THE CHANT PROJECT

MODELIZATION AND PRODUCTION, AN ENVIRONMENT FOR COMPOSERS
including THE FORMES LANGUAGE

FOR DESCRIBING AND CONTROLLING SOUND AND MUSICAL PROCESSES

X. RODET, J.-B. BARRIERE, P. COINTE, Y. POTARD

IRCAM, 31 Rue Saint Merri 75004 Paris FRANCE
1. INTRODUCTION

We present the recent developments of the CHANT project, which was set forth at the 1980 and 1981 Computer Music Conferences. As a point of departure let us hear some sound examples synthesised in 1981 with the CHANT program and the help of the environment created around:

- A study of "extreme" voices inspired from Tibetan voices. Components have been introduced in singing voices and they correspond, for non-vocal sounds, to time variations of spectrum and to noise components. We have used them in this example to suggest a certain rhythm. It should be noticed that we use only synthesis by rules from written text, and no analysis-synthesis technique. This example also results from a study of different features which influence timbre: noise components, random variations of spectrum envelope or excitation, role of odd and even harmonics, etc... Quasi-instrumental sounds are also heard, drums and cymbals, which are easily synthesised with the CHANT program.

[sound example]

- In this study, we have focused on the relations between timbre, articulation and "phrase", in one of the most difficult passages of the classical repertoire. This is the aria of the Queen of the Night from Mozart's opera The Magic Flute. The accompaniment has the advantage of a human pianist and a real piano.

[sound example]

- A kind of a model of a "soprano voice" was written for the preceding example. As will be explained, such a model should be as general as possible. Thus, we proposed to this synthetic soprano, as a test, a new serial melody. We insist that our aim is not to imitate natural singing or instruments for themselves; they are only sources of possible structures for developing "models", and tests of our competence.

[sound example]
Let us conclude this introduction with a remark about "representations". To describe how some given example was synthesised, it is not rare for reasons to propose a sort of representation like the following, which, of course is supposed to be very straightforward:

If not, another representation can be "preferred":

[Diagram with labeled parts and connections]
The CHANT project

II. THE CHANT PROJECT

1. Presentation

The CHANT project is a research on analysis, formalisation and synthesis of the laws and rules that take place in musical creations and performances (including sounds, structures and their relations). That knowledge is to be represented by "models".

First let us recall the starting points:

a. Analysis of sound productions and musical performances, particularly voices.

b. The needs encountered by composers and their wishes.

c. Data and problems in the field of perception.

Now, let us describe the aims of this research:

a. A deeper knowledge of the laws and rules of sounds and structures and of their use in musical creations and performances.

b. The development of powerful and "intelligent" tools and concepts for musical synthesis.

c. A joint research of scientists and musicians in domains 1 and 2.

Special interest for voice raised, due to the multiplicity of its productions, richness and complexity, reaching self-completeness, and also because voice is a domain where human perception is particularly focused and skilled even for non-musicians; furthermore, it benefits from the link between language and voice. In actual fact, we consider vocal productions to be the most strongly structured. Voice is considered at any level: sound emission, sound forces and patterns within phonemes sequences, words, paragraphs, etc... All parameters are examined, like rhythm, fundamental frequency, intensity level, spectrum envelope, noise, etc... But of course we are concerned with all other kinds of musical productions and performances from which fertile models can be extripated as will be seen in the following.

2. MODELS

Thus, we intend to build models of the laws and rules that take place in musical creations and performances, i.e. "Models of Knowledge" for various sorts of musical phenomena:

* pre-composition, that is the processing of existing sound material

* composition

* different layers of the synthesis process, from very global control to very precise sound details, all interacting.

In other words, these algorithms have to describe as exhaustively as possible the sound and its evolution, the structure in which it is embedded and the dynamic interaction between sound and structure.

But naturally, it is not these precise models that are of direct interest to composers. Our principal aim is to try to extract general types of models in order to suggest new efficient ones for musical creation, by imitation, modification, concatenation, negation, etc... of already discovered models.
III. THE CHANT PROGRAM FOR SYNTHESIS

1. Presentation

Let us first present the work done with the CHANT program. After being initiated as a research tool, CHANT is now fully used for composition and production at IRCAM. It has proved very successful and efficient within the constraints of musical production for scheduled concerts or research periods, by providing ease of use and better adaptation to the composer's thought. For instance, CHANT is encouraging types of representation closer to the musical imagination than the traditional ones inspired by classical analogical synthesizers or signal descriptions. We think that computer music tools should not be stuck to particular synthesis techniques but should move towards higher level representations, closer to composing activity.

In this direction, we prefer spectral representations and, furthermore, we favour those features of the spectral representation that are perceptually relevant. According to that perspective, a model, that is a set of rules, will describe a correlation between specific parameters, keeping from a production system, those characteristics that are musically efficient.

This way of considering analysis and synthesis has appeared very satisfying in the course of the IRCAM workshops where even total beginners in Computer Music easily obtain complex results in a minimum of time.

2. Sound Examples

Let us hear some examples synthesised with CHANT.

a. Some percussions examples studied by composer J. Tielens.

b. An attempt to force unrelated models to cooperate in a quasi-random way.

c. Extract from J.-B. Barrière’s "Chronicle" from a library of objects among which vowels and consonants, an algorithm selects a sequence of objects in a controlled random way.

d. Composer M. Tabachnik has built his specific world, using CHANT in an non-orthodox way that is additive synthesis. However, CHANT was found convenient and successful for that particular use far away from our ordinary concerns.
Different efforts are now pursued by the CHANT team along which we notice the implementation of CHANT on the 4K machine for real-time application. We are also going to write a "C", "coherent", version of CHANT, to be available for all computer musicians.

IV. MODELS: DEFINITION, PROPERTIES AND DERIVATION

1. Example and derivation

As said before, our purpose is not to limit ourselves to models of natural musical performances, but to derive new models or new types of models from previous ones: a "type-model" being a more general source of knowledge extrapolated from already known models.

An example of such models and types are "consonants", on which part of our effort is presently centered. They represent one of the most difficult processes to model. To our knowledge, we are going to hear the first examples ever done of synthesis by rules of consonants in the singing voice.

**sound example**

EP In this context, a very specific model would be the one for any occurrence of, let us say, the syllable "KA".

EP From models of "KA", "KG", "KI", etc... we can derive a more general model for the consonant "K".

EP From models for "K", "P", "S", etc... we can derive a model for a large set of consonants.

EP From models of consonants, attacks of instrumental sounds, legatos and staccatos, we can derive a model for any of these transitional sounds.

Precisely, such a model is a set of rules that applies to any specific context, for example, on any sustained sound, in order to form it into the desired transition. It will be shown how this is implemented in objects and rules in the FARMES language. A model is represented as a hierarchy of objects reflecting a musical structure.

2. Properties of MODELS

Noticeable properties of such models include:

1. _Generality:_ apart from a specific application a model is intended to be as general a representation of a process as possible, and not a portrait of a particular sound or note: the model of a crescendo or of an attack pattern should apply to as many different sounds as possible.

2. _Universality:_ a model should be independent of a particular synthesis technique and should refer to universal concepts as found for example in acoustics or psycho-acoustics.

3. _Compatibility:_ models should apply in any context in which they are placed: in any combination they should gracefully cooperate and interact in the universe created by the composer.
Finally, one of our goals is to build a kind of "living memory", an environment in perpetual evolution where, however, each contribution is constitutive, that is, kept and classified so that it becomes available for users concerned with some areas of interest or for further developments in other contexts.

3. Other requirements

Although these models may be rather complex, one should be allowed to use any number of them, written by anyone, in any order, simply by invoking them, without having to write a new subroutine.

Models can be modified very easily. One should especially be able to invoke a model by its symbolic name, with the temporary and local modifications simply attached to the name.

New models are easy to create from old ones from analysis data or from algorithms proposed by composers.

Models can be easily organized into a composition, eventually very complex, which supposes that they can be dynamically invoked and modified, even by other models.

We should place into the program as much "knowledge" as possible, i.e. the kind of knowledge about which almost anyone would agree. Of course, it should be easy to nullify any of this knowledge.

Models will be all the simpler as following common human communication conventions and presuppositions.

The majority of the information that is not supplied by the user receives a default value. Effort is done for simplicity of program text.

The composer should be relieved of all the tasks that the machine can do equally well.

When possible, natural language constructs, otherwise simple and clear symbolism, should be used.

Integration of models should be direct and easy.
U. THE FORMES REPRODUCTION ENVIRONMENT

1. Presentation

The complexity of the processes encountered in musical synthesis has
considerably the authors on the way of thinking and realizing a
programming environment of higher level than are purely synthesis programs,
(like Music-V, Music-10, Chant, etc...), to deal with control structures and
forms.

More precisely, FORMES includes a programming language for manipulating
"objects" which represent "models of knowledge" (as previously defined) of a
certain sound production or sound phenomenon. This new interactive
object-oriented language is intended for describing all dimensions or
characteristics of a sound process temporal evolution.

These evolutions are described within objects by "rules". The structure
of the sound process is reflected into a hierarchy of objects considered as
building blocks and the rules implement desired correlations between
specific parameters.

For instance, this evolution can be parameterized at each quantum by
updating and evolution of a computing tree feeding periodically the inputs
of a synthesizer by a set of new values. In other words, the set of
evolutions can be "stored" in a certain number of models. Each of these
models is generally representative of one characteristic of the sound
production and/or of its perception.

These models will be represented in FORMES by "rules" (or actions)
embedded into "objects". It is thus clear that, while sometimes very
complex, these objects or models must be compatible, that is suitable for
simultaneous use in any kind of order. Furthermore, it is not a given
representation of a model within an object which is important, but the
precise behavior or function that it fulfills: this means that an object
can be replaced by anyone having the same behavior, independently of the
implementation, so that the composer is free to choose whatever type of
representation he prefers at any point.

The creation of an object consists in defining a totally new model, or
more simply in modifying a new one by modifying a previously existing form.
When an object becomes "active" the model it represents is applied. An
object can contain sub-objects ("children") and can be embedded into a
super-object ("father"). So that the activation of an object will be made
simply either at the top level by typing its name, or indirectly by its
class position in a hierarchy, as a sub-object of an object executed at the top
level. It is then very simple to test the elementary objects independently
from others before grouping them in higher level objects.

In order to reflect, as seen previously, the invariance of a perceptual
effect, the precise contribution of an object to the sound process depends
upon the context in which it is activated. In order to reflect, as seen
previously, the invariance of a perceptual effect, this is one of the
important ideas focused on in FORMES: the concepts of "intrinsic semantic"
and "context sensitivity" of objects. These features are essential for
fulfilling two fundamental requirements:

- compatibility between objects, itself intimately connected with

- organization of knowledge by creating and deriving nodes,
progressively building a large library of all sources of musical
knowledge

405
Let us now unite in FGIES a very simple example of a piece, consisting of a sequence of notes.

```
(cred SEQUENCE
    rules:  ( (GECA) (GECO) )
    sub-objects: ( new NOTE A4 duration quaver)
                   (new NOTE C3 duration half-note attack-strength strong)
)
```

CRSDB is the construct which creates an object, and GECA and GECO are functions describing some global evolution of attack and decay characteristics during the sequence. A more concise but less clear text could be obtained through micro-definitions. Notice that SEQUENCES make no assumption about the nature of the objects NOTE except what they have duration and attack-strength.

In this sequence NOTE is an object already defined. Suppose that its attack and decay are specified as modifications of a sustained sound called SOUNDBODY.

```
(cred NOTE
    one-time:  ( (my-RS = (pitch (self\'s name))) )
    rules:    ( (10 = my-RS)
               (amplitude = (combination my-amplitude amplitude)) )
    sub-objects: (ATTACK SOUND-BODY DECAI)
                  (my-RS 100 my-amplitude 1)
)
```

Remark that NOTE can be used in any "father" object different from SEQUENCE and that NOTE makes no assumption upon the nature of ATTACK, SOUND-BODY and DECAI.
Suppose that we simply describe SOUND-BODY in terms of its spectral envelope, by the location of the maximum of this envelope and the richness of the sound in high frequencies, designated by two variables "Fmax" and "richness":

\[
\begin{align*}
&\text{rules:} \\
&\quad \text{If } (\text{Fmax} = 1000) \quad \text{richness} = \text{low}) \\
&\end{align*}
\]

"low" is a symbol that receives a value convenient for a low richness, let us say: (low = 0.00) and (high = 10.). Similarly we could define: (quaver = 1.) and (half-quaver = 0.5) etc...

Suppose ATTACK is simply a modification of the spectrum and fundamental frequency:

\[
\begin{align*}
&\text{rules:} \\
&\quad \text{If } (\text{ramp f0 (start f0 f0)} ) \\
&\quad \text{ramp amplitude 0 amplitude}) \\
&\quad \text{envi (duration function attack-strength))} \\
&\end{align*}
\]

The best DECAY would be defined similarly.

Now, how do we apply these models? When started, FORMES asks:

\textbf{What do you want to sing?}

If the answer is:

* \textbf{SOUND-BODY} \textbf{E} we get a sustained sound

* \textbf{NOTE} \textbf{E} we get one note

* \textbf{SEQUENCE} \textbf{E} we get the desired sequence.

Then, if satisfied, we can start from such sequences as building blocks for writing a longer piece.
3. Conclusion

The programming environment FORMATS includes different tools like:
- Multi-windowing output
- Graphic display of parameter values
- On-line documentation
- A collection of sound analysis and display programs
- Data base of predefined objects and rules (in study)

We would like to end this presentation with a sound example of the work done with composer Harrison Birtwistle. He was looking for the "voice of God" and "birth of the language". In our library of sounds issued from continuous research, he found some models close to his wishes. He was going to hear the derivation of these models towards the precise effect he wanted:

[Sound example]