



**CONSIGLIO NAZIONALE  
DELLE RICERCHE**

Goffredo Haus <sup>^</sup>, Alberto Stiglitz <sup>^</sup>

**The Functional Performer System**

N. 7/121

5/93

**PROGETTO FINALIZZATO  
SISTEMI INFORMATICI E CALCOLO PARALLELO**

**SOTTOPROGETTO 7**

**SISTEMI DI SUPPORTO AL LAVORO INTELLETTUALE**

**PROGETTO FINALIZZATO**  
**SISTEMI INFORMATICI E CALCOLO PARALLELO**

**SOTTOPROGETTO 7**  
Sistemi di Supporto al Lavoro Intellettuale  
Coordinatore Cesare Maioli

Goffredo Haus <sup>^</sup>, Alberto Stiglitz <sup>^</sup>

The Functional Performer System

N. 7/121

5/93

Reprint

<sup>^</sup> LIM - Laboratorio di Informatica Musicale  
Dipartimento di Scienze dell'Informazione  
Università degli Studi di Milano.



## The Functional Performer System

Goffredo Haus and Alberto Stiglitz

### ABSTRACT

Music performance traditionally consists in sequences of events (notes and chords, for example) as the basic objects. In this paper we show how music can be performed at a more abstract level in which the basic music object is a pattern of musical text (i.e. a chunk of a score). We have designed, implemented and experimented FP (Functional Performer), a software tool that allows the real-time processing of music patterns by means of music functions (e.g. tonal and modal transposition, mirror inversion, juxtaposition, superimposition, time shrinking, and so on); functions are activated by the ASCII keyboard keys and are either deterministic or non-deterministic. Patterns obtained by functions may be arguments for other functions during the current performance. Music is generated as MIDI data and real-time executed by MIDI devices controlled by FP.

### INTRODUCTION

In the music domain, the instrument revolution determined by the development of digital devices has cast doubt on the classic roles of those who produce and enjoy music: the composer, the performer and the listener. In fact, the new computer music workstation features allow the musician to assume all three roles either sequentially or simultaneously, according to the features of music systems (interactive systems or real-time systems).

Therefore, recent trends in computer music research are devoted to a not yet well explored field, i.e. the development of new computer music instruments. Actually, digital systems may be programmed and devices may be connected to each other to build complex systems; this kind of experience leads to a new idea of musical instrument and related music-making techniques.

Particularly, we have had two main aims in designing the Functional Performer (FP): an original and advanced use of commercially available low-cost devices and the realization of a computer music instrument for real-time performance.

In the following sections we show the most relevant features of the FP with respect to similar systems; then we briefly describe how it works by means of some examples.

## THE CONCEPTUAL ENVIRONMENT

This research has been carried out in the frame of the more general "Intelligent Music Workstation" Project (Camurri et al., 1989; Stiglitz, 1991); within the entire project we have accepted as a standard some concepts which seem to have a basic role in both the creation and perception of music; therefore, in the following we will briefly discuss: music levels of representation and related languages, music objects and their transformations by operators, the evolution of music performance pragmatics.

### Music levels of representation

A relevant feature introduced by computer applications to music is the possibility of describing music information at many different levels of abstraction (Haus, 1984; Pope, 1986). Those levels may be accessed concurrently depending on user will. So, we can choose the level which is closer to the activity the user wishes to do, without restrictions about moving up and down within hierarchical music descriptions during user-system interaction. We may think in terms of having at our disposal a kind of "elevator" to access the various levels of abstractions.

From an abstract point of view, we can think that music can be described by a number of levels, as in the paper by De Poli and Haus (1982):

- structural level:* where relationships among music events are defined;
- symbolic level:* where elementary music events (i.e. notes) juxtapositions and superimpositions are defined (as we can do this on the traditional score and also by composing and sequencing software packages);
- timbre level:* where the timbre models of music events are defined;
- operative level:* where operative codes define the timbre models of music events;
- audio signal level:* where music is simply described as sequences of instantaneous values of amplitude (i.e. samples).

Every common music activity generally involves more than one level of representation: for example, music performance requires actions mostly regarding notes, but also timbre models and timing control. Therefore, three levels are involved: symbolic, timbre and operative.

### Music objects

In our research we have adopted the notion of music object (Degli Antoni & Haus, 1983) which is more general than music event and also less related to a particular level of abstraction. Music object means anything that could have a musical meaning and that we think of as an entity, either simple or complex, either abstract or detailed, an entity with a name and some relationships with other music objects.

Starting from this concept we can define as a music object, for example, both a structure of notes and a timbral characteristic. You can see in Fig. 1 four examples of music objects which belong to different levels of representation.

### Music operators

We assume that music pieces are made up of music objects and their relationships which, in general, grow dynamically from the beginning to end of the piece (Haus, 1979). So, more precisely, we can speak of their transformations vs. their relationships, because they can appear and disappear, change or repeat; but anyway they are the information base of the growing process within music pieces, from both creative and perceptual points of view.

We can simply say that a transformed music object  $S'$  is obtained applying the transforming process  $P$  to the original music object  $S$ ; the transforming process  $P$  may be made up of many operators. This schema is shown in Fig. 2.

In a previous research, one which has deeply influenced the present work, we explored how to describe music by means of homological operators (De Stefano & Haus, 1980; De Stefano et al., 1981).

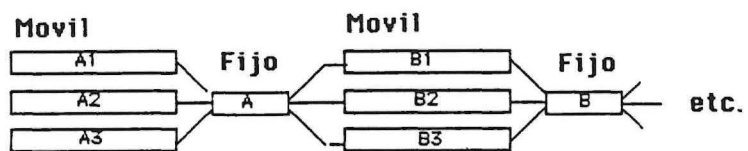
### New trends for computer music live performances

According to the traditional meaning, a concert of music is based on some "actors" (musical interpreters and performers) who produce sound processes using musical instruments (acoustical or electroacoustical or, in a particular case, the human voice) directed to the audience. Therefore, information which is produced and received during the concert depends on:

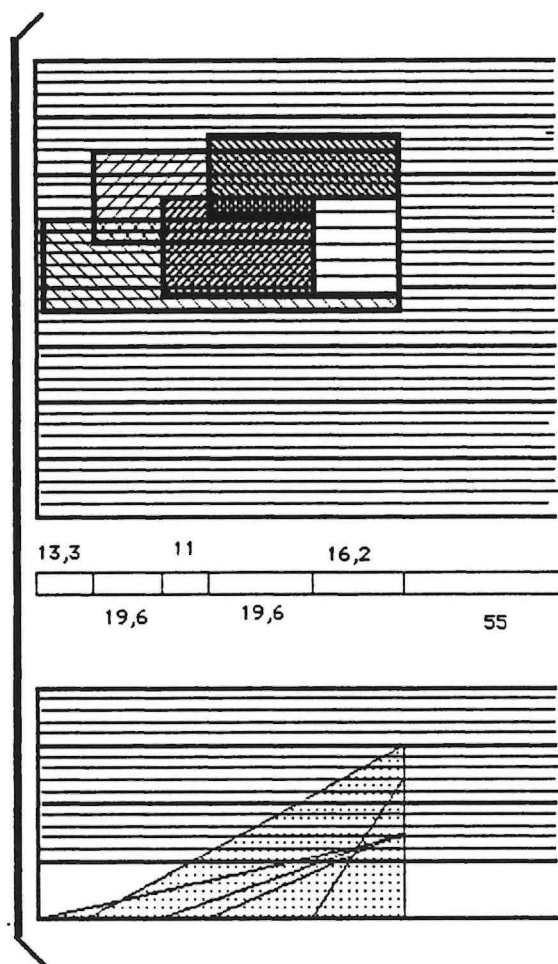
- a) musical interpretation;
- b) performance and interpretation on the instruments;
- c) spatial disposition of performers and instruments on the stage;
- d) morphological features of the concert-hall;
- e) executive pragmatics in the scenic sense (visual).

The evolution of this informational context has already gone through a great number of experiences which, in a variety of different ways, allowed the testing of a new form of spectacular fruition. We cite only some of the most significant:

- \* in advanced music (especially electronic), the magnetic tape concert: sound information depends only on loudspeaker position, visual information is almost meaningless, there are no human "actors";
- \* in a live electronic music concert, particularly with computer aid, sound spatialization: even in a traditional concert, sound is distributed in space, but according to a static scheme, while with electronic experiences the control of sound spatial distribution is generalized by means of models allowing live dynamic variation;
- \* in a computer-music concert, the possibility of changing timbre during the performance: performer's pragmatics change (Waisvisz, 1985), instruments (which act here on spectral content of sound rather than on sequences of



a)



c)

Figure 1a & 1c. music objects of the structural level (a) and electronic symbolic level (c).

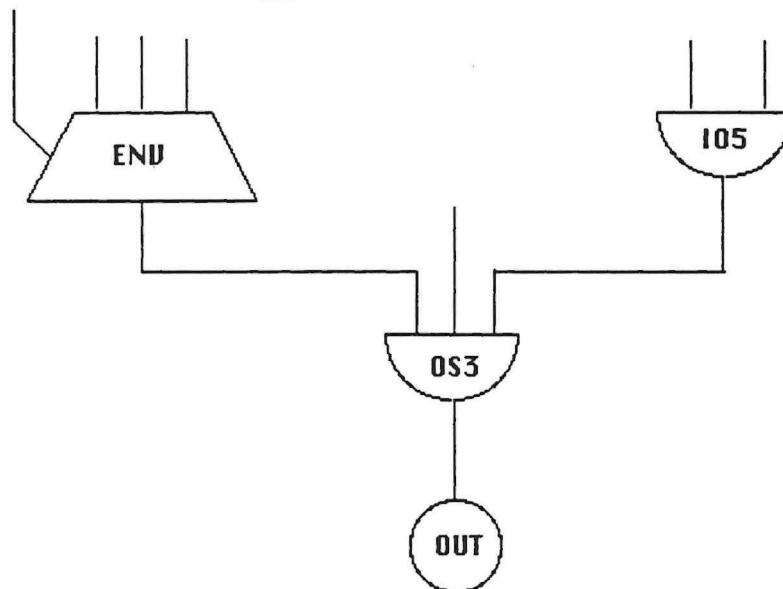


b)

```

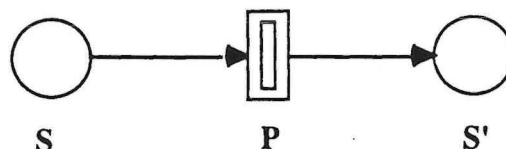
Instrument = INS 1
Mod. Envelope = ENV P5 dB fa b5 ATOK
                (P2,P8,P9,P9)
Mod. Linear Osc. = 1 o 5
Mod. Osc. = PSC3 b5 p5 Hz b6 b5 f1
Mod. OUT = OUT b5 b1
END

```



d)

Figure 1b & 1d. music objects of the traditional symbolic level (b) and timbre level (d).



timbrically preconstituted sounds) change (Loy, 1985); musical information (in which its more physical aspects are enriched) changes;

- \* many other kinds of performances in which cards of traditional concept are shuffled in regard to information content and channels through which information passes from the transmitter to the receiver (Haus, 1991).

Looking at the new scenario of music performance, we think it may be interesting for performers to have at their disposal tools which allow them to perform at a level more abstract than that of the traditional performance for two main reasons:

- a) the new computer-music instruments are able to work with music objects which are more abstract than notes; see, for example, the work by Zicarelli (1987);
- b) the new computer-music instruments make it possible to finely adjust timbre model during performance, so that performers need more time to control timbre.

In other words, we think in the future music performance may have a greater emphasis at the structural, timbre and operative levels vs. symbolic level actions.

Our research aims primarily to give the performer the possibility of acting on a keyboard at the structural level, so that his actions have more power and he can have more time to act at timbre and operative levels on specific controls and devices.

### PERFORMANCE FEATURES

Due to the motivations mentioned above, the availability of a musical instrument which can act at the structural level and involves few actions by the human performer makes a crucial point of both the definition and choice of music objects for the performance. There are two most relevant constraints conditioning the choosing process: the characteristics of transforming functions of the FP and the music statements desired. In fact, the kind of chosen music materials (i.e. predefined music objects) is more relevant with respect to the gestural behaviour, the FP characteristics and the timbral patches.

On the other hand, having more time free, the musician can design his performance in order to:

- a) apply transforming actions to timbral patches (i.e. the "orchestra") during execution;
- b) control by himself the mixing process;
- c) control by himself the spatialization of audio processes;
- d) control further devices for multimedia performance (by example, computer graphics in real time, etc.).

#### **Planning a performance: pre-definition of music objects**

As we show below in speaking about the FP system, the horizon of possible transformations actually concerns pitch and duration domains; then, music objects really are melodic fragments (sequences of notes or chords). Anyway, in spite of actual limitations, we can work with higher formalizing and quicker abilities both in the selection and synthesis processing of music texts.

It seems useful to make a distinction about the organization characteristics of thematic materials (melodic fragments' files):

- a) fragments which are fit for sequential performance (antecedent -> consequent) or denote causal relationships;
- b) fragments which tend away from sequential and causal logics and lead to a more advanced use of the implemented algorithms for music objects' transforming.

In fact, fragment files can acquire new functionalities if they are put in or called from a context of rules which is tending to not respect their initial semantic characteristics.

#### **Performance as improvisation: real-time definition of music objects**

Real-time definition of new music objects is now possible, so the musician can modify his thematic environment during performance: for example, basing changes on stimuli coming from the performance environment, both when music objects are ex novo defined and when they are obtained composing and decomposing the originals.

In that case, as in the case of real-time changes of dynamics and timbres (which we will discuss later), there are computational problems: solving the same represents one of the research topics currently going on at L.I.M.

#### **Real-time processing of music objects**

The real-time processing of music objects, that is, one of the main features for making a functional performance possible, involves the solving of some computing problems concerned with processes' concurrence and optimizing resources' management. Both are crucial factors in low-cost computing systems and require major efforts from the viewpoint of the design and implementation of software packages, in order to make up for computing power bounds. On the other hand, the problem is mitigated due to the need of music texts computing vs. audio signal processing, by reconfiguring the problem as a function of complexity and needs stated in the design phase of the system.

### Musician/computer interface requirements

It is necessary to provide the computer-music instrument with a suitable user interface to allow a kind of real-time performance satisfying the above stated requirements. The most relevant characteristics of such an interface have to be:

- a) the ease and immediateness of use; during performance there is little time: necessary acts have to be few and simple;
- b) the highest quality and lowest quantity of information; the performer must know about global flows of information and is not very interested in details he can only optionally control; this approach is suitable in making easier the understanding of control data (preferably expressed as graphics) about performance.

A hypothetical solution we have adopted concerns the use of the ASCII alphanumeric keyboard to immediately activate the transforming functions associated to the keys and the use of the mouse device to act onto a graphic representation of data and music objects' flows (either to make selections or to get global control operations, or as an alternative method to activate transforming functions).

### FP - FUNCTIONAL PERFORMER

A methodology of research for transforming and synthesizing music texts based on *homological operators* (De Stefano & Haus, 1980; De Stefano et al., 1981) was outlined with the implementation of the Functional Performer (FP); when we speak of homological operators we mean operators that apply homological transformations to music objects (homologies are a particular kind of geometric transformations whose natural environment is an n-dimensional space).

The FP system is distinguished from other music performance systems mainly for its real-time processing feature, which acts at the higher level of music representation: that is, the *structural level* (Haus, 1984), allowing essential and limited gestural intervention, i.e. the musician presses ASCII keyboard keys. In other words, we have a revaluation of gesture with respect to the pragmatics of traditional performance. Now gesture acquires a particular equilibrium: it is no longer a virtuoso act of the performer by means of traditional instruments (where a single sonic event corresponds to each act); but neither is it the insensitive act of the finger pressing a button to activate either a prerecorded sound process or a predefined automatic performance. Instead, it is an act to either control the execution of a music object or start a generative process of a new music object (see Fig. 2) by means of the transformation of a predefined music object. Therefore, the human performer becomes free from the constraints of a punctual execution of the music text (i.e. the score), acquiring major freedom to control other performance aspects (timbre patches, spatialization, etc.) and careful control over transformation processes of music texts.

### System architecture

The FP system may be defined as a system made up of a "slow" general purpose computer (for music text computing and synthesis) connected with some MIDI devices (for sound computing and synthesis).

In general, the FP has innovative aspects with respect to the upgraded power for controlling *concurrent processes*. It unfolds by means of *acts* (by the ASCII keyboard) devoted to the *processing of music primitives*, which can be represented as *melodic fragments*, defined as events of the pitch-duration kind. Pitches are related to the twelve-tone tempered scale.

For every group of user defined primitives characterized by homogeneity or heterogeneity criteria, we speak of a *file* (i.e. a library of music objects), whose dimensions (one or more fragments) and the criteria for material selection (intervals, scales, patterns, modes, series) can be newly defined as suitable with respect to the amount of transformations expected.

The use of polyphonic and polytimbral MIDI devices allows us to imagine a considerable range of *timbral solutions*. Then it is possible to define strategies for improving the organization of information flows toward peripherals in the most efficient way; furthermore, a feedback feature to assure the control of timbres' choice and primitives' modelling is very useful.

### The performance language

The operative environment we have spoken of allows the real time execution of *operative scores* represented as sequences of algorithms. It is also possible that the *functional performance* is free; in other words, that the performance is based on the ability of the musician to improvise variations (or, better, transformations) by means of the ASCII keyboard according to the file of predefined music primitives.

The various experiences the composer and the performer have with the FP system may be unified as a single kind of experience by a new pragmatics consisting of production, experimentation, growing, reflection, means' improvement, deeper definition of aims (Stiglitz & Tanzi, 1989).

### Implementations

The FP system was conceived by Haus and realized in the earlier versions on a MSX computer by Pagni (1986) and successively by Stiglitz (1988); a further version, currently under development, was then designed by Madeddu (1992) for the Macintosh.

### The MSX version

The MSX version of the FP system was developed between 1987 and 1988. It is based on an MSX2 ( $\mu$ p Z80A) microcomputer equipped with 128 KB RAM, 128 KB VideoRAM and floppy disc drive. The computer is connected to a Yamaha SFG-05 FM sound synthesis module with MIDI interface for communicating with other MIDI equipments. The software environment is made up of the MSX

Disc Basic and the Yamaha ROM FM Music Macro II including the MSX Disc Basic extension for SFG-05 module controlling.

The system has the following functionalities:

- a) creation of melodic fragments' files with related functions for modifying, storing, loading and printing;
- b) real-time execution/transformation of melodic fragments;
- c) deterministic and pseudo-random functions for executing and transforming;
- d) 4 voices canon imitation (polytimbral);
- e) Change Program MIDI functions;
- f) operative scores' employment (see the two examples at the end of the paper);
- g) direct control of the SFG-05 module, employing it as a synthesis module.

The module for the editing of melodic fragments is separated from the performance one due to the lack of RAM.

The real-time transformation operators available are the following:

- pitch mirror inversion
- durations mirror inversion
- intervals dilatation
- intervals shrinkage
- transposition up
- transposition down
- performance acceleration
- performance deceleration
- pitch retrogradation
- durations retrogradation
- total retrogradation
- data flow control (MIDI channel redirection and/or MIDI program changes, melodic fragments selection, etc.)

Almost all the operators are available in two distinct versions: QUICK and RANDOM. The QUICK version allows us to directly transform the original music object, and the new one is immediately executed, while the RANDOM version produces a sequence of intermediate steps between the original music object and the new one. The number of intermediate steps can be set by the user; they are synthesized by means of a pseudorandom algorithm that substitutes elements of the original music object with new ones until the complete new transformed music object is obtained.

From the viewpoint of implementation, we touch on the following four main topics:

- a) musical data representation;
- b) transformation operators;
- c) parallelism management and timing;
- d) user interface.

We briefly show some solutions we have found to solve the problems posed by the hard limitations of computing power and memory capacity of the MSX-MUSIC MACRO h/w and s/w system.

### **Musical data representation**

When the MSX system configured as described above starts up, only 19 KB are available for the BASIC FP program and for the related data. In the development phase of the FP program, we made tests to verify some hypotheses about the musical data representation to find the best way in terms of shortness and efficiency. The results of the tests led us to represent each melodic fragment by one couple of arrays of integers (the first array for the pitches and the second for the corresponding durations). Consequently every melodic fragments' files consists in sets of couples of arrays of integers.

### **Transformation operators**

In order to simplify (and then speed up) the computing processes, we have imposed the same length on all arrays belonging to one melodic fragments' file; and we have defined the algorithms of the transformation operators so as to conserve the length of the arrays representing the melodic fragments. Then the algorithms of the transformation operators are sequences of arithmetical and/or logical expressions operating on the integer numbers representing the indexes or the elements values of the arrays.

### **Parallelism management and timing**

The s/w system "MSX Disc Basic+Yamaha ROM FM Music Macro II" implements the process of the musical texts playing in a similar way as the usual sequencers. Therefore we developed the FP application program by exploiting this feature fully.

Starting from arrays of numerical values for pitches and durations (the musical text), the application program makes some sequencer-like tracks (maximum 8 tracks in our s/w system). The tracks contain the data of the melodic fragments in the correct time playing order and some special purposed FM Music Macro II commands allow us to control the playing of the tracks. But the s/w system doesn't allow us to update one track when it is playing. Therefore, in order to update the tracks with the new data generated run time by the transformation operators, the application program uses 2 tracks for each voice, limiting to 4 the number of the voices available. While the first track of one voice is playing, the application program updates the second one and viceversa. In order to recognize when one track has finished playing, the program uses an error-trapping method provided by the s/w system.

Another feature of the s/w system is the following: it is possible to update more tracks in different times (in an asynchronous way), and successively to start contemporareously (in a synchronous way) the playing of all the tracks updated. The application program uses this feature to manage the polyphonic synchronization of the canon imitation structure.

### User interface

Due to severe limitations of the computer, particularly with respect to RAM, we have chosen to renounce a complete graphic user interface; meanwhile, we use graphics features to enhance information useful for the performer's system controlling activity.

The user controls the music real time performance by pressing the keys of the MSX computer ASCII keyboard. The keys are related to the control functions and to the real time transformation operators implemented into the FP program. By pressing the keys, the user executes the functions or the operators directly. Figure 3a,b shows the MSX computer ASCII keyboard functional map.

### The Macintosh version

The emergence on the market of Macintosh PowerBook (the Macintosh portable model) has provided a very powerful platform for running the FP program. So, we have multiplied the efforts to develop short-term the Macintosh version of the FP program.

This version offers many improvements in terms of graphics, ergonomics, efficiency and functionalities with respect to the MSX; furthermore, it can import/export Standard MIDI File 1.0 scores. Therefore it is possible to save a Standard MIDI File 1.0 score containing the musical data generated real-time during one performance and to use them successively with some other music application programs.

The musical data are represented in internal format by means of sequences of records. Each record is associated to one quantized instant in the time axis and contains the data related to the events occurring in that instant.

The data in the records are updated run time and the application program manages the timing process to synchronize the playing of the different tracks (8 polyphonic tracks are available at maximum).

Compared to the MSX version, the transformation operators set as well has grown.

The following functions are defined:

- pitch mirror inversion
- durations mirror inversion
- dynamics mirror inversion
- intervals dilatation
- intervals shrinkage
- durations dilatation
- durations shrinkage
- dynamics dilatation
- dynamics shrinkage
- transposition up
- transposition down
- performance acceleration
- performance deceleration

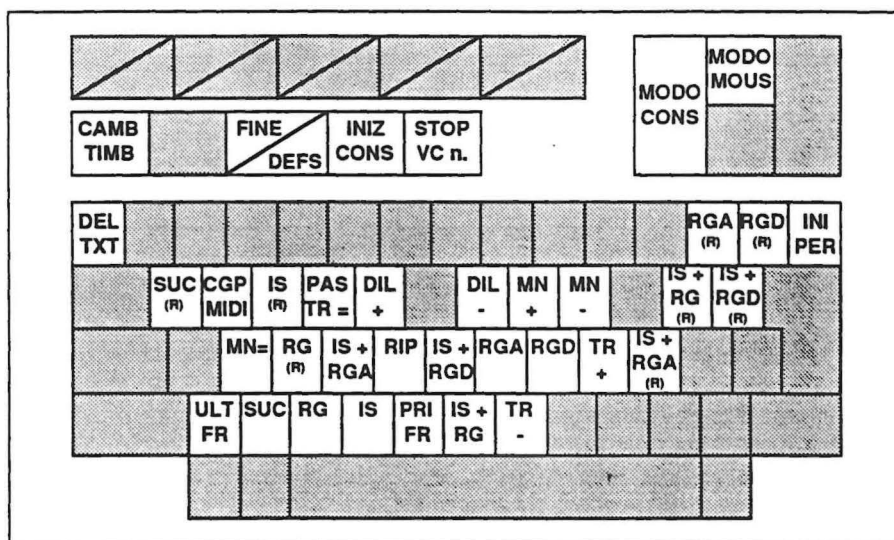


Figure 3a. The MSX computer ASCII keyboard functional map.

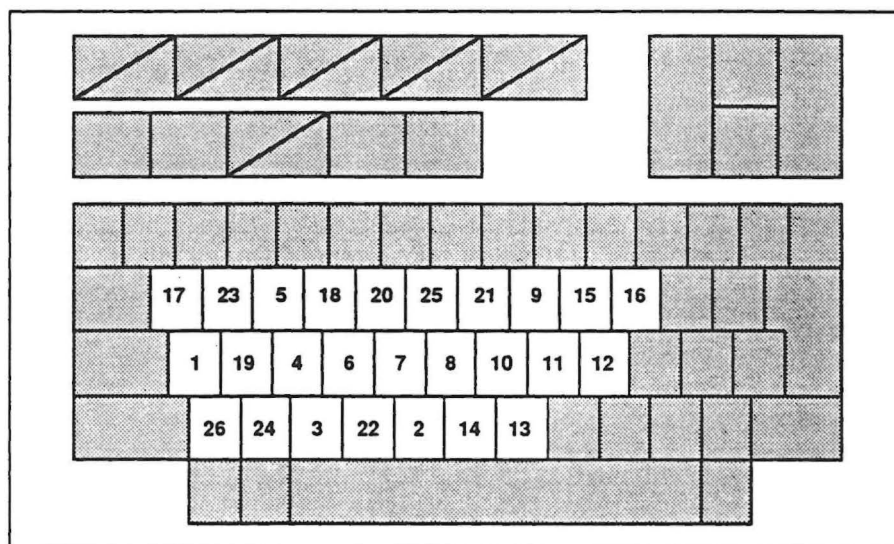


Figure 3b. The MSX computer ASCII keyboard numerical data map.

- dynamics crescendo
- dynamics diminuendo
  
- pitch rotation
- durations rotation
- dynamics rotation
- total rotation
- pitch retrogradation
- durations retrogradation
- dynamics retrogradation
- total retrogradation
  
- subfragments multiplication
- subfragment notes filtering
- subfragment notes cutting
- complementary subfragment notes filtering
- complementary subfragment notes cutting
  
- fragments merge with left alignment
- fragments merge with right alignment
- fragments merge with centering
- fragments merge with justification

Therefore it is possible to define some musical dynamics' values masks; the masks could implement, for example, some metric models. These masks can be applied at run-time to filter the data related to the musical objects to make the musical performance follow some predetermined rhythmical structures.

The user interface is completely graphic: Figure 4 shows the main window. The score is represented by a music object map made up of 8 strips (one for each polyphonic track available) within the window on which a time scanning bar slides. The time axis can be quantized by either beats and ticks or n-seconds' blocks. The 8 strips are associated to 8 of the 16 MIDI channels; the association table is built by the user.

When the user selects a fragment, by clicking the mouse into the coloured box associated to the music object he is interested in, that object becomes active and its total duration is computed and a graphic copy of it is drawn below the music object map; then it is possible to apply some operators, selected by the available set of transformations, to the active music object.

A fragment or its transformed instance may be object of further transformation until it is put into the score. This feature allows us to apply composite functions (in the algebraic sense).

When the user puts an object into the score, after the dragging, the mouse button has to be released only when we are on the desired strip. The fragment aligns itself on the left where the nearest reference point of the temporal grid is

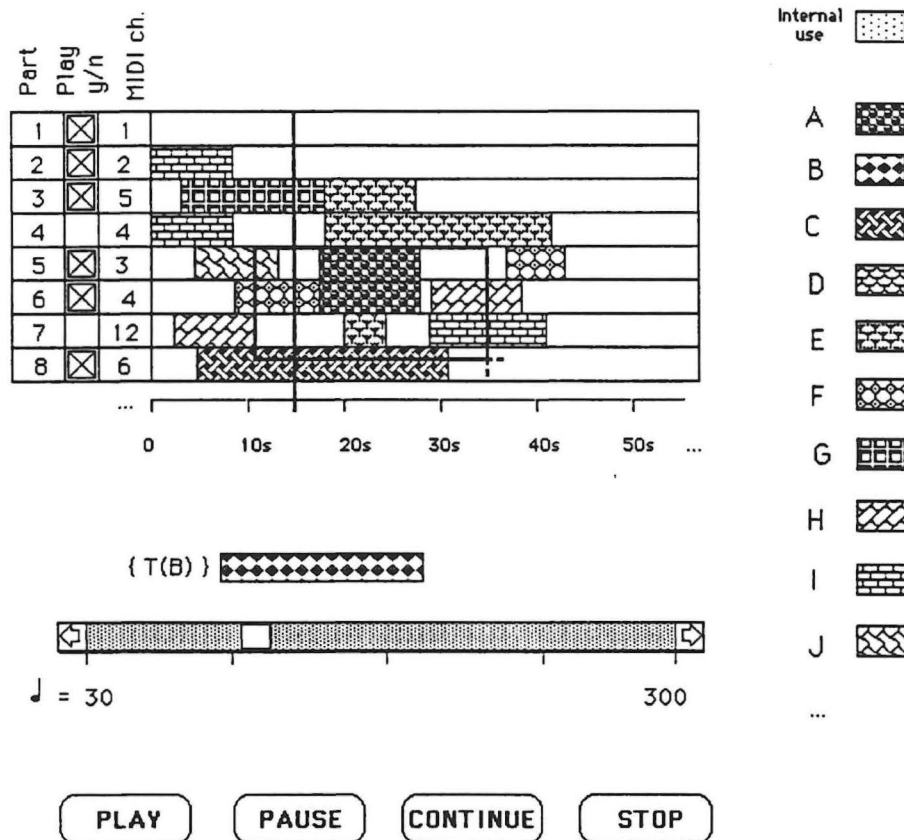


Figure 4. the main window of the FP system (Macintosh version).

available. The grid density is defined by the user within the set  $\langle 1, 2, 3, 5 \rangle$ . When the user asks for density changing, the new grid is active starting from the point where the time scanning bar is.

When the user selects a graphic music object from the score already synthesized, a new graphic pattern (labeled "internal use" in Figure 4) is assigned by default. Starting from that moment the selected music object is treated as a new one and can be put again into the score after some transformations activated by the user.

#### FP and performance orchestration

We have seen that the FP system actually manages only melodic fragments: that is, the music objects' transformations affect only one plane (i.e. the melodic one) of the sound space (De Stefano/Haus, 1980). Then dynamics and timbres are not affected by transformations. In the ideal real-time global music performance, the

performer would also directly control these two sound components. But before that we have to solve some problems related to the computing performances of low cost systems, and we rely on some powerful and cheap DSP boards to add on FP systems.

At present we are adopting some alternative solutions:

- a) we delegate the single notes' MIDI parameters to control local dynamics (the performer can set up the values of these parameters at the moment of melodic fragments defining); in the Macintosh version of the FP there is also the possibility of defining before the performance some musical dynamics' values masks;
- b) we delegate the timbral patches' set up of single digital music instruments and the real time mixing operations to control the dynamics of more general levels (i.e. single instruments);
- c) in regard to the timbral aspects and the orchestration, we have to program the digital music instruments before the performance to define a set of timbral patches which will be changed during the performance by MIDI *Program Change* messages sent by the performer.

#### COMPOSITION AND PERFORMANCE EXPERIENCES WITH THE FUNCTIONAL PERFORMER

Since 1988, we have been using the FP system for music and multimedia live performances. The work of designing and making performances has allowed us to set up a working methodology for the analysis and checking of the performing environments' characteristics (Tanzi, 1990). The reflections' main topics are:

- a) *time-gesture* interaction, which depends on the FP model of generating *the* and interacting *with* music events;
- b) *sound-gesture* interaction, which depends on the relations between computers and music devices (these relations are different for every single composition);
- c) *image-gesture* interaction, which depends on the characteristics of further devices for multimedia performance;
- d) organization of *operative scores*, realized as sequences of algorithms;
- e) interaction among a few FP systems and interaction between the FP systems and any set of *acoustic instruments* ;
- f) interaction between all electroacoustic ensemble members and the *sound engineering and mixing*.

The FP system offers valuable experimentation' opportunities to anyone interested in making music in a way not purely instinctive. These opportunities can make the process of music ideas' *designing* as an *inquiring* into the cultural matrices which the composer would draw from. Using FP, the composer can study the relations between music objects and the rules which govern concurrency,

The figure shows 12 staves of musical notation, labeled A through L. Staves A through G are in treble clef, and staves H through L are in bass clef. The notation includes various rhythmic values (eighth, sixteenth, and quarter notes) and rests. Some staves have articulation marks like slurs and accents, and some have fingerings indicated by numbers 1, 2, 3, 4, 5, and 6 above the notes.

Figure 5. Melodic fragments for FP1 in Pro.Fumo (Dante Tanzi - 1988).

the causality relations or even the music events' indetermination in a barely changing universe (i.e. the functional performance).

In conclusion, it is important to point out that in this work, as well as in other

step : 4  
time : 60

Instruments : MSX Philips  
timbres S900 (sampling)  
timbres TX802 (FM synthesis)

## I

INIT (8)  
NEXTQ (I)  
NEXTQ (J)  
  
NEXTQ (K)  
  
NEXTQ (L)  
NEXTR (E)  
RGDR  
TR + (D)  
RGHTR  
INTERV - (A)  
INTERV - (A)  
CHNGPROG (1) A'  
RESTART (8)  
NEXTQ (I)  
NEXTQ (J)  
NEXTQ (K)  
NEXTQ (L)  
NEXTR (E)  
RGDR  
TR + (E)  
INTERV - (A)  
INTERV - (A)  
NEXTR (E)

## II

CHNGPRG (3)  
RESTART (1)  
NEXTQ (B)  
MN +  
NEXTQ (C)  
MN +  
RGDR  
DIL + (C)  
TR + (B)  
RGHTQ  
NEXTQ (D)  
DIL - (A)  
DIL - (A)  
DIL - (A)  
DIL - (A)  
C CHNGPROG (4)  
RESTART (5)  
RGDR  
NEXTQ (F)  
NEXTQ (G)  
RESTART (8) ←  
RGHTR  
NEXTQ (K)  
RGHTR  
NEXTQ (K)  
RGHTR  
D CHNGPROG (3) →  
or  
CHNGPROG (4) →

Figure 6. operative score for FP1 in Pro.Fumo (Dante Tanzi - 1988).



Figure 7. Melodic fragments for FPI in Louvre (Dante Tanzi - 1989).

L.I.M. works, we propose a substantial change of mentality, plus a rejection of the bad habit of considering acquired a *fixed hierarchy* of relations between the means and the aims, or between the music languages and music events, or be-

step : 2  
time : 80

timbres S900 (sampling) : flatonly-louvre  
timbres TX802 (FM synthesis) : louvre

|        |          |                  |
|--------|----------|------------------|
| 0. 50" | INIT     | ( 1 ) only TX802 |
| 1. 00" | NEXT/Q   | ( K ) also S900  |
| 1' 20" | TR +     | ( L )            |
| 1'45"  | IS + RGD | ( A )            |
| 2'00"  | TR -     | ( A )            |
| 2'12"  | NEXT/R   | ( D )            |
| 2'30"  | NEXT/R   | ( G )            |
| 2'47"  | IS + RGA | ( B )            |
| 3'45"  | AGAIN    | only TX802       |
| 4'55"  | AGAIN    | only TX802       |
| 6'00"  | TR -     | ( A )            |
| 6'12"  | TR -     | ( A )            |
| 6'24"  | TR -     | ( A )            |
| 6'36"  | TR -     | ( A )            |
| 6'48"  | TR -     | ( A )            |
| 7'00"  | PROX/R   | ( F )            |
| 7'25"  | TR +     | ( A )            |
| 7'37"  | TR +     | ( A )            |
| 7'49"  | TR +     | ( A )            |
| 8'01"  | TR +     | ( A )            |
| 8'13"  | TR +     | ( A )            |
| 8'25"  | TR +     | ( L )            |
| 8'40"  | TR +     | ( L )            |
| 8'55"  | TR +     | ( L )            |
| 9'10"  | TR +     | ( L )            |
| 9'25"  | TR +     | ( L )            |
|        | END      |                  |

Figure 8. Operative score for FP in Louvre (Dante Tanzi - 1989).

tween the scores and the processes, or between the symbols and the structures.

We show two examples which represent two important moments of this research:

**Ex. 1. Pro. Fumo (Dante Tanzi - 1988)**

"PRO.FUMO", for *two FP systems (MSX version)*, was performed for the first time at VII Colloquium on Musical Informatics (Rome, 1988).

During the performance, the timbral patches have to change by *program change* commands, and the polyphony (4+4 real parts) have to increase to a highest number of 16 parts. The music objects for the transformations belong to one melodic fragments' file splitted in two subsets and the two FP systems (FP1 and FP2) follow two distinct operative scores.

In Fig. 5 we show the melodic fragments' file for the part of FP1; in Fig. 6 we show also the respective operative score.

**Ex. 2. Louvre (Dante Tanzi - 1989)**

"LOUVRE", for *flute, percussions, Functional Performer (MSX version) and Mouse Performer* (another program for music live performance developed at L.I.M.), was performed for the first time at the Musica del Nostro Tempo L.I.M. concert (Milan, 1989).

"LOUVRE" is the first work made at L.I.M. for real time computer instruments and acoustic instruments' live performers.

In Fig. 7 we show the melodic fragments' file for the part of FP: it is one melodic theme split in 11 segments that is the basic reference for the definition of main structures of composition. In Fig. 8 we show also the respective operative score.

## ACKNOWLEDGEMENTS

Special thanks are due to Dante Tanzi for the extensive experimentation of Functional Performer within his own compositions.

Many thanks to Antonio Rodriguez and Dante Tanzi, for the constructive discussions we have had to specify the functionalities of the software tools, to Claudio Pagni, for his software development activity related to the MSX version of the Functional Performer, and to Francesco Madeddu, for his software development activity related to the Macintosh version.

This research has been partially supported by the Italian National Research Council in the frame of the MUSIC Topic (LRC C4): "INTELLIGENT MUSIC WORKSTATION", Goal C: SISTEMI AVANZATI DI PRODUTTIVITA' INDIVIDUALE, Subproject 7: SISTEMI DI SUPPORTO AL LAVORO INTELLETTUALE, Finalized Project SISTEMI INFORMATICI E CALCOLO PARALLELO.

## REFERENCES

- Camurri, A., Haus, G., Stiglitz, A., & Zaccaria, R. (1989). *Tema MUSIC: architettura, specifiche funzionali di alto livello e standard di documentazione della Stazione di Lavoro Musicale Intelligente* (Tech. Report 7/09, Finalized Project "Sistemi Informatici e Calcolo Parallelo"). Roma: Italian National Research Council.
- Degli Antoni, G., & Haus, G. (1983). Music and Causality. *Proceedings of the 1982 International Computer Music Conference* (pp.279-296). San Francisco: Computer Music Association.
- De Poli, G., & Haus, G. (1982). Ingegneria del software e informatica musicale. *Proceedings of the Annual Congress Associazione Italiana Calcolo Automatico* (pp.415-430). Padova: Università di Padova.
- De Stefano, S., & Haus, G. (1980). A Mathematical Approach to the Representation of Musical Structures. *Proceedings of the 2° International Conference on Data Bases in the Humanities and Social Sciences*. Madrid: Facultad de Informatica.
- De Stefano, S., Haus, G., & Stiglitz, A. (1981). Descrizione di processi musicali per mezzo di operatori geometrici: un esempio applicativo. *Proceedings of the 4° Colloquium on Musical Informatics* (pp.37-61). Pisa: CNUCE-CNR.
- Haus, G. (1979). Trasformazione di testi musicali per mezzo di operatori. *Proceedings of the 3° Colloquium on Musical Informatics* (pp.178-193). Padova: Università di Padova.
- Haus, G. (1984). *Elementi di informatica musicale*. Milano: Gruppo Editoriale Jackson.
- Haus, G. (1992). Multimedia Performance: Experiences and Discussions. submitted to *Interface*.
- Loy, D. G. (1985). Designing an Operating Environment for a Realtime Performance Processing System. *Proceedings of the 1985 International Computer Music Conference* (pp.9-13). San Francisco: Computer Music Association.
- Madeddu, F. (1992). *Un sistema per la performance musicale funzionale*. Unpublished doctoral dissertation, Università degli Studi, Milano.
- Pagni, C. (1986). *Generazione automatica di strutture musicali ritmiche e melodiche per l'elaboratore CX5M*. Unpublished doctoral dissertation, Università degli Studi, Milano.
- Pope, S. T. (1986). The development of an intelligent composer's assistant: Interactive graphic tools and knowledge representation for music. *Proceedings of the 1986 International Computer Music Conference* (pp.131-144). San Francisco: Computer Music Association.
- Stiglitz, A. (1988). *Un sistema per l'elaborazione automatica di strutture musicali mediante operatori geometrici*. Unpublished doctoral dissertation, Università degli Studi, Milano.
- Stiglitz, A. (Ed.) (1991). *Specifiche funzionali di alto livello dei moduli della Stazione di Lavoro Musicale Intelligente - Unità Operativa dell'Università degli Studi di Milano* (Tech. Report M/06, Finalized Project "Sistemi Informatici e Calcolo Parallelo"). Roma: Italian National Research Council.
- Stiglitz, A., & Tanzi, D. (1989) Strumenti software per la performance in tempo reale di Louvre. *Proceedings of the 3° Colloquium on Musical Informatics* (pp.227-234). Cagliari, Italy: AIMI/Spaziomusica.
- Tanzi, D. (1988). PRO.FUMO. *Proceedings of the 7° Colloquium on Musical Informatics* (pp.167-168). Roma: AIMI/CIDIM/SIM.
- Tanzi, D. (1990). *Manipolazione di testi musicali con il Functional Performer (L.I.M. Tech. Report)*. Milano: Università degli Studi.
- Waisvisz, M. (1985). The HANDS: A Set of Remote MIDI Controllers. *Proceedings of the 1985 International Computer Music Conference* (pp. 313-318). San Francisco: Computer Music Association.
- Zicarelli, D. (1987). M and Jam Factory. *Computer Music Journal* 11(4), 13-29.



Goffredo Haus, Scientific Director  
Laboratorio di Informatica Musicale  
Dipartimento di Scienze dell'Informazione  
Università degli Studi di Milano  
via Comelico, 39  
I-20135 Milano  
ITALIA

G. Haus has been the Scientific Director of L.I.M.-Laboratorio di Informatica Musicale since its founding. Since 1992 he has been working as an associate professor in Cybernetics at the Dept. of Computer Science of the University of Milan. He is one of the founders of the Italian Association of Musical Informatics (A.I.M.I.) of which he has been Board Member (81-89) and President (84-87). He has also been Board Member of Computer Music Association (82-87).

He has published about 200 scientific-technical papers as well as the academic text "Elementi di informatica musicale". He has organized and directed two-year courses on Electronic Music, at the EMIT in Milan (84-86). He coordinates many research projects on scientific, technological, aesthetic and multimedia topics; in that frame he cooperates with many Italian and international institutions and industries. He has fundings from the Italian National Research Council and from the Ministry of University and Research.

He is also a composer of music by computer methods at both the score and the sound level. Among his works, the "Musiche per poche parti", whose scores have been synthesized by Petri Nets execution, have been published by StileLibero/VirginDischi (SLCD0012 and SLLP0012).



A. Stiglitz Studied Electronic Music Composition at the "G. Verdi" Conservatory of Milan (with M<sup>o</sup> Angelo Paccagnini). He studied also at the University of Milan, earning his degree in Physics (his thesis work was developed at L.I.M. with Goffredo Haus and Giovanni Degli Antoni).

Since January 1989 he has been working at L.I.M. involved in some musical informatics research projects: for example, the Finalized Project "Sistemi informatici e calcolo parallelo - STAZIONE DI LAVORO MUSICALE INTELLIGENTE" of the Italian National Research Council.

Since June 1990 he has also been working for the Philips company as a designer of CD-I (Compact Disc Interactive) applications.

