Composition Processes

1

By musical composition we generally understand the production of an instrumental score or a tape of electronic music. However, we also understand composition as the result of composing: the scores of instrumental or electronic pieces, an electronic tape, even a performance (we say for instance: "I have heard a composition by composer X"). The concept of composition is accordingly closed with regard to the result, but open with regard to the making of a composition; it tells us nothing about preparatory work, whether it is essential for the composition or not. Preparatory work includes the choice of instruments and values for dynamics or durations, but it also includes the definition of sounds in electronic music and can even be extended to cover the invention of special graphic symbols. Electronic sounds or graphic symbols are not always additions to composition; they are often "composed" themselves, i.e. put together according to aspects which are valid for actual composing.

These considerations give rise to the following questions:
what do we mean by "composing"?
do we mean the composition of musical language structures?
do we mean the composition of sound structures?
do we mean the composition of single sound?

To begin with the last one: can we call a single sound, especially in electronic music, a "composition" or at least the result of composing? In the early days of electronic music the Cologne studio stressed the fact that not just a work but each of its individual sounds had to be "composed"; by this they meant a way of working in which the form of a piece and the form of its sounds should be connected: the proportions of the piece should be reflected as it were in the proportions of the individual sounds. It is better to call a list of sound data having no direct connection with the structure of a piece a description of the sounds. In terms of Cologne aesthetics it is then perfectly possible to talk about the composition of single sounds, but this brings us to the next question as to what a single sound is. The term comes from instrumental music, where it is most closely involved with questions of performance and notation technique. To give a tentative and rough description of the single sound, it is characterized by an unmistakable start ("entry") and an unmistakable end and consequently by an unmistakable duration, furthermore by uniform pitch, loudness and timbre. We can specify this rough description in more details by the following remarks:

- timbre changes in the single sound play such a slight part as to be negligible here,
- changes of loudness in the single sound (crescendo, decrescendo, tremolo) generally belong to performance or expressive characteristics, the above definition (start, end, duration and pitch) being unaffected; sounds starting "inaudibly" pp or "dying away" to pp are exceptions which are justified by the general redundance of the context,
- pitch-changes in the single sound (glissando) restrict the above definition more closely; we might however take into account the fact that glissandi frequently occur as mere transitions between stationary sounds (especially for singers and string-players), and that independent glissandi contradicting harmonic unambiguity, form, like pitchless percussion sounds or clusters, a category of their own in which the conditions of beginning, end and duration are still valid.

As this definition shows, we can only speak of single sounds in instrumental music really (and even then only within limits) and in the first phase of electronic music, which was closely linked with instrumental traditions. In the ensuing period of electronic music it is better to speak, instead of the composition of single sounds, of the composition of sound-events or even sound-fields, since a sound-event is not only assembled from pitches, degrees of loudness and durations, but includes to an increasing extent transformations as well: the uniform composition of an event frequently results in an auditory impression whose variability contradicts the definition of individual sound in various parameters; the beginning, end and duration are all that are left of the definition. These quantities also describe the entire work, though, which might consequently be seen as a single, complexly modulated sound. As we see, considerable difficulties can be involved in making a distinction between a composition and its individual sounds, so that we can only answer the question as to whether we should understand composition as the composition of single sounds in the affirmative when there is a continuous structural connection between the overall form and its parts, right down to the physical sound data, and only then when – in the sense of instrumental tradition – the whole can be heard to consist of individual parts.

Detailed discussion of the single sound has shown that it is only covered by the term composition to a limited extent. The composition of sound structures seems to fit into our subject more appropriately. This is because in sound structures physical sound data and musical structural quantities meet. The sound structure is not tied to the narrow definition of the individual sound, but may, as we have seen, consist in the auditory impression of single sounds. According to its definition the sound structure is more complex and usually longer than a single sound, thus more closely approaching a form-section, virtually the whole work. Nonetheless the sound structure can also be said to cover a partial aspect of composition: it would either have to be described as a more complex, assembled single sound or as an unfinished piece. However, the technical circumstances of working in an electronic studio or with a computer often lead to composing in sections; problems of sound structure can therefore be treated just as well under the musical structures of entire pieces.

This brings us to the last of the questions posed before: by composition processes do we mean the composition of musical language structures? Emphatically, yes. Composing terminates in pieces, and the extent to which pieces are put together from sounds, and the relations prevailing among these sounds, are a matter of how a composer works. Composition is the application of a grammar which generates the structures of a piece, whether the composer is aware of an explicit grammar or not. The sound elements (I leave the question open as to whether these are single sounds or sound-events) to be composed into structures do not have to be in an unambiguous relationship either to one another or to the structures; assembly – "composing" – always takes place when something big consists of smaller parts. In more simplified terms, then, we can say that composition refers to elements which need not themselves be the subject of composition; the consideration of composition processes can disregard questions of sound production; sound production is not interesting as a composition process until it becomes integral, i.e. until the structure-generating grammar refers to sound data instead of to given sound elements.

2

We are faced with a distinction between structure and sound as soon as a composer not only writes a score but makes a sonic realization of it as well. This occurred for the first time in electronic music when not only single sounds but entire sound structures could be produced, particularly with the aid of voltage control. The compositional rules for giving form to the individual events as well as for connecting them in time were notated in wiring diagrams which could be reproduced to a certain extent in the form of studio patches. Not until digital computers were used did it become possible however to execute compositional rules of any desired degree of complexity, limited only by the capacity of the computer. Automatic realization of entire electronic pieces or at least of lengthy sections using voltage control systems seems to be the exception, though, and in the field of computer music much more attention appears to be paid to problems of sound production than to those of composition.

In his article "A Composer's Introduction to Computer Music"[1], William Buxton makes a distinction between 'composing programs' and 'computer-aided composition'. As examples for composing programs the article refers to Hiller's *ILLIAC Suite*, Xenakis' *ST* programs and my own programs, *Project One* and *Project Two*.[2] Though this list is not complete, it is conspicuous

[1] Buxton, W., "A Composer's Introduction to Computer Music", *Interface* 6,2, Amsterdam and Lisse, 1977.

[2] a) Hiller, L., Isaacson, L., "Musical Composition with a High-Speed Digital Computer", *J.A.E.S.* 6,3 (1958).
b) Hiller, L., Isaacson, L., *Experimental Music*, New York, McGraw-Hill, 1959.
c) Hiller, L., "Computer Music", *Scientific American* 201,6, (1959).

for its brevity. A reason for this might be the practical impossibility of describing the composition process entirely in the form of computer programs. Although a composer runs through more or less fixed sequences of decisions determining his personal style, and also employs consciously chosen rules limiting the freedom of his decisions in the individual case, he is still, whether he is aware of this or not, under the impression of a musical tradition which values a composer's originality more highly than his skill in using established patterns. A composer is more accustomed to being influenced by a spontaneous idea than by prepared plans; he decides and discards, writes down and corrects, remembers and forgets, works towards a goal, replaces it during his work by another – guided by criteria which are more likely to be found in psychology than in music theory. This is why computers are more likely to be used for purposes of composition: (1) to solve parts of problems or to compose shorter formal sections instead of complete pieces, (2) to tryout models greatly simplifying compositional reality and supplying the composer with a basic scheme which he can elaborate as he feels best, (3) to compose an individual piece for which the composer writes a special program more resembling a score than a solution for a number of problems.

In the chapter on computer-aided composition, Buxton refers to the *SCORE* program, *MUSICOMP*, the *GROOVE* system and the *POD* programs, among others.[3] This list demonstrates how difficult it is to separate the actual composition of a piece of music from auxiliary actions which are partly predominant or subordinate in the composition, or which partly overlap. Here we are faced by the issue of whether we are going to understand composing as the entire process from planning via writing the score (or producing a tape) right up to performance, or merely as the intellectual act of invention. If we limit ourselves to the intellectual act of invention, we speak of "composing programs", of musical grammar, of a score as a document of intellectual activity, inspiration, creative powers. If on the other hand we envisage the entire process, it can be divided into a number of single activities which can be performed by different agents: composers, musicians, generators, computers, not to forget the listeners. These auxiliary services include, as far as we are dealing with computers: (1) the sonic realization of previously fixed score data, (2) the processing of parts of problems using libraries of subprograms, (3) the production of graphic scores or musical graphics, (4) sound production based on simple compositional rules, so that, say, the sound models from a sound library are assembled to form sound structures. The computer performs various services in these examples: in the sonic realization of score data it replaces an electronic studio or an orchestra, whilst not being responsible for the score; dealing with parts of problems with the help of a subprogram library can be expanded to become a complete description of the act of composing; the production of graphic scores replaces the copyist, a musical graphic leaves the completion of a piece to – more or less – improvizing players; sound production according to simple compositional rules has the character of a model both with regard to the sounds and to the combinatorial methods – the same group of sounds can be subjected to different combinations or different groups of sounds can be arranged similarly.

There are advantages and drawbacks to distinguishing composing programs and computer-aided composition. An advantage is that in the case of composing programs the computer is expected to supply the compositional work, whilst in computer-aided composition the responsibility is entirely the composer's. A consideration of composition processes might

d) Xenakis, I., *Formalized Music*, Bloomington, Indiana University Press, 1971.

e) Koenig, G.M., "Project 1", *Electronic Music Reports* 2, Utrecht, Institute of Sonology (1970).

f) Koenig, G.M., "Projekt 2 – A Programme for Musical Composition", *Electronic Music Reports* 3, Utrecht, Institute of Sonology (1970).

[3] a) Smith, L., "SCORE – A Musician's Approach to Computer Music", *J.A.E.S.* 20,1 (1972).

b) Tanner, P., "MUSICOMP, an Experimental Aid for the Composition and Production of Music", *ERB-869*, Ottawa, N.R.C. Radio and Electrical Engineering Division, 1972.

c) Mathews, M., Moore, F., "GROOVE – A Program to Compose, Store, and Edit Functions of Time", *Comm.* ACM 13 (Dec. 197).

d) Truax, B., "The Computer Composition – Sound Synthesis Programs POD4, POD5, & POD6", *Sonological Reports* 2, Utrecht, Institute of Sonology (1973).

e) Buxton, W., *Manual for the POD Programs*, Utrecht, Institute of Sonology, 1977.

therefore be limited to composing programs. A drawback is that the composition process consists of activities which cannot be separated into main and auxiliary activities so easily; even in composing programs the composer is still chiefly responsible because he must at least prepare the data on which the composition process basically depends, if he does not in fact also write the program himself. – In what follows I shall limit myself mainly to the invention of music in the form of representative models, without going into the distinction between main and auxiliary actions.

3

We have been occupied with programmed music at the Institute of Sonology at Utrecht University since 1965; by programmed music we mean the establishment and implementation of systems of rules or grammars, briefly: of programs, independent of the agent setting up or using the programs, independent too of sound sources. This means that programmed music covers: (1) instrumental scores which a composer writes at his desk on the basis of binding compositional rules, which do not fundamentally differ from computer programs, (2) electronic compositions which like the said instrumental scores are systematically composed, then to be "mechanically", i.e. without additions and cuts, realized on studio apparatus, (3) electronic works produced automatically by the use of studio patches, (4) instrumental scores based on computer programs, (5) tapes based on sound data which were calculated and converted by a computer program. In this field of programmed music, instrumental and electronic pieces have been realized with and without the computer, and for some years our lecture schedule has included a series of lectures with this title alongside the subject of computer sound synthesis which does of course occasionally overlap the first one. The experience we have gained during the years and which I assume is fairly similar to experience gained elsewhere can be summarized as follows:

The composing process

Opinions differ as to what a composing process is, there being all gradations between constructive and intuitive composers. Investigations in the field of programmed music can only be expected from composers who already have highly constructive inclinations or previous knowledge or, although keener on free expression, want to discover a new realm of experience. Among composers with constructive inclinations one often observes a tendency towards processes which to a fair extent exclude compositional decisions made in advance, i.e. the input of structure-conditioning data. They prefer to choose what corresponds to their personal taste from among the automatically produced results. If, for instance, there is a choice between two composing programs with differing input formats, the program with the smaller input format is more likely to be chosen. It is often only taken as a model for a composer's own program which gets by with even fewer data. Syntactic rules are replaced as far as possible by random decisions.

The other extreme occurs too, but rarely. Here composers mistrust the automaton or chance, and try to control the process down to the last detail. This leads to programs with large input formats, with a detailed dialogue between composer and computer, and to the composition and frequent correction of smaller and smallest form-sections. We again observe here the smooth transition from composing programs to computer-aided composition. – I am speaking here, incidentally, of tendencies observed in composers working at our institute, who after becoming acquainted with existing programs go their own way and contrive their own composing systems, occasionally taking years over them. There is no doubt as to the popularity which systems for computer-aided composition generally enjoy, but this is not covered by my subject.

The fundamental difficulty in developing composing programs is indubitably in determining the dividing-line between the automatic process and the dynamic influence exerted by the composer by means of input data and dialogue. To put it briefly: when there are few data and little dialogue, automata are expected to produce the composition; when there are a lot of data and dialogue, the responsibility remains the composer's, the computer being degraded to an intelligent typewriter. The dividing-line between composer and automaton should run in such a fashion as to provide the highest degree of insight into musical-syntactic contexts in the form of the program, it being up to the composer to take up the thread – metaphorically speaking – where the program was forced to drop it, in other words: to make the missing decisions which it was not possible to formalize. The development of composing programs consists of pushing this dividing-line further and further away. Whoever goes along with this personal opinion of mine will realize the difficulties involved in this approach. For the attempt to formalize is not only oriented towards a medium – music – which, as opposed to natural language tends to unravel rather than to fix; every composer will moreover imagine the dividing-line to be in other areas, depending on his stylistic criteria and expressive requirements. The computer program with which a composer is confronted may pose him a puzzle instead of solving it.

Compositional rules

In the search for compositional rules for making composing programs, there are three main avenues.

The first one leads to the analysis of existing music of the past and present. The premise here is that the rules, or at least the regularities in a composer's output or in a stylistic period can be discovered if one examines the scores closely enough. Regardless of the use of such analyses for musicological research, say for purposes of comparing styles or classifying anonymous scores, the question remains as to whether it supplies the required indications for the synthesis of music. Analysis and synthesis do not cover each other perfectly enough for the results of analysis, if used productively, to lead back to significant music; analysis proceeds from questions which are not necessarily those of the composer; after all, in order to arrive at statements of sufficient generality, not only must very complex questions be formulated, but a vast number of scores be subjected to such an analysis. The historical line of sight would at the same time be unhistorical, because it would ignore historical development and measure works from different periods by the same standards. One might also ask whether a composer who wants to create something new can benefit from frozen models from the past. In all this we must of course not overlook the fact that knowledge of compositional means as developed during the past centuries and exerting influence right up to the most advanced composing, is an absolute prerequisite.

The second way leads to introspection. The composer/programmer analyses his own experiences, he investigates whether and to what extent his way of composing is formed by habits which can be formulated as rules. He refers to the music of his predecessors from whom he learnt his craft, going as far back in history for this as he wants. Systematic analysis is replaced by intuition, at least as far as he discovers rule-like aspects in his models and renders them fertile for his own work. An exchange of ideas with colleagues, even with pupils, can be significant because it results in a higher degree of generality. Introspection has the drawback of being less objective and hence less generally valid; on the other hand introspection has the advantage of proceeding less from analytical, but more from synthetic problems; it is aimed more directly at matters of compositional craft and therefore does more justice to reality. Introspection also presents an opportunity for describing compositional ideals instead of limiting them to the recapitulation of what has gone before.

The third way leads to the description of models. As long as we do not know how musical language functions, nor see how to derive its grammar from everything composed up to now, we can still assemble fragmentary knowledge and assumptions to form a model which can tentatively represent reality. Models describe partial aspects, and the results achieved with their help can only be compared with partial aspects of the reality they are substituting. The systematic approach which makes the first way of analysis, striving towards completeness, seem attractive, appears here again in the form of methodical experiments. Repeated application of a model under changed circumstances makes its limits clearer; accumulation and correlation of the results cause the model to reveal itself and at the same time the extent to which it coincides with a part of musical reality. In this it may even transcend the reality experienced up to that point by exposing contexts which had escaped analytical scrutiny. The analytical task – given the music, find the rules – is reversed: given the rules, find the music.

Compositional results

Another issue closely linked with that of rules is the compositional result achieved with composing programs. Rules abstracted from music by means of analysis, introspection or model construction result primarily in the acoustic (or graphic) equivalent of this abstraction; the relation to music has to be created again. This, too, can be done in different ways. I expressly mention this retranslation because it should be already kept in mind when a composing program is being designed. There are various ways of doing this too; I shall deal with three of them here.

One possible evaluation is the comparison with precedents which are to be imitated by means of the program or suitable input data. This particularly applies to programs written on the basis of extensive analysis of existing music. Apart from the trivial question as to whether the program is carrying out the given rules correctly so that the composed result contains the desired quantities in the desired combinations, it would be good to examine whether, when listening to the results, there is an aesthetic experience comparable to the precedent. I use this vague term, aesthetic experience, to designate the quality which distinguishes, say, a written score from its performance, or a composer's material from the constellations in which it eventually appears in his music.

Another evaluation refers to the expectations of the writer or user of the program. Especially in the case of the already mentioned introspection, this does not involve reviewing existing aesthetic products, but in a way looking forwards for ideals to inspire a composer in his work. The result of such an evaluation depends on goals which do not refer to precedents with which they can be compared. This evaluation is consequently less communicable than the first one involving the formalizable comparison of original and copy.

A third evaluation is to look for musically experienceable references in a context produced by means of models. Since the model merely marks out the framework within which aesthetically communicable contexts are presumed to be, the results produced by means of the model cannot be compared with precedents, nor with ideals. They appeal rather to the evaluator's capacity for discovering what is special in what is general, i.e. the accumulation of what is significant in surroundings whose significance is at the most latent. As already explained, it is up to methodical experimentation to determine the validity-range of the model – and thus the probability of accumulating aesthetically communicable constellations; one must not forget that the models only describe basic structures which will need detailed elaboration in the form of a score or tape. – What I have been saying here obviously also applies to the evaluation of precedents and expectations.

Compositional methods

I shall now turn to some compositional methods which are due to introspection or which might be useful in constructing models, but which in any case represent generalizations of the concrete process of composing. Note, though, that they remain within the range of experience of the composer writing, or just using, a computer program.

Interpolation might be a good name for a method which so to speak pushes forwards from the outer limits of the total form into the inner areas; applied to the dimension of time this would mean: dividing the total duration into sections, the sections into groups, the groups into subgroups and so on, until the durations of the individual sounds can be established. We could apply this method accordingly to other dimensions too, by speaking of aspects, partial aspects, variants and modifications.

By contrast, extrapolation would proceed from the interior towards the outside: from the individual sound to the group of sounds, thence to the super-groups, via sections to the total form. Both methods are concentric; the formal shells which so to speak enclose the nucleus of the form exist in ideal simultaneity; the form is not unfolded teleologically but rather pedagogically, the details being presented in such a way that the relation of the detail to the whole is always quite clear to the listener.

As opposed to these two methods of interpolation and extrapolation there is a third which I should like to call chronological- associative. The composing process unfolds along the time-axis, thus being put in the position of the ideal listener. Note that in this way every event is given its irremovable place in time, whereas in the previous examples of interpolation and extrapolation the events were interchangeable.

A combination of methods more oriented towards time or space can be found in the composition of blocks; by a block I mean a part of a structure which requires complementing by other blocks but which is still complete in itself. It is easier to state rules for blocks than for entire pieces, because they are of shorter duration and do not have to meet the demands made on pieces.

Individual blocks can be produced by means of interpolation, extrapolation or the chronological-associative method; their order is determined by the composer, i.e. outside the scope of the formalization in the program.

The chronological-associative method can finally be extended to the teleological or goal-oriented method by means of feedback. Here the composer supplements individual data and syntactic rules describing only local strategy by objectives with which local events are continually compared. This type of method seems to approach most closely the real process of composition, but it also involves the greatest difficulties of representation in program structures.

Practical aspects

To close this section, I shall talk about a few practical aspects of writing composing programs, their accessibility and the forms of data output.

The writer of a composing program must first of all clearly define the points of departure and goals. Points of departure are chiefly in the relation between computer and composer, i.e. between the musical knowledge stored in the program and the input data the composer uses to manipulate this knowledge. Goals are related to the extent and kind of the expected results. The definition of the compositional method is also important; for instance, rules, probability matrices, weighting factors, chance etc. have to be taken into account.

The accessibility of composing programs is primarily a question of the available computer system: how much computer time can be given to users, either in the single-user or time-sharing mode; furthermore it is a question of program construction: whether input data must be read in or whether the composer in the course of a dialogue with the computer can continually influence the program; accessibility also depends on turn-around time, i.e. on how long it takes for the composer to receive the output; it is finally a question of the program language if only a subprogram library is available and the composer has to write his own main program.

Data is usually output in the form of tables, musical graphics or sounds. Tables sometimes need to be laboriously transcribed into musical notation, musical graphics are restricted to standard notation; it is very practical to have a sound-output of a composed text giving the composer a first impression in the three parameters of pitch, loudness and duration, before he decides to have tables printed or musical graphics executed. Things are different with systems which do not produce a score but only a sound result; the above-mentioned criteria of accessibility play an important part here.

4

To round off this paper on composition processes I shall deal in more detail with a few programs developed or in the process of being developed at the Institute of Sonology. I shall classify them as composing programs for language structures (instrumental music), sound-generating programs in the standard approach, sound-generating programs in the non-standard approach, program-generating systems based on grammars.

Composing programs for language structures

From 1964 to 1966 I wrote a composing program myself. I regarded it as a first attempt, and therefore called it *Project One*, abbreviated in *PR1*.[4] It has had a lively history: after its first version in FORTRAN II, tried out on an IBM 7090, I made an ALGOL 60 version for an ELECTROLOGICA X8 computer at the computer department of Utrecht University. When the Institute of Sonology acquired its own computer in 1971, I made a FORTRAN IV version of the program for our PDP-15. Still a few years later, after six VOSIM generators had been built according to Kaegi's model, I gave *Project One* a sound output making it possible for a composition to be heard in the three dimensions of time, pitch and loudness.

I had the idea of collating my experience with programmed music at the desk and in the electronic studio to form a model which would be able to produce a large number of variants of itself almost fully automatically. Faithful to the fundamentals of the nineteen-fifties, all the

[4] See remark 2e.

parameters involved were supposed to have at least one common characteristic; for this I chose the pair of terms, "regular/irregular". "Regular" means here that a selected parameter value is frequently repeated: this results in groups with similar rhythms, octave registers or loudness, similar harmonic structure or similar sonorities. The duration of such groups is different in all parameters, resulting in overlappings. – "Irregularity" means that a selected parameter value cannot be repeated until all or at least many values of this parameter have had a turn. The choice of parameter values and group quantities was left to chance, as was the question of the place a given parameter should occupy in the range between regularity and irregularity. A composer using this program only has to fix metronome tempi, rhythmic values and the length of the composition, in other words: he only decides on the time framework of the result, and this only roughly, because all details are generated by the automatism of the program.

Experiments on a large scale were not made with this program, however, because it isalaborious and time-consuming business to transcribe the tables printed by the computer into notation; the turn-around time was too long as well, as long as the program was still running at the University computer centre. Not until the program was installed on our own computer was it possible for a composer to produce several variants in a similar length of time and compare them; but by then I had already written *Project Two*, which I shall discuss presently, and which allows the composer to have more influence on the composition process. *Project One* therefore gathered dust until – about a year ago – we could build six VOSIM generators which play composers their results in real time. The large number of experiments which could be made in a short time revealed a second "dividing line"; the first one, mentioned earlier, separates the independent achievements of the program from the composer's possibilities of influencing it. The second one, discovered with the sound results of *Project One*, separates the significance of random decisions in the micro and macro range of the form. In my first design of *PR1* I was led by the idea that since the details of the form already depend on chance, although within certain limits, the overall form can be subjected to chance too as long as care is taken that the various aspects of the form described by the program are really given the opportunity. More recent experiments with the program have shown, however, the extent to which the rules for the overall form affect the course of the form in detail; it appears that the comprehension of a listener horizontally combining single tones to groups of tones, groups of tones to larger units, and vertically observing the changing structure of different durations, pitches and loudnesses and connecting these data in an as yet uninvestigated manner with impressions, changing in time, of musical states, is increased when either the details of such a state or the states themselves arouse expectations which can be fulfilled in the further course of developments. In order to observe this phenomenon more closely, I first created a possibility of revising, by means of user data, the random decisions for the overall form which had been made by the program. The results are encouraging and indicate ways of defining more precisely the said "dividing line" between the micro and the macro-form. The reverse method is in preparation too, by which the user will have greater influence on the construction of the detail.

This work on *Project One* took up the time which I had really intended to spend on a new version of *Project Two*. Since this second composing program has already been exhaustively documented in the "Electronic Music Reports", published by our Institute, I can make do with a brief summary here.[5]– As I have already mentioned, *PR2* was written shortly after I had finished *PR1*, about 1966 to 1968. Up to now it has run in an ALGOL 60 version at the University computer department, and therefore suffered from the same problems as the first project. I am at present translating the ALGOL program into a FORTRAN version which will be able to run on our own PDP-15 and be given a VOSIM sound output. I am also planning an extended and improved version.

Two properties chiefly characterize "Project Two". On the one hand the user is expected to supply a lot of input data not only defining the value-ranges in eight parameters but also making the parameters interdependent; on the other hand the individual decisions within the form-sections are not made to depend on chance, as in *PR1*, but on selection mechanisms specified by the composer. *PR2*, like *PR1*, realizes the idea of a form-model which can be tested in any

[5] See remark 2f.

number of variants. Both programs, *PR2* more so than *PR1*, require careful preliminary work from the composer, since they are not interactive. The planned new version of *PR2* will however make it possible for the computer and composer to hold long dialogues.

Sound programs in the standard approach

The question as to composition processes inevitably leads to that of the construction of the sounds in composed structures, inasfar as the latter did not precede the former. In sound-generating computer programs we distinguish, as proposed by Holtzman[6], between the "standard" and the "non-standard" approach. To quote Holtzman: "Standard approaches are characterized by an implementation process where, given a description of the sound in terms of some acoustic model, machine instructions are ordered in such a way so as to simulate the sound described; the non-standard approach, given a set of instructions, relates them one to another in terms of a system which makes no reference to some super-ordinated model, [...] and the relationships formed are *themselves* the description of the sound. Standard systems are seen as more or less 'top-down' systems where the synthesis technique is conceived of as manipulated in terms of a given acoustic model. In digital synthesis, programs developed by M. Mathews, i.e. *Music IV-V*, exemplify the standard approach to sound synthesis and form the basis of other major synthesis programs, e.g. Vercoe's *Music 360*, Howe's *Music 4BF*, etc.".[7] The VOSIM sound output to *PR1* and *PR2* also belong to standard systems, and so do programs for a digital hardware Fourier generator, two digital hardware frequency modulation generators (after Chowning's model)[8] and Kaegi's MIDIM system. This list should also include Truax' *POD5* and *POD6*[9] which there is not enough time to examine more closely here. I am also skipping the Fourier generator and a program written by William Matthews for using FM generators. The said programs are chiefly for pure sound production and will not be able to take part in the composition of language structures until they are embodied in suitable composing programs. Unfortunately I cannot say very much about Kaegi's *MIDIM* program, since at the time of writing this paper his manual was not yet available. It is based – as a sound-generating program – on the VOSIM system[10] for the minimal description of speech sounds, which has since been expanded to apply to instrumental sounds. At the same time, however, it is a transition to composing systems, only that here one proceeds from the sound to the composition instead of the other way round – as far as I know, this is a unique case. This transition is caused by having a library of instrument definitions continuously compared with the structure-generating grammar.

Sound programs in the non-standard approach

Among "non-standard" systems, as produced at the Institute of Sonology, Paul Berg's *Pil*e and my own *SSP* can be named. (Kees van Prooijen's *CYCLE* program is so similar to *PILE* that it is sufficient to mention it.) To clarify these I quote another passage from Holtzman's article: "... Samples are related *only* one to another, the relationships created determining the timbre, frequency, etc.; related *only* one to another suggests that the relationships are diacritically defined and do not refer to some super-ordinate model or function. For example, given a set of possible relationships that may exist between samples in a digital computer, one considers the

[6] Holtzman, S.R., "A Description of an Automated Digital Sound Synthesis Instrument", unpublished manuscript, April 1978.

[7] a) Mathews, M., *The Technology of Computer Music*, Cambridge, M.I.T. Press, 1969.
b) Vercoe, B., "The MUSIC 360 Language for Sound Synthesis", *American Society of University Composers Proceedings* 6 (1971).
c) Vercoe, B., "Reference Manual for the MUSIC 360 Language for Digital Sound Synthesis", Cambridge, unpublished manuscript, Studio for Experimental Music, M.I.T., 1975.

[8] Chowning, J.M., "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation", *J.A.E.S.* 21,7 (1973).

[9] See remark 3d.

[10] Kaegi, W., Tempelaars, S., "VOSIM – A New Sound Synthesis System", *J.A.E.S.* 26,6 (1978).

relationships only in terms of computer instructions – i.e. they may be related by, and only by, machine-instructions which can alter the state of a certain register, e.g. the accumulator. In such a system, one sample may be related to the previous two samples as the result of their "XORing". The samples are conceived of in terms of machine-instructions rather than on the basis of some acoustic theory."[11] Holtzman's example of two samples only related by an XOR refers to Paul Berg's *PILE* compiler.[12] The *PILE* language was written following the development of more than 20 *ASP* programs in which random operations, referring to the accumulator, and also arithmetical and logical operations, were represented systematically. The computer acts as a sound-generating instrument sui generis, not imitating mechanical instruments or theoretical acoustic models. *PILE*, which is very popular among our students, has instructions such as BRANCH, CHOOSE, CHECK, SWITCH, STORE, SELECT, CONVERT, SEED and similar ones, which are translated by the *PILE* compiler into the assembly language of the computer so that students can also study the application of machine-language for sound production in practical examples. Typical of this nonstandard approach is that a student or composer has hardly any possibility of describing concrete musical quantities such as pitches, timbres or loudnesses and arranging them in time. Instead he must try to describe and order elements of musical language such as short, long, uniform, varied, contrast, silence, similar, dissimilar, transition and the like, these terms referring to the microstructure of the sound and to the macrostructure of the form.

The *Sound Synthesis Program* (*SSP*) I designed in 1972 proceeds from comparable viewpoints. In this non-standard approach samples are not generated by random or arithmetical/logical machine operations, but collected in sound segments which in their turn are taken from separate amplitude and time lists. The selection of amplitude and time values is made according to principles originating in *PR2*. The number-pairs in a segment designate turning-points of the oscillation curve which are interpolated linearly in real time during sound production. The number of segments which a composer can define is limited only by the capacity of the core memory; the order of segments is free within the framework of the same selection principles according to which the segments were produced. At the Institute we are considering the development of a special digitally-controlled generator which will call the sound segments from disk files, thus doing away with the limitations of the core memory. It would then be possible to realize the concept on which *SSP* is based: to describe the composition as one single sound, the perception of which is represented as a function of amplitude distribution in time as sound and silence, soft and loud, high and low, rough and smooth.

Program-generating systems

I can be brief on the subject of program-generating systems, because their development is in full swing at present, and there are no tangible results as yet. Still, investigation into composing programs and sound programs have consequences which are beyond the scope of the individual composition or construction of sounds. Although composing programs do contain fundamental statements about musical language systems, as well as personal strategies, they have neither been systematized, nor do such composing programs permit systematic research. It looks as though a super-individual approach must be found.

Steve Holtzman, of the Artificial Intelligence Department at the University of Edinburgh, has been looking into this problem recently, and has described his work at Edinburgh and Utrecht in various articles. I shall close my paper by quoting a few passages from these articles.

"The basic proposition is that music be considered as a hierarchical system which is characterized by dynamic behaviour. A *system* can be defined as a collection of connected objects ... The relation between objects which are themselves interconnections of other objects define hierarchical systems ... It should be made clear that in talking of the meaning of music or language, it is not necessary to have a referent for each sign ... Meaning is a question of

[11] See remark 6.

[12] a) Berg, P., "PILE2 – A Description of the Language", Utrecht, unpublished manuscript, Institute of Sonology, January 1978.
b) Berg, P., "A User's Manual for SSP", Utrecht, unpublished manuscript, Institute of Sonology, May 1978.

positional value. One is not concerned with the *idea* or referent as objects but rather *with values which issue from a system*. The values correspond to cultural units but they can be defined as pure differences ... *Units of meaning* are not defined referentially but structurally diacritically. We can look at a structure that consists of (diacritically) defined units in a complex of transformations and relations. Meaning becomes not *what units say/refer to/etc.* but *what they do* – a question of function in a structure."[13]

"In recent research, we have been developing a machine which itself, i.e. automatically, can generate program text to synthesize distinctive sounds and control-programs to manipulate smaller chunks of program. The machine approaches sound synthesis in the so-called non-standard manner [...]

The program generator at present occupies 5K core, with remaining free core (i.e. 20K) available for object text. It is implemented on a PDP-15 and requires a dedicated system. Special hardware used is some digital-to-analog converters with 500 nano-second response time, a hardware random number generator and a hardware arithmetic unit.

The program works in a *bottom-up* fashion first writing small chunks of text (in compiled machine-code) that create distinctive sounds, then writing control-functions to manipulate these sound-producing programs to create *phrases* of juxtaposed sounds, again, control-programs for the subordinated phrase-programs to generate larger structures, and so on. The synthesis system is hierarchical and consists of a number of distinct levels, each in turn subordinated to another [...]

Over all the rules presides what we call the complexity factor. The relations between the parameter values must interact in a manner which is within the bounds of an evaluation of their complexity measure [...] The complexity evaluation, for example, considers the number of samples that compose the sound, the larger the number of samples the more complex the wave is said to be, and similarly, the more operators used or the more variables, the greater will be the complexity. At present [...] we are trying to develop an algorithm which might be said to embody an understanding of these relationships and which could be used as the basis for a grammar to generate a grammar (which in turn generates a sound producing function)."[14]

[June 1978]

[13] Holtzman, S.H., "Music as System", DAI Working Paper 26, Department of Artificial Intelligence, University of Edinburgh, April 1978.

[14] See remark 6.