A SOFTWARE TOOL FOR THE FUNCTIONAL PERFORMANCE OF MUSIC †

Goffredo Haus & Alberto Stiglitz

L.I.M. - Laboratorio di Informatica Musicale
Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano

INDEX

† This research is partially supported by the Italian National Research Council in the frame of the MUSIC Goal: "Intelligent Music Workstation", Subproject 7 SISTEMI DI SUPPORTO AL LAVORO INTELLETTUALE, Finalized Project SISTEMI INFORMATICI E CALCOLO PARALLELO.

0. Abstract

Music performance is traditionally made up by sequences of events (notes and chords, by 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, specular inversion, juxtaposition, superimposition, time shrinking, etc.); 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.

Therefore, FP is a tool for the performing of music structures and their processing; the executed score is depending on the music patterns previously defined (music objects) and gestural acts we do during performance (music functions). The same sequence of gestural acts may give different results with respect to music patterns; particularly, if we only use the subset of linear functions we have a kind of music which is consistent with the music patterns: if we define serial patterns we get serial music, if we define modal patterns we get modal music. Otherwise, we can use non-linear functions which modify the syntactic characteristics of the music patterns.

At present, the first version of FP is completed and is usually experimented at L.I.M. concerts; it runs on MSX computers. A second version is under development on the Macintosh II family of computers; it has many improvements on graphics, ergonomics, efficiency and functionalities and can import/export Standard MIDI File scores.

1. Introduction

In the music domain, the instruments revolution caused by the coming of digital devices has cast doubt on the classical roles of the protagonists of music production and enjoyment processes: the composer, the performer and the listener. In fact the new informatic music workstation features allow the musician to assume all the 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 one each other to build complex systems; this kind of experiences leads to a new idea of music instrument and related making techniques.

Particularly, we have had two main aims in the design of the FP: an original and advanced use of commercially available low-cost devices and the realization of an informatic music instrument for the real time performance.

In the following paragraphs we show the most relevant features of the FP with respect to similar systems; then we briefly describe its working with some examples.

2. The conceptual environment

This research has been carried on in the frame of the more general "Intelligent Music Workstation" Project [Camurri et al., 1989]; within all this project we have accepted as a standard some concepts which seem to have a basic role both in music thinking and understanding i.e. in music creation and perception; therefore, in the following paragraphs we will briefly discuss about: music levels of representation and related languages, music objects and their transformations by operators, evolution of music performance pragmatics.

2.1. Music levels of representation

A relevant feature introduced by computer applications to music is the possibility to describe music information at many different levels of abstraction [Haus, 1984] [Pope, 1986]; those levels may be concurrently accessed depending on the user will; so, one can choice the level which is closer to the activity the user wish to do, without restrictions on moving up and down within hierarchical music descriptions during user-system interaction; we may think as if we have 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 the following levels (see also [De Poli/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 on the traditional score and also by composing and sequencing software packeges);
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: by example, music performance requires actions mostly regarding notes but also timbre models and timing control. Therefore, three levels are involved: symbolic, timbre and operative.

2.2. Music objects

In our research we have adopted the notion of music object [Degli Antoni/Haus, 1983] which is more general than the music event one and it is also less related with a particular level of abstraction. Music object means anything that could have a musical meaning and that we think 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 belongs to different levels of representation.

2.3. Music operators

We assume that music pieces are made up of music objects and their relationships which, in general, dynamically grow from the beginning to the 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, both from the creative and the 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.

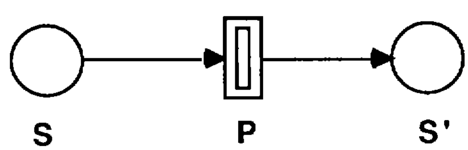
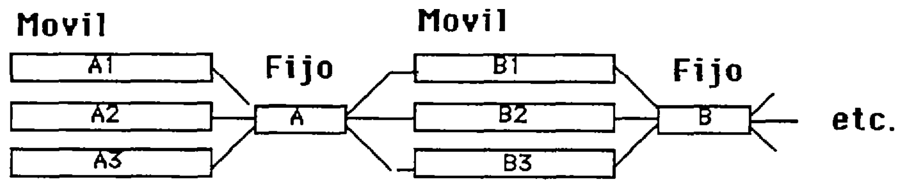
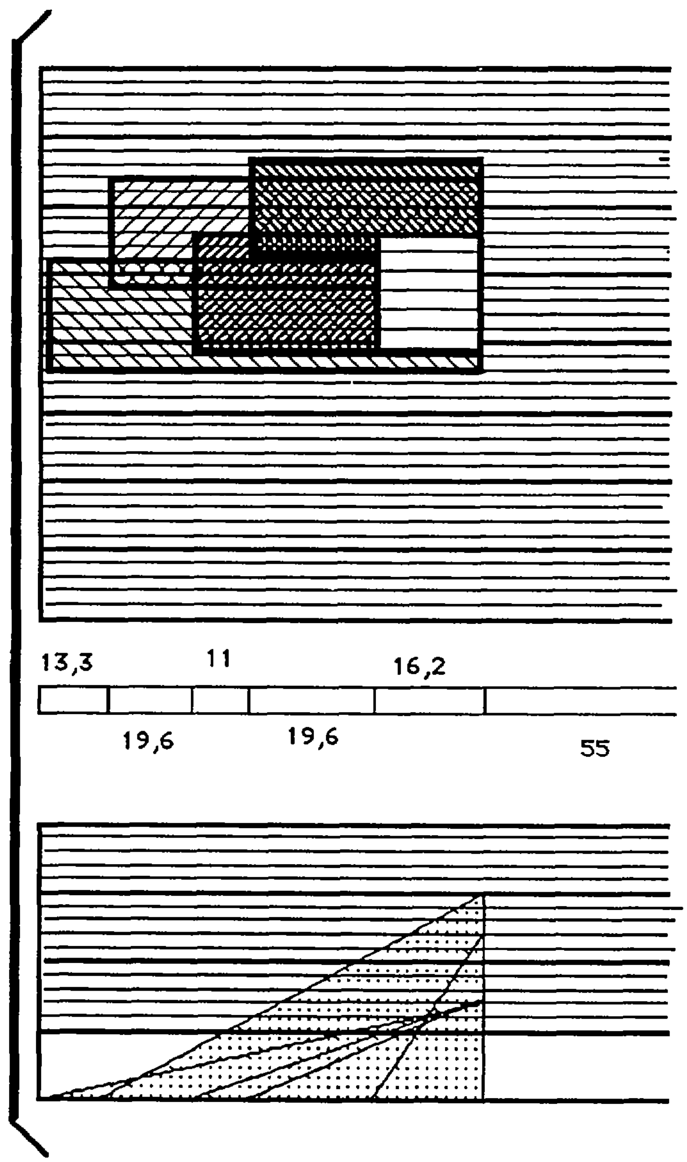


Figure 2: the elementary music object transformation process.

In a previous research, which has deeply inluenced the present work, we have explored how to describe music by means of homological operators [De Stefano/Haus, 1980] [De Stefano et al., 1981].



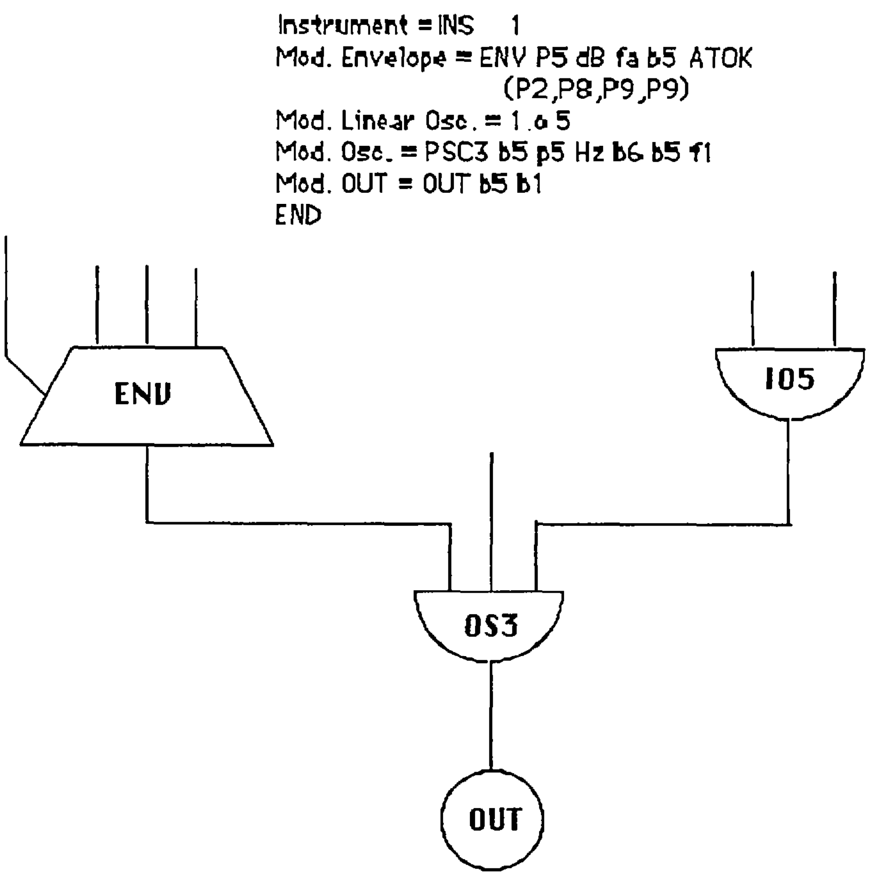
a)



c)

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

b)



d)

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

2.4. New trends for computer music live performances

The music concert by the actual traditional meaning is based on some number of "actors" (musical interpreters and performers) who produce sound processes using musical instruments (acoustical or electroacoustical or, particular case, 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) morphologic features of the concert-hall;
e) executive pragmatics in the scenic sense (visual).

Evolution of this informational context has already passed across a great number of experiences which, with a variety of different ways, allowed to test new form of spectacular fruition; we cite only someone of the most significant:

* in advanced music (especially electronic), the magnetic tape concert: sound information depends only on loudspeaker position, visual information is almost unmeaningful, there are no human "actors";

* in live electronic music concert, particularly with computer aid, sound spatialization: even in 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 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 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 as to information content and channels by which information passes from the transmitter to the receiver [Haus, 1990].

Looking at the new scenario of music performance we think that may be interesting for performers to have at their disposal tools which allow to perform at a more abstract level than that of the traditional performance one 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, [Zicarelli, 1987]);
b) the new computer music instruments make possible to finely adjust timbre model during performance, so that performers need more time to control timbre.

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

Our research has as its main aim to give the possibility to the performer 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.

3. Performance features

Due to the motivations we have spoken above, the availability of a music instrument, which can act at the structural level and engrosses a few actions by the human performer, makes a crucial point both the definition and the choice of music objects for the performance. Two are the 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 the choosen 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 the 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.).

3.1. Planning a performance: pre-definition of music objects

As we show in § 4. 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 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 are tending 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.

3.2. 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, by 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 original ones.

In that case, as in the case of real time changes of dynamics and timbres (see § 4.4.), there are computational problems, the solving of which represents one of the research topic currently going on at L.I.M.

3.3. Real-time processing of music objects

The real time processing of music objects, that is one of the main features to make possible a functional performance, involves the solving of some computing problems concerned with processes' concurrency and optimizing resources' management. Both of them are crucial factors for low cost computing systems and then require major efforts from the point of view of the design and implementation of software packages, 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 which were stated in the design phase of the system.

3.4. 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) easiness and immediateness of use; during performance there is a few time: needed acts have to be few and simple;
b) highest quality and lowest quantità of information; the performer must know about global flows of information and is little interested in details that, anyway, can optionally control; this approach is suitable to make easier the understanding of control data (preferably expressed as graphics) about performance.

A hypothesis of solution we have adopted concerns the use of the ASCII alphanumeric keyboard to immediately activate the transforming functions, that are 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.

4. The Functional Performer system

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, on the other hand, 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 more a virtuoso act of the performer which acts on traditional instruments (where a single sonic event corresponds to each act), but it is not even the insensitive act of the finger that presses a button for activating 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 on transformation processes of music texts.

4.1. 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* , which unfolds by means of *acts* (by the ASCII keyboard) devoted to the *processing of music primitives* , that 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 etherogeneity 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 it is suitable with respect to the amount of transformations expected.

The use of polyphonic and polytimbral MIDI devices allows 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.

4.2. 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* be free; in other words, that the performance be based on the ability of the musician to improvise variations (or, better, transformations) by means of the ASCII keyboard according with the file of predefined music primitives.

The various experiences the composer and the performer have by the FP system may be unified as an only kind of experience by a new pragmatics made up of production, experimentation, growing, reflection, means' improvement, deeper definition of aims [Stiglitz/Tanzi, 1989].

4.3. Implementations

The FP system was conceived by Goffredo Haus and realized in the earlier version on an MSX computer by Alberto Stiglitz at the L.I.M. - Laboratorio di Informatica Musicale of the Computer Science Dept., State Univ. of Milan; a further version, currently under development, was then designed by Marco Benvestito for the Macintosh.

4.3.1. The MSX version

The MSX version of the FP system was developed between 1987 and 1988; it is based on an MSX2 (μp 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 to modify, store, 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;
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 specular inversion
- durations specular 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 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.

Due to the heavy limitations of the computer, particularly with respect to RAM, we have chosen to renounce to a complete graphic user interface, while graphics features are used to enhance informations that are useful for the system controlling activity of the performer.

4.3.2. The Macintosh version

The Macintosh version of the FP system has many improvements on graphics, ergonomics, efficiency and functionalities with respect to the MSX one and can import/export Standard MIDI File 1.0 scores. The user interface is completely graphic and the transformation functions' set has grown. Figure 3 shows the main window. The score is represented by a music object map made up of 8 strips 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 we select a fragment, by clicking the mouse into the coloured box associated to the music object we are 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. The following functions are defined:

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

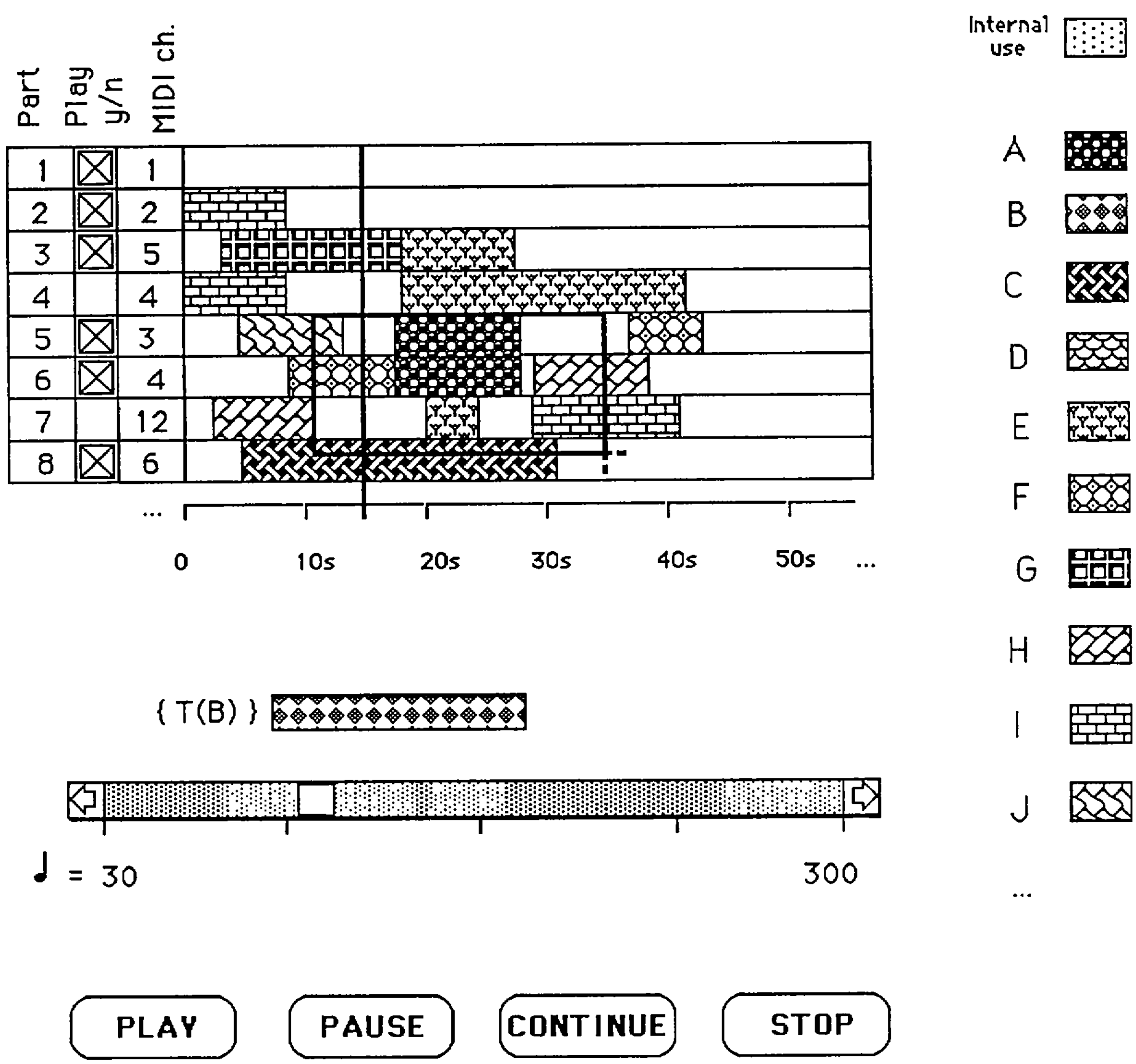


Figure 3: the main window of the FP system (Macintosh version)

- fragments merge with centering
- fragments merge with justification

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

When we put an object into the score, after the dragging, the mouse button has to be released only when we are upon the desired strip. The fragment aligns itself on the left where it is available the nearest reference point of the temporal grid. The grid density is defined by the user within the set <1, 2, 3, 5>. When the user ask for density changing, the new grid is active starting from the point in which the time scanning bar is.

When we select a graphic music object from the score already synthesized a new graphic pattern (labeled "internal use" in Figure 3) 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.

4.4. Functional Performer 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 directly control also these two sound components. But before of 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 adopt 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);

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) as it regards 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 performer.

5. Composition and performance experiences with the Functional Performer

Since 1988, we use the FP system for music and multimedia live performances. The work of designing and making performances has allowed to set up a working methodology for the analysis and the check 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 precious experimentation' opportunities anyone who would 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, in a little changing universe (i.e. the functional performance), concurrency, the causality relations or even the music events' indetermination.

In conclusion, it's important to point out that in this work, as well as in other L.I.M. works, we propose a substantial mentality change and the refusal 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 between 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)*, has performed 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. 4 we show the melodic fragments' file for the part of FP1; in Fig. 5 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.), has performed 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. 6 we show the melodic fragments' file for the part of FP: it's one melodic theme splitted in 11 segments which is the basic reference for the definition of main structures of composition. In Fig. 7 we show also the respective operative score.

Figure 4: melodic fragments for FP1 in Pro.Fumo (Dante Tanzi - 1988)

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 5: operative score for FP1 in Pro.Fumo (Dante Tanzi - 1988)

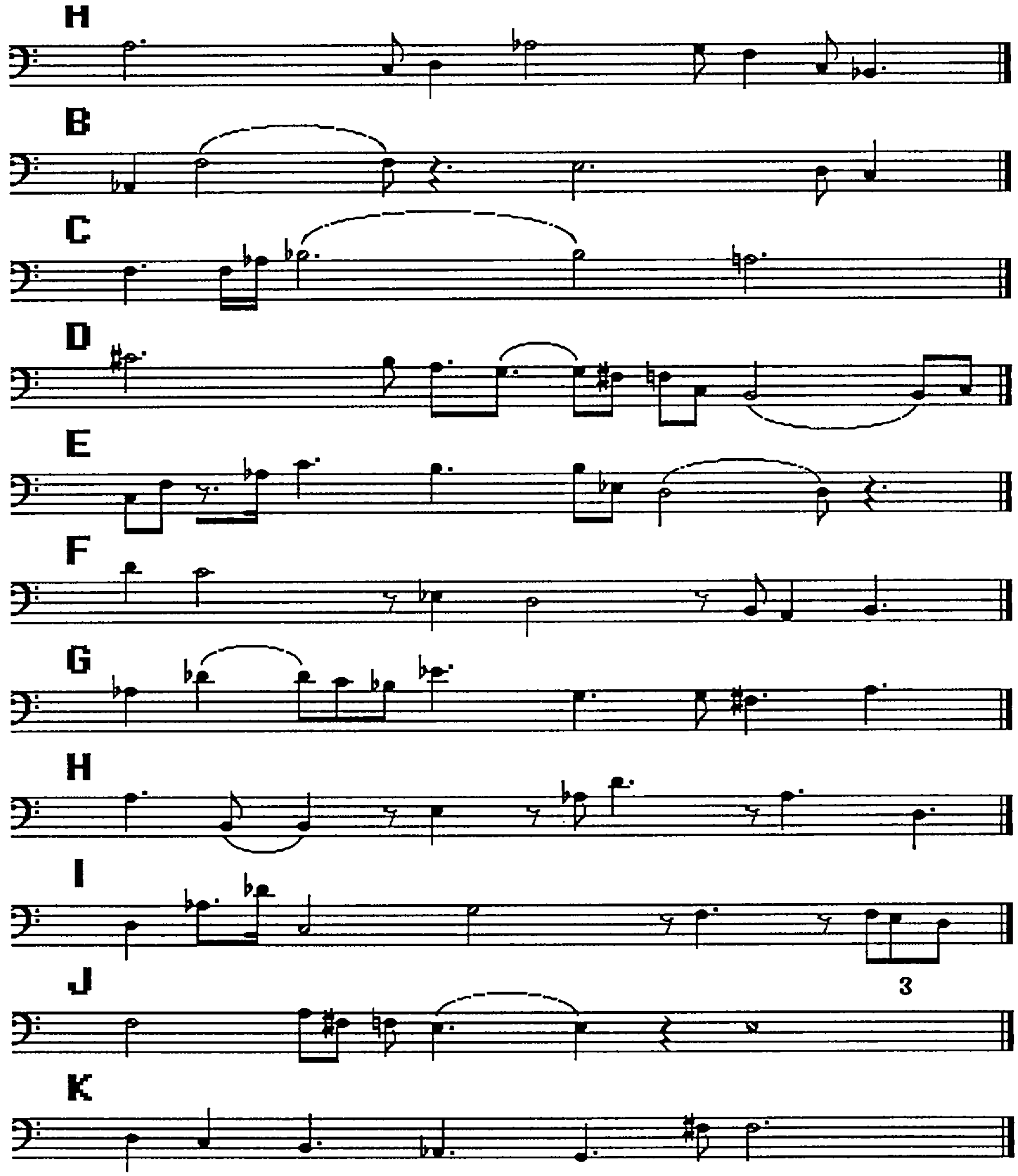


Figure 6: melodic fragments for FP in Louvre (Dante Tanzi - 1989)

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 7: operative score for FP in Louvre (Dante Tanzi - 1989)

Acknowledgements

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

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

6. References

A. Camurri, G. Haus, A. Stiglitz, R. Zaccaria: *"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", C.N.R., Roma, 1989.

G. Degli Antoni, G. Haus: *"Music and Causality"*, Proc. 1982 ICMC, pp.279-296, La Biennale, Venezia, Computer Music Association Publ., San Francisco, 1983.

G. De Poli, G. Haus: *"Ingegneria del software e informatica musicale"*, Proc. Annual Congress A.I.C.A. (1982), pp.415-430, Università di Padova, 1982.

S. De Stefano, G. Haus: *"A Mathematical Approach to the Representation of Musical Structures"*, Proc. 2° International Conference on Data Bases in the Humanities and Social Sciences, 22pp., Facultad de Informatica, Madrid, 1980.

S. De Stefano, G. Haus, A. Stiglitz: *"Descrizione di processi musicali per mezzo di operatori geometrici: un esempio applicativo"*, Proc. 4° Colloquium on Musical Informatics (1981), pp.37-61, CNUCE-CNR, Pisa, 1981.

G. Haus: *"Trasformazione di testi musicali per mezzo di operatori"*, Proc. 3° Colloquium on Musical Informatics (1979), pp.178-193, Università di Padova, Padova, 1979.

G. Haus: *"Elementi di informatica musicale"*, 234 pp., Gruppo Editoriale Jackson, Milano, 1984.

G. Haus: *"Multimedia Performance: Experiences and Discussions"*, (paper submitted to a scientific review), 1990.

D. G. Loy: *"Designing an Operating Environment for a Realtime Performance Processing System"*, Proc. 1985 ICMC, pp. 9-13, Computer Music Association Publ., San Francisco, 1985.

S. T. Pope: *"The development of an intelligent composer's assistant: Interactive graphic tools and knowledge representation for music"*, Proc. 1986 ICMC, pp. 131-144, Computer Music Association Publ., San Francisco, 1986.

A. Stiglitz, D. Tanzi: *"Strumenti software per la performance in tempo reale di Louvre"*, Proc. VIII Colloquio di Informatica Musicale, pp.227-234, AIMI/Spaziomusica Publ., Cagliari, 1989.

D. Tanzi: *"PRO.FUMO"*, Proc. 7° Colloquio di Informatica Musicale (1988), pp.167-168, Roma, AIMI/CIDIM/SIM, 1988.

D. Tanzi: *"Manipolazione di testi musicali con il Functional Performer"*, L.I.M. Tech. Report, Università degli Studi, Milano, 1990.

M. Waisvisz: *"The HANDS: A Set of Remote MIDI Controllers"*, Proc. 1985 ICMC, pp. 313-318, Computer Music Association Publ., San Francisco, 1985.

D. Zicarelli:*"M and Jam Factory"*, Computer Music Journal, 11(4), pp.13-29, Cambridge, MIT Press, 1987.