all MIDI channel messages.

The Envelope process manages, for each envelope involved in the tone definition and for all active voices, a segment based
data structure which can be modified by MIDI controls during performance.
The LFO process manages, for each LFO involved in the tone definitions and for all active voices, a MIDI controlled data
structure to produce LFO samples that will be sent to the LFO inputs of the algorithm.
The Control process updates note parameters which depend on MIDI continuous controllers such as pitch bend, aftertouch
and control change messages that last for the duration of the active voices.
Developers can easily modify RT20M code by adding other processes and drivers.

5. COMMUNICATION

Graphical and performance environments communicate via the MIDI serial line and/or the Atari VME bus connected to the
high speed SM1000 16-bit parallel port. Communication via the MIDI serial line is MIDI Protocol compatible. Such
compatibility makes the performance environment open to MIDI devices (keyboards, instruments, sequencers, samplers,...),
thus enriching the SM1000's musical facilities (see Figure 2.b). Furthermore, a set of System Exclusive Messages was defined to fully program and dynamically access both the hardware
and software resources of the performance environment.

MIDI Communication may be too slow and hence unacceptable when involving long samples or complex orchestras. The
user may want to write "C" applications that use high-speed management of objects in the performance environment (for
example, to analyze data coming from a SM1000 ADC, display it and then send the information back to SM1000 for
synthesis). For this reason, the set of Exclusive Messages was ported over to the high-speed connection of the Atari VME
Bus with the SM1000 Parallel Port.

A "C" Transmission Library implements the set of messages. It makes an application independent of the System Exclusive
Protocol and of any low level I/O aspects related to the Atari and the selected transmission line. The graphical package
EDIT20 is an example of a complex application using the Transmission Library. A set of tools using this library has also
been developed at IRIS (oscilloscope, real time FFT analysis and synthesis and real-time LPC synthesis) (Armani 1992).

6. CONCLUSION

MARS is an open system which allows signal processing and musical applications. Programmability, interactivity and
symbolic data representation have been the guiding principles in MARS development. The three level architecture of
MARS hardware strongly influenced our software specifications and increased the difficulty of programming and debugging.
However, the graphical interface seamlessly unites and simplifies the vast complexity underlying MARS. MIDI
performance improvements and the porting of the MARS graphical user interface to other platforms such as Apple
Macintosh, NeXT or IBM PC will be part of IRIS program for the future.

ACKNOWLEDGEMENTS

Special thanks to Jean Pierre Armand who contributed to the MARS design and implementation.

REFERENCES

- S. Cavaliere, G. Di Giugno, E. Guarino, "MARS: X20 architecture and SM1000 sound generation board description",
- E. Favreau and al., "Software developments for the 4X Real Time System", Proc. of the ICMC, pp. 369-373, ICMA, S.
  Francisco, 1986.
- ITIS, "SM1000 Documentation, IRIS internal communication, 1991.
- M. V. Mathews, F. R. Moore, "GROOVE: A program to Compose, Store and Edit Functions of Time", Comunications of
  the ACM 13(12), 1970.