Haas, Petr

The language of the assistant and the language of the creator : selected aspects of computer assisted composition

Musicologica Brunensia. 2017, vol. 52, iss. 2, pp. 121-139

ISSN 1212-0391 (print); ISSN 2336-436X (online)

Stable URL (DOI): https://doi.org/10.5817/MB2017-2-12
Stable URL (handle): https://hdl.handle.net/11222.digilib/137654
Access Date: 17. 02. 2024
Version: 20220831

Terms of use: Digital Library of the Faculty of Arts, Masaryk University provides access to digitized documents strictly for personal use, unless otherwise specified.

MUNI ARTS Masarykova univerzita Filozofická fakulta

Digital Library of the Faculty of Arts,
Masaryk University
digilib.phil.muni.cz

Musicologica Brunensia 52 / 2017 / **2**
DOI: 10.5817/MB2017-2-12

The Language of the Assistant and the Language of the Creator: Selected Aspects of Computer Assisted Composition

Petr Haas / 215418@mail.muni.cz
Department of Musicology, Faculty of Arts, Masaryk University, Brno, CZ

Abstract

This paper is concerned with the creative process involved in making a "computer assisted composition". It follows the relationship to the technologies used, from the point of view of the communication between composer and electronic machine. It offers a perspective on the relation between the act of "programming" and the act of "composing". From this point of view, it compares the current state of the creative process with the situation at the time of the first computer-assisted realizations.

Keywords

Computer-Aided Composition, Computer Assisted Composition, computer music, language, programming

Introduction

The existence of musical works making use of computational technology is logically connected with the attainment of a necessary level of technological development. The related algorithmic or otherwise formally developed compositional thinking, however, appears much earlier in the history of Western music. Its formulation began during the development of polyphony, whose complexity grew in relation to the flowering of notation. Notation is a symbolic representation of music and a specific kind of abstraction. Its existence allows conceptual manipulation of the composition's constitutive elements. The resultant compositional methods are then a demonstration of rules, procedures and various other attempts to formalize compositional thought. Procedures and rules state how a result can be achieved. In this way, they are of an algorithmic nature. That is why some researchers point to the relation between "computer-assisted composition" (CAC)[1] and compositional practice from the beginnings of composed (primarily Western) music as such. For this same reason, they propose the term "*Algorithmic Music Composition (AMC) instead of CAC. Unfortunately, AMC* [...] *does not contain the word 'computer'. (Interestingly, I like this omission since AMC includes those non-computer composers who throughout history have used paper algorithmic composing techniques; for example, the use of 13th century isorhythm in France,* [...] *and so on.)*".[2]

We can view the compositional practice of Guillaume de Machaut or Phillipe de Vitry as a mark of rationalization, as well as belief in the effect of following compositional rules (even at the cost of conflict with the text being set). We should also mention the instructively normative position of Johann Joseph Fux's famous treatise, *Gradus ad Parnassum*. In the language of CAC discourse, this text presents a "rule-based" compositional system. The more exact the formulation of the rules; the "stricter" the counterpoint or other method – serialism being a typical example – the easier it is to formalize into the language of a machine.

1. Demonstrating Possibilities and "Computer Assisted Composition" as a Category

After a number of preceding experiments,[3] 1957 saw the first successful attempt to formalize a compositional method into the language of an electronic computer. The result

1 Other alternatives of the name include "Computer-Aided Composition" or "Composition Assistée par Ordinateur" ("CAO") in French.

2 COPE, David. Preface. In *The OM composer's book Vol. 2*. Jean Bresson – Carlos Agon – Gérard Assayag (eds.). Paris: Editions Delatour France & Ircam, 2008, p. ix.

3 In 1956, R. C. Pinkerton created the computer program that analyzed 39 melodies on the basis of the probability of occurrence of individual parameters. The computer then created its own versions of the melodies through random selection. The same principle was the basis of the "Machine to Compose Music" program (J. Cohen, J. Sowa, 1956).

was *Illiac Suite for String Quartet*, which was intended, first and foremost, "*to demonstrate that standard musical techniques could be handled by computer programming*".[4] Strict contrapuntal methods were used, which were *"formalized for pedagogical purposes in the Gradus ad Parnassum* [...] *by J. J. Fux* [...]."[5] Serial methods were also applied, in order *"to demonstrate that a computer can produce novel musical structures in a more contemporary style."*[6] The work's composer was Lejaren Hiller, who wrote the computer program itself, in collaboration with his colleague Leonard Isaacson. The piece represents one of the two basic approaches to musical creation with the use of the computer: *"One concerned with the manipulation of musical sounds, and the other concerned with symbolic representations of music. The two are iconized by Max Mathews's MUSIC program and Lejaren Hiller's Illiac Suite, both of 1957* [...]. *The two branches might provisionally be given the names 'Computer Generated Music'* [...] *and 'Computer Aided Composition' – or CGM and CAC for short."*[7]

The presence of symbolic representation of music, whose usual form is conventional notation, implies the use of acoustic instruments, usually those of the "serious" or "art" music tradition. This is connected to the fact that "*to this day, the field of CAC looks primarily to the classical 'art music' tradition*".[8] This perception of CAC is doubtless supported by the fact that spectral music, following in the tradition of New Music and the musical (post) avant-garde, is itself a specific category of CAC. "*Computer Assisted Composition (CAC) has headed towards becoming an independent discipline for the last several years. Spectral composers have made important contributions to this evolution* [...]."[9]

Furthermore, CAC is defined as a category through reference to the fact that the computer "assists" only in certain parts of the compositional process, chosen by the composer. A typical mode of "assistance" is the distribution of composed parameters using methods that are impossible to realize without a computer, or would prove unnecessarily demanding and/or time consuming. "*When doing computer-aided composition* [...] *composers decide which compositional parts or aspects of the music the computer generates, and which parts are composed manually*."[10]

Furthermore, the emphasis is on the formal parameters of the composition. In the words of the co-author of many influential programs in recent years: "*CAC systems focus on the formal structure of music*".[11]

---

4 HILLER, Lejaren – ISAACSON, Leonard. *Experimental music – Composition with an electronic computer*. New York: McGraw-Hill, 1959, p. 4.

5 HILLER – ISAACSON, 1959, op. cit., p. 84.

6 Ibid., p. 4.

7 PUCKETTE, Miller. Preface. In *The OM composer's book Vol. 1*. Carlos Agon – Gérard Assayag – Jean Bresson (eds.). Paris: Editions Delatour France & Ircam, 2008, p. ix.

8 PUCKETTE 2008, op. cit., p. xi.

9 DAUBRESSE, Eric – ASSAYAG, Gérard. Technology and creation – The creative evolution. *Contemporary music review*, 2000, vol. 19, Part 2, p. 67.

10 ANDERS, Torsten – MIRANDA, Eduardo R. Interfacing Manual and Machine Composition. *Contemporary Music Review*, 2009, vol. 28, no. 2, p. 134.

11 AGON, Carlos – ASSAYAG, Gérard (et al.). *Computer Assisted Composition at Ircam – PatchWork & OpenMusic* [online]. 1999. [cit. 2016-12-20]. URL: <http://recherche.ircam.fr/equipes/repmus/RMPapers/CMJ98/>.

2. The Creative Process in the Two Phases of Development of CAC, Concept and Formalization

For the purposes of this paper, we can schematically express the creative process with the assistance of a computer as a succession of three steps:

CREATION OF A CONCEPT –> FORMALIZATION INTO THE LANGUAGE OF THE MACHINE –> REALIZATION BY THE MACHINE

The concept, in this case, is the artistic or compositional intention, e. g. the decision to compose a serial composition, or to create an entirely new compositional style. A part of the concept is then the ensuing application of the method in order to create a plan of the whole piece. The creation of a concept is a traditional compositional act. Formalization into the language of the machine is an expression of the concept in programming language, which leads to the creation of a program, that is, a tool for shaping the structure. Creating a tool is an act of programming. The tool functions according to the rules given, which are an expression of the method. In the case of a serial composition, the tool is capable of generating various manipulations of the row. After the tool is created, its mechanical application follows – the realization by the machine.

This description of the process is theoretical. In practice, the individual steps can overlap and loop.

Currently, there are two elementary ways of creating the instrument. The first is connected with the very beginnings of CAC: the use of a programming language. It assumes the composer is also a competent programmer. The second, available since the late 1980s, offers specialized "programming environments" and their "visual language". These allow non-experts to create tools in direct interaction with the computer. This defines two moments in the development of CAC. From the point of view of this text, the confrontation that follows offers several insights into the nature of the creative process with the assistance of a computer.

3. Moment One, the Beginnings of CAC: Software as an Image of Method

Since 1957, several programs[12] – that is, tools – were created with one feature in common: they were made directly by composers, who also acted as programmers. The pro-

12 For a brief overview, mostly from a computer science perspective, see ASSAYAG, Gérard. Computer Assisted Composition today. *1st Symposium on Music and Computers. "Applications on Contemporary Music Creation: Esthetic and Technical Aspects"* [online]. 1998. [cit. 2016-12-20]. URL: <http://www.ircam.fr/equipes/repmus/RMPapers/Corfou98/>.
For a detailed survey focusing on applied technology in practice, see ROADS, Curtis (et alii). *The Computer Music Tutorial*. Cambridge: MIT Press, 1995, chapter V, p. 611.

grams' operation then corresponded to their more or less clearly defined compositional thinking. They wrote the program in languages not intended for the creation of compositional software. The author was the program's only user, perhaps with the exception of a small group of colleagues or students. The following statement has almost universal validity in this case: *"a mature musical system reflects what it is we want to accomplish, so in a very direct way the software we use reflects what it is we want to do."*[13] Typical examples are programs like SMP, Projekt 1 and MUSICOMP. The operation of the first two allowed practically no deviation from their creators' compositional position. MUSICOMP, on the other hand, can seem like a universal compositional tool.

3.1. Free Stochastic Music Program

The Free Stochastic Music Program (SMP or SM) was created by Iannis Xenakis in 1962.[14] SMP is based on the concept of stochastic music, presented by Xenakis already in 1958.[15] The logic in the operations of SMP are shown schematically in Fig. 1, the form of the output in Fig. 2. The composer can work with parameters such as the density of a section of the piece (number of notes per second), the average duration of a tone or section, or the speed and density of a glissando. First came the concept, then a schematic plan of the computation, which was then transferred into programming language. The result is what Xenakis called a "program". "*Once the program is transcribed into language that the machine's internal organization can assimilate, a process that can take several months, we can proceed to punching the cards.* [...] *The final phase is the decoding of the results into traditional notation, unless an automatic transcriber is available.*"[16] Where the possibility of transcription was unavailable, the composer had to perform it himself.

13 GERZSO, Andrew. Paradigms and Computer Music. *Leonardo Music Journal*, 1992, vol. 2, no. 1, p. 73.

14 For the genesis of the collaboration between Xenakis and IBM-Paris, see XENAKIS, Iannis. *Formalized Music, Thought and Mathematics in Composition*. Rev. ed. Stuyvesant, NY: Pendragon Press, 1992, chapter V, p. 131.

15 See XENAKIS, Iannis. Auf der Suche nach einer stochastischen Musik/Search of a Stochastic Music. In *Gravesaner Blätter / Gravesano Review*, 1958, no. 11–12, p. 98–112. Subsequently XENAKIS, Iannis. Grundlagen einer stochastischen Musik / Elements of Stochastic Music. In *Gravesaner Blätter / Gravesano Review*, 1960, no. 18, p. 61–84. 1960, no. 19–20, p. 128–140. 1961, no. 21, p. 102–113. 1961, no. 22, p. 131–144.

16 XENAKIS, Iannis. *Formalized Music*, op. cit., p. 142.

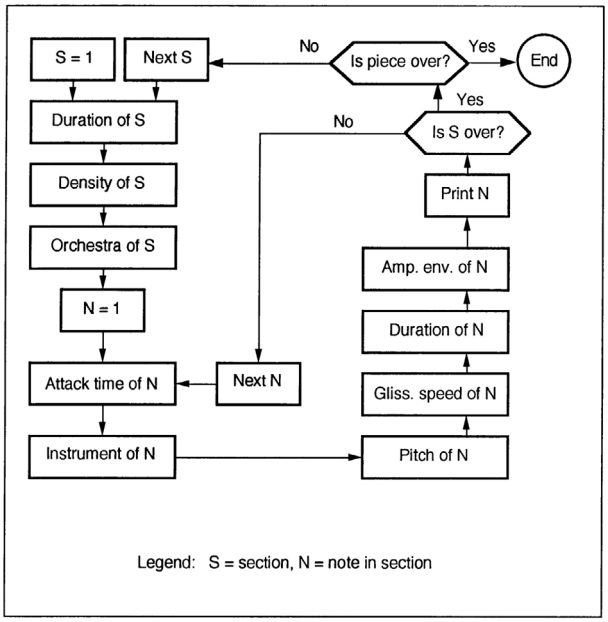


Fig. 1. A schematic description of the operation of Free Stochastic Music Program. Source: ROADS, Curtis (et alii). *The Computer Music Tutorial*. Cambridge: MIT Press, 1995, p. 838.

```
C     PROGRAM FREE STOCHASTIC MUSIC  (FORTRAN IV)                        XEN    6
C                                                                        XEN    7
C     GLOSSARY OF THE PRINCIPAL ABBREVIATIONS                            XEN    8
C
C     A - DURATION OF EACH SEQUENCE IN SECONDS                           XEN    9
C     A10,A20,A17,A35,A30 - NUMBERS FOR GLISSANDO CALCULATION            XEN   10
C     ALEA - PARAMETER USED TO ALTER THE RESULT OF A SECOND RUN WITH THEXEN   11
C     SAME INPUT DATA                                                    XEN   12
C     ALFA(3) - THREE EXPRESSIONS ENTERING INTO THE THREE SPEED VALUES   XEN   13
C     OF THE SLIDING TONES ( GLISSANDI )                                 XEN   14
C     ALIM - MAXIMUM LIMIT OF SEQUENCE DURATION A                        XEN   15
C     (AMAX(I),I=1,KTR) TABLE OF AN EXPRESSION ENTERING INTO THE         XEN   16
C     CALCULATION OF THE NOTE LENGTH IN PART 8                           XEN   17
C     BF - DYNAMIC FORM NUMBER. THE LIST IS ESTABLISHED INDEPENDENTLY    XEN   18
C     OF THIS PROGRAM AND IS SUBJECT TO MODIFICATION                     XEN   19
C     DELTA - THE RECIPROCAL OF THE MEAN DENSITY OF SOUND EVENTS DURING XEN   20
C     A SEQUENCE OF DURATION A                                           XEN   21
C     (E(I,J),I=1,KTR,J=1,KTE) - PROBABILITIES OF THE KTR TIMBRE CLASSESXEN   22
C     INTRODUCED AS INPUT DATA, DEPENDING ON THE CLASS NUMBER I=KR AND   XEN   23
C     ON THE POWER J=U OBTAINED FROM V3*EXPF(U)=DA                       XEN   24
C     EPSI - EPSILON FOR ACCURACY IN CALCULATING PN AND E(I,J),WHICH     XEN   25
C     IT IS ADVISABLE TO RETAIN.                                         XEN   26
C     (GN(I,J),I=1,KTR,J=1,KTS) - TABLE OF THE GIVEN LENGTH OF BREATH    XEN   27
C     FOR EACH INSTRUMENT, DEPENDING ON CLASS I AND INSTRUMENT J         XEN   28
C     GTNA - GREATEST NUMBER OF NOTES IN THE SEQUENCE OF DURATION A      XEN   29
C     GTNS - GREATEST NUMBER OF NOTES IN KW LOOPS                        XEN   30
C     (HAMIN(I,J),HAMAX(I,J),HBMIN(I,J),HBMAX(I,J),I=1,KTR,J=1,KTS)      XEN   31
C     TABLE OF INSTRUMENT COMPASS LIMITS, DEPENDING ON TIMBRE CLASS I    XEN   32
C     AND INSTRUMENT J.  TEST INSTRUCTION 480 IN PART 6 DETERMINES       XEN   33
C     WHETHER THE HA OR THE HB TABLE IS FOLLOWED. THE NUMBER 7 IS        XEN   34
C     ARBITRARY.                                                         XEN   35
```

Fig. 2. Output from the Free Stochastic Music Program. Source: XENAKIS, Iannis. *Formalized Music, thought and mathematic in composition*. Rev. ed. Stuyvesant, NY: Pendragon Press, 1992, p. 145.

3. 2. Projekt 1

Gottfried Michael Koenig worked on the first version of the Projekt 1 software for almost two years between 1964 and 1966.[17]

The software reflects a specific experience with the methods of serial composition: "*I cast my mind back to the early Ferienkurse in Darmstadt; was it not time to put these techniques to a systematic test?* [...] *The only answer was to try it out. The outcome was 'Project l', a* [...] *program which described a generalized model of serial composition* [...]. *Generalization was largely a matter of renouncing the 'original row plus permutation' constellation, which I replaced with random series (and hence random permutations) of data lists.*"[18] The only input one could give the program was the number of events–steps, the distance between sound events ("entry delay"), tempo and, finally, the amount of randomness ("random number"). The amount of randomness influenced the choice of rule applied to the composition's formation. The rules were based on serial methods, which were implemented in the program as "model". The author of the "model" was G. M. Koenig. Further versions of the program were developed later.[19] The output from the program was alphanumerical.[20]

3. 3. MUSICOMP

MUSICOMP (Music Simulator Interpreter for Compositional Procedures) was created in 1963.[21] Robert Baker wrote it with the assistance of L. Hiller (author of Illiac Suite for String Quartet). The software offered stochastic processes and serial compositional rules worked out in great detail. The serial rules were derived from P. Boulez's work.[22] Its "universality" consisted in also making use of the rules of tonal harmony. Apart from that, it remained an image of the methods that its creators considered note-worthy. The output was at first alphanumerical, but for the realization of *Computer Cantata* (L. Hiller, 1963), it was possible to connect the computer to another machine and receive output in the form of electronic sound.[23]

17 For the genesis, see KOENIG, Gottfried Michael. *Composition Processes* [online]. 1978, p. 7. [cit. 2016-12-20]. URL: <http://www.koenigproject.nl/indexe.htm>.

18 KOENIG, Gottfried Michael. *Working with "Project 1". My Experiences with Computer Composition* [online]. 1990, p 1. [cit. 2016-12-20]. URL: <http://www.koenigproject.nl/indexe.htm>.

19 Projekt 2 followed, see LASKE, Otto E. Compositional theory in Koenig's Project one and Project two. *Computer music journal*, 1981, vol. 5, no. 4, p. 54–65.
The current version of the program is available with agreement from G. M. Koenig, see the "download" section on the author's website [online]. URL: <http://www.koenigproject.nl/indexe.htm>.

20 For a description of *Projekt 1*, see LASKE 1981, op. cit.

21 *See* HILLER, Lejaren. Music Composed with Computers: A Historical Survey. In *The Computer and Music*. Lincoln, Harry B. (ed.). Ithaca, NY: Cornell University Press, 1970, p. 42–96.

22 *The models were Structures, livre I, 1952; Structures, livre II, 1961.*

23 See HILLER, Lejaren A. – BAKER, Robert A. Computer Cantata: A Study in Compositional Method. *Perspectives of New Music*, 1964, vol. 3, no. 1, p. 62–90.

3. 4. Closing Remarks on the Nature of the First "Assistants"

Programs like SMP or Projekt 1 were created in the 1960s and presented as illustrations of the starting point of CAC. The standard procedure involved choosing or formulating a compositional intention, the creation of a method and plan for the piece, in short, creating a concept. The following formalization into the language of the machine was a complicated process. Its length was usually in the range of months. The result is a compositional tool that can offer "assistance". No such "assistant" was universal. It could only accomplish those tasks that the composer taught it while writing the program. If the composer was unsatisfied with the assistant, it was necessary to repeat the lengthy process. Here, the act of composing preceded the act of programming, and the mutual interaction between these activities was minimal or nil.

With a certain amount of simplification, we can say that the creative process corresponded to the model given: CREATION OF A CONCEPT –> FORMALIZATION INTO THE LANGUAGE OF THE MACHINE –> REALIZATION BY THE MACHINE.

The composer could check the results on the basis of the output. These were generally alphanumerical; an automatic transcription into conventional notation was rarely available.[24] Sonic reproduction through the use of synthesis was a process whose length was in the range of hours. Programs, and also computers, were not interactive in today's sense of the word. In general, we can say that in the 1950s and 60s, *"to create a sound* [...], *composers typed instructions on a deck of punched cards.* [...] *They submitted the deck to a human 'computer operator' who fed the cards* [...] *into the computer. Returning the next day, the composer picked up a report of the job* [...]."[25]

4. Moment Two, The Development of CAC: User Interfaces and Interactive Communication

The second half of the 1980s saw the creation of the first CAC programs that functioned as "programming environments". They followed the lead of the music editors of the time, which used the so-called Common Music standard,[26] as well as making use of "object-oriented programming"[27] and programming languages such as Smalltalk.[28] A "programming environment" is a program for creating programs. In principle, they

24 For a detailed survey of the development of automatically generated results in conventional notation, see ROADS, Curtis (et alii). *The Computer Music Tutorial*. Cambridge: MIT Press, 1995, chapter 16, p. 703.

25 Ibidem, p. 613.

26 See YAVELOW, Christopher. *MacWorld Music and Sound Bible*. San Mateo: IDG Books Worldwide, 1992.

27 See *The Well-Tempered Object – Musical Applications of Object-Oriented Software Technologies*. Pope, Stephen Travis (ed.). Cambridge: MIT Press, 1991.

28 See POPE, Stephen Travis. A Smalltalk-80-based Music Toolkit. In *ICMC Proceedings 1987*. San Francisco: International Computer Music Association, 1987, p. 166–173.

worked as a user interface, thanks to which the composer could program their own "assistant" without having a programmer's competences.

These events took place in connection with research into the symbolic representation of musical structures. The Institut de Recherche et Coordination Acoustique/Musique (IRCAM) became an important center, and 1992 saw the establishment of the Représentations musicales[29] research team, focused entirely on this phenomenon. *"The Crime environment* ([...]*1985*) [...] *was the first attempt at IRCAM to realize a general CAC environment where the user could define and control abstract musical formalism*."[30]

4. 1. MIDI, Interactivity and the New Control

The changes described here coincide with the appearance of the term "personal computer" (PC). The first model was introduced to the market in 1982.[31] A year later, *"the full MIDI 1.0 Detailed Specification was first released to general public in August 1983"*.[32] Among other things, MIDI (Musical Instrument Digital Interface)[33] allowed communication between electronic devices in real time. Thanks to this, the computer could also offer an immediate interpretation of the music being created. The composer acquired a new form of control over the process as part of their communication with the machine.

4. 2. The Current Prototype and "Visual Programming"

A prototype of an entirely current form of CAC is represented by PatchWork, in development since 1989. "PatchWork (PW) is a tool for computer-assisted composition, [...]. *any* [...] *function can be translated to a graphically operable box. The programming syntax consists of making connections between boxes* [...]. PatchWork is a visual language."[34] The individual steps undertaken by the composer in the creation of his own tools were visualized on the work screen as mutually interconnected object-boxes. The objects provided outputs, thanks to which the composer could control the individual phases of the process. The outputs could be in conventional notation or a different mode of representation, and it was possible to immediately play them back. Work space with objects, cf. Fig. 3.

A group of objects created its own program inside the program, which became known as a "patch" within PatchWork. The software communicated through a clear graphical

29 A description of the team's activities, see *Music Representations Team*. [online].[cit. 2016-12-20]. URL: <http://repmus.ircam.fr/cao>.

30 ASSAYAG, Gérard – RUEDA, Camilo (et.alii). Computer – Assisted Composition at IRCAM: from PatchWork to OpenMusic. *Computer Music Journal*, 1999, vol. 23, no. 3, p. 59.

31 See CAMPBELL-KELLY, Martin. Not Only Microsoft: The Maturing of the Personal Computer Software Industry, 1982–1995. *The Business History Review*, 2001, vol. 75, no. 1, p. 103–145.

32 ROTHSTEIN, Joseph. *MIDI: A Comprehensive Introduction.* Madison: A-R Edtion, 1985, p. 11.

33 For one of the first reflections on MIDI, see LOY, Gareth. Musicians Make a Standard: The MIDI Phenomenon. *Computer Music Journal*, 1985, vol. 9, no. 4, p. 8–26.

34 ASSAYAG, Gérard – RUEDA, Camilo (et.alii). Computer – Assisted Composition at IRCAM: from PatchWork to OpenMusic. *Computer Music Journal*, 1999, vol. 23, no. 3, p. 59.

interface and used a principle known as visual programming, which can be used by non-experts (non-programmers) as well.

PatchWork, like other subsequent programs, can also be controlled by an expert programming language in order to further expand its possibilities.

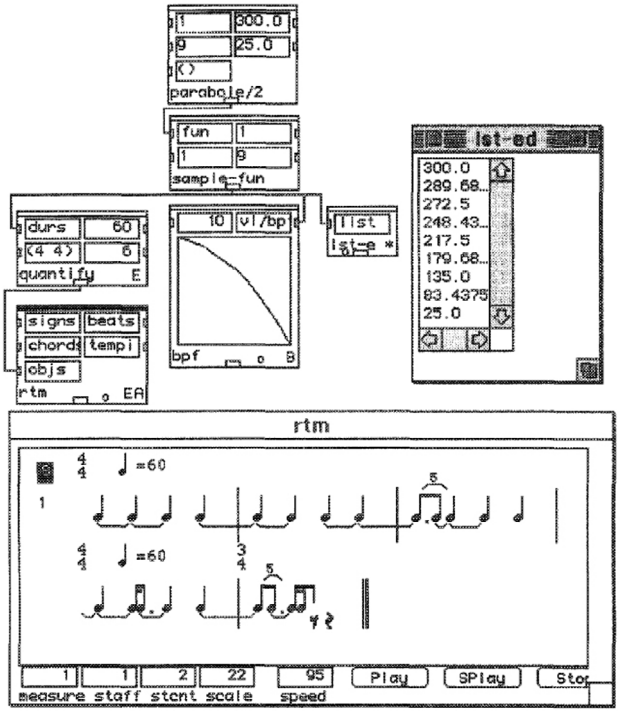


Fig. 3. Work environment with objects, a "patch" in PatchWork. Source: ASSAYAG, Gérard – RUEDA, Camilo (et.alii). Computer – Assisted Composition at IRCAM: from PatchWork to OpenMusic. *Computer Music Journal*, 1999, vol. 23, no. 3, p. 61.

Quite apart from this line of development, a number of other programs were written at his time, like Common Music (1991).[35] Their communication with the user is on the borders of a visual language like PatchWork and a programming language that is out of the layman's reach. The output can be played back or visualized in conventional notation within the program, or after exporting to another computer application.

35 See TAUBE, Heinrich. Common Music: A Music Composition Language in Common LISP and CLOS. *Computer Music Journal*, 1991, vol. 15, no. 2, p. 21–32.
For a practical description of working in the Common Music program, see TAUBE, Heinrich. *Notes from the Metalevel – Introduction to Algorithmic Music Composition*. London: Taylor & Francis, 2004.
For the program's website, see *Common Music* [online]. [cit. 2016-12-20]. URL: <http://commonmusic.sourceforge.net/>.

4. 3. Closing Remarks on the Current Prototypes

Visual programming creates a specific language on the edges of traditional representations of musical structures and a programming language. This also makes it more difficult to separate the acts of "programming" and "composing". Thanks to the interactive nature of the programming environments being created, it is possible to verify the individual steps leading to the creation of a compositional tool immediately. The verification can have practically any form, including conventional notation and playback by the computer.

A tool for future assistance can be created on the computer itself. The necessity of creating a concept before approaching the work with the computer is thus eliminated.

5. The Present of CAC

The terms "patch" and "visual programming" represent principles on which most contemporary CAC programs are based, functioning as "object-oriented visual programming environments" or "visual languages". Among the most popular are "*OpenMusic – object-oriented visual programming environment for Music Composition, Analysis and Research*",[36] developed since 1997 at IRCAM, or the "*visual programming environment called PWGL*"[37] developed since 2002 at the *Centre for Music and Technology* (CMT) at the *Sibelius Academy*. An example of the work space illustrates the manner of working, cf. Fig. 4. To use them, the composer can use visual programming, as well as expert programming methods. A popular programming paradigm, which is made available by visual programming, is so-called "constraint programming".[38] It is usually used in "rule-based" composition,[39] whose users often point out the existence of similar concepts – that is concepts based on strict adherence to rules – in pre-modern or otherwise distant times.[40]

36 BRESSON, Jean – AGON, Carlos – ASSAYAG, Gérard. OpenMusic – Visual Programming Environment for Music Composition, Analysis and Research. *Proceedings of the 19th ACM international conference on Multimedia*. Scottsdale: ACM MultiMedia, 2011 [online]. [cit. 2016-12-20]. URL: <https://hal.archives-ouvertes.fr/hal-01182394/document>.

For the program's website, see *OpenMusic* [online]. [cit. 2016-12-20].URL: <http://forumnet.ircam.fr/fr/produit/openmusic/?lang=en>.

37 LAURSON, Mikael – KUUSKANKARE, Mika– NORILO, Vesa. An Overview of PWGL, a Visual Programming Environment for Music. *Computer Music Journal*, 2009, vol. 33, no.1, p. 19. For the program's website, see *PWGL* [online]. [cit. 2016-12-20]. URL: <http://www2.siba.fi/PWGL/index.html>.

38 TRUCHET, Charlotte (Ed. and Intro.) – ASSAYAG, Gérard (Ed. and Intro.). *Constraint Programming in Music*. London: ISTE, 2011.

39 ANDERS, Torsten. *Composing Music by Composing Rules: Design and Usage of a Generic Music Constraint System*. PhD thesis. Queen's University, School of Music & Sonic Arts, Faculty of Arts, Humanities and Social Sciences, 2007.

Also see SANDRED, Örjan. PWMC, a Constraint-Solving System for Generating Music Scores. *Computer Music Journal*, 2010, vol. 34, no. 2, p. 8–24.

40 SANDRED, Örjan. Approaches to Using Rules as a Composition Method. *Contemporary Music Review*, 2009, vol. 28, no. 2, p.149–165.

For completeness' sake, we should also mention other programs such as PureData (1997),[41] which can also be used for multimedia work, or MAX, in which the "BACH – Computer-assisted Composition in Max" (2010) program is currently being developed.[42]

The contemporary programs listed here also provide most of the available methods of sound synthesis.

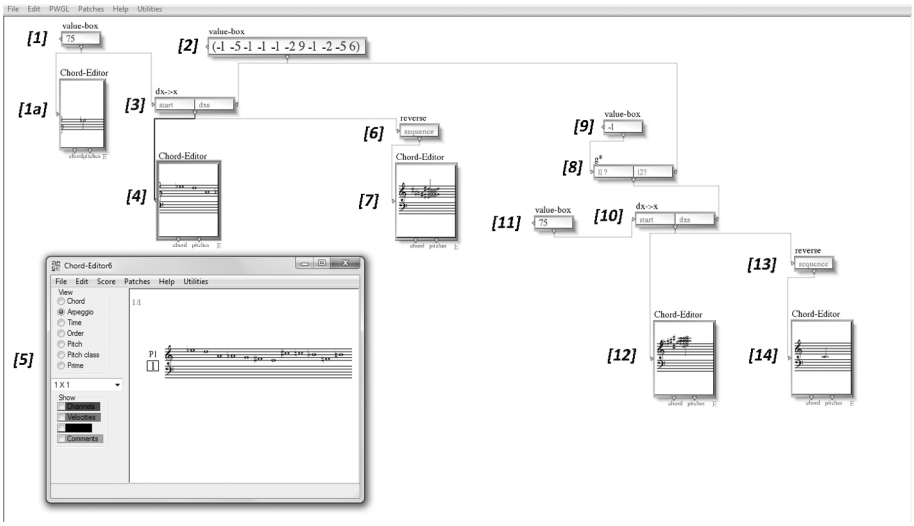


Fig. 4. Work environment in PWGL, creating the row in prime form and its variations (Boulez, Structures 1a); the input value [1] determines the pitch of the first note in the row (expressed in midicents, no. 75 represents 7500 midicents, that is D#/Eb², object [1a] shows a representation in conventional notation), input values [2] determine the interval structure of the row (expressed in midicents, negative numbers represent descending intervals, positive numbers represent ascending intervals, -1 means -100 midicents, that is a descending semitone, -5 is five descending semitones, 9 means nine ascending semitones, etc.), the function "dx->x" [3] creates a series based on the parameters given, the Chord-Editor object [4] displays the result and offers the possibility of editing [5]. The "reverse" command [6] creates a retrograde, object [7] displays the result. By multiplying [8] by -1 [9], the ascending intervals become descending and vice-versa, function [10] creates a new form of the series, that is, an inversion of the prime from the starting note given [11]. Object [12] displays the result. Function [13] creates a retrograde inversion, object [14] displays the result. Source: created by the author in PWGL.

41 For the program's website, see *PureData* [online]. [cit. 2016-12-20]. URL: <https://puredata.info/>.

42 For a theoretical reflection, see AGOSTINI, Andrea – GHISI, Daniele. A Max Library for Musical Notation and Computer-Aided Composition. *Computer Music Journal*, 2015, vol. 39, no. 2, p. 11–27.
For the project's website, see *Bach Project* [online].[cit. 2016-12-20]. URL: <http://www.bachproject.net/>.

5. 1. Symbolic Representation, Rhythm Trees and "Maquettes" of the Composition

Research into symbolic representations of music offered several new forms. This is particularly true of treatments of musical time. An example is the concept of a Rhythm Tree.[43] It represents a specific formalization of proportional expression. It offers a new method of composing, developed by computer science, cf. Fig. 5.

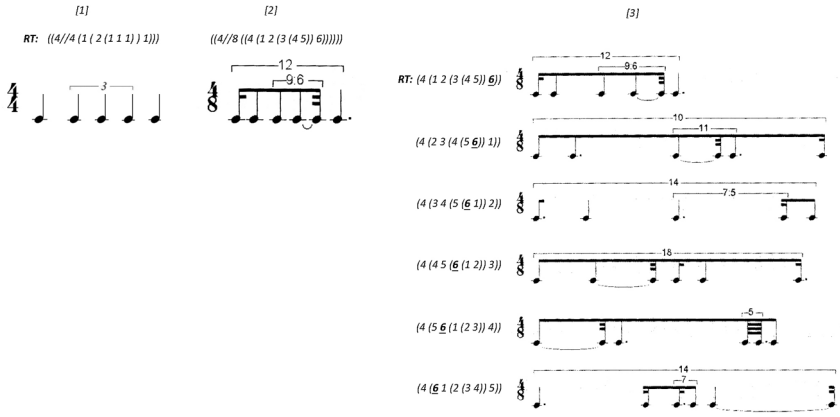
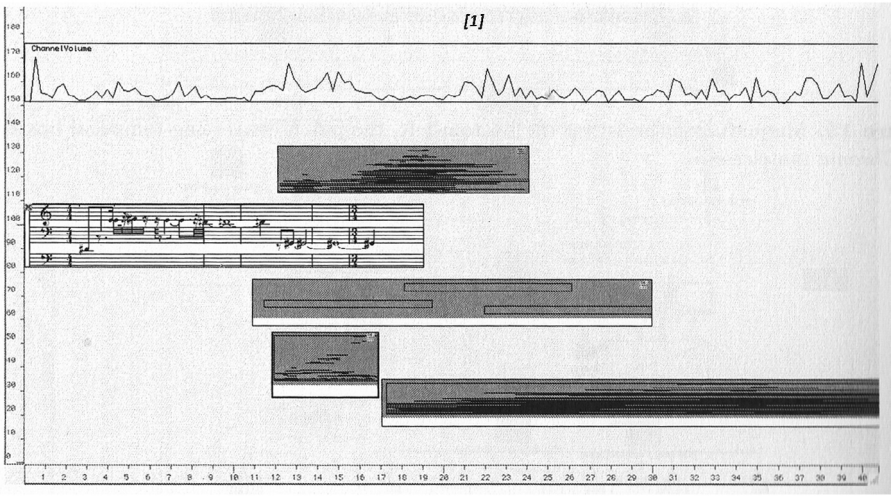


Fig. 5. Representation of rhythm with Rhythm Trees (RT), a transcription of RT into conventional notation and one of the many ways of composing musical time with RT; presentation of the basic logic and syntax of RT [1], one of the originating rhythmical units used in *String Trio,* B. Ferneyhough, 1995 [2], an outline of the logic of creating variants of the originating rhythmical unit (circular rotation highlighted by underlining the number 6)" [3]. Source: MALT, Mikhail. Some Considerations on Brian Ferneyhough's Musical Language – Part I. In *The OM composer's book Vol.2.* Jean Bresson – Carlos, Agon – Gérard, Assayag (eds.). Paris: Editions Delatour France & Ircam, 2008, p. 12, edited by the author.

This research has also led to new forms of control of the musical structure being composed. From among the many options, we should mention the representation of relations between individual compositional parameters through the use of two- or three-dimensional graphs. Furthermore, there is the option of creating "maquettes" of the composition. A maquette offers complex control when connecting various constructive elements within the piece, cf Fig. 6.

43 For the theoretical background, see AGON, Carlos – HADDAD, Karim – ASSAYAG, Gérard. Representation and Rendering of Rhythmic Structures. *International Conference of Web Delivering of Music – WedelMusic*, Darmstadt: Computer Press, 2002, p. 109–116.
For practical uses, see the OpenMusic manual, chapter *Rhythm tree* [online]. [cit. 2016-12-20]. URL: <http://support.ircam.fr/docs/om/om6-manual/co/RT1.html>.



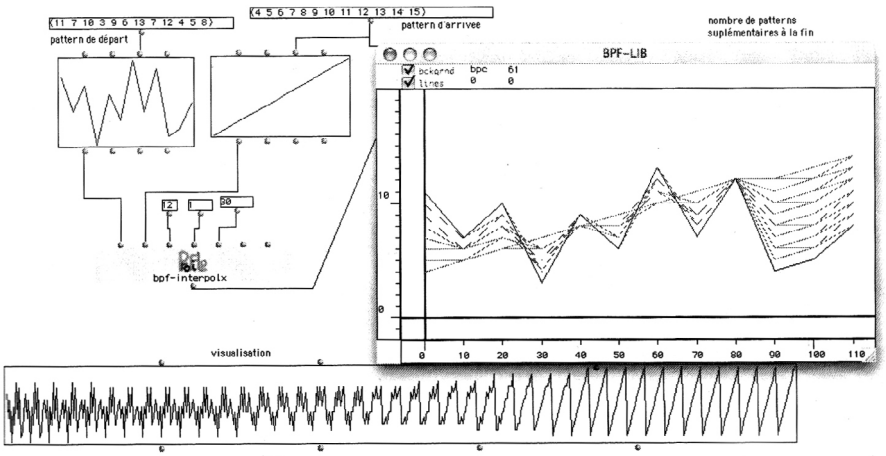


Fig. 6. A maquette of the composition and a representation of the musical structure in a two-dimensional graph, OpenMusic work environment; it is possible to place various sections of the piece into the Maquette object [1] and thus create a "maquette" of the micro- or macro-structural relationships (Maquette of the sections of *Metathesis,* Tolga Tüzün, 2006). Contemporary programs also offer the possibility of representing structures in, for example, a two-dimensional graph (interpolation of melodic lines *Hors-Jeu,* Philippe Hurel, 2006) [2]. Source: TÜZÜN, Tolga. Maquette As Data Structure and Synthesis Agent in Metathesis. In *The OM composer´s book Vol. 2.* Jean Bresson – Carlos, Agon – Gérard, Assayag (eds.). Paris: Editions Delatour France & Ircam, 2008, p. 276. DAUBRESSE, Eric – HUREL, Philippe. Creating Polyphonic Sequences in the Form of Rhythmic and Melodic Canons. In *The OM composer's book Vol. 2.* Jean Bresson – Carlos, Agon – Gérard, Assayag (eds.). Paris: Editions Delatour France & Ircam, 2008, p. 61.

5. 2. Composition as Interactive Communication with the Machine

Since the end of 1980s, CAC is no longer an area with only a few characteristic works. On the contrary, it is a pluralistic environment containing both stylistically defined approaches and any number of stylistic fusions.[44] One of the causes of this situation is that today's programming environments are neutral or universal environments with a user interface, which communicates through an easy-to-use "visual language". They are developed as non-commercial products within research projects, they are free, and the computer is a readily available consumer good.

With a view to the creative process, we can say that the method of contemporary CAC is "*to experiment with musical concepts and intuitions, to implement models, to actualize musical ideas using computer instruments and formalisms*".[45] The term "model" provides a relationships and ties between the composed parameters and that which determines "what" and "based on what" to compose. Experimenting with the musical concept, one's intuition and the actualization of the musical idea can be interactive, the individual steps can be tested individually, in real time. The program offers one the option of judging, at any moment, the partial or any other result through playback or representation in a relatively wide range of options. Using these features of contemporary CAC programming environments, the compositional process takes the form of an interactive dialogue with the machine. The composer proposes a method and the computer realizes it. The composer assesses the result. If the result is unsatisfactory, they adjust the method, and the cycle repeats. Interactive communication can replace the – previously indispensable – formulation of the concept. The concept can be absent, or it is possible to find it during the communication with the machine. Or interactive communication itself can become a concept, as in the case of "composition through interactive models".[46] A "model", in this sense, is a constellation of constraining conditions and rules, which can change in the process of dialogue. *"The machine generates one or more solutions. Being confronted with these, the composer becomes aware of aesthetic problems and refines the rule system. The musical result is optimised through repeating this loop of interaction between man and machine"*.[47]

---

44 For a unified position, see e.g. the collections describing realizations in OpenMusic: *OM Composers book, vol. 1 (2006), vol. 2 (2008), vol. 3 (2016)*; see the bibliography. Another similarly coherent collection is that of the *PRISMA* group (Pedagogia e Ricerca Internazionale sui Sistemi Musicali Assistiti /Pedagogy and International Research on Assisted Musical Systems), see BABONI-SCHILINGI, Jacopo (et alii). *PRISMA 01*. Milano: Euresis Edizioni, 2003. (The anthology *PRISMA 01* is currently unavailable. The individual papers are available upon request. A list of papers and members of the group is published on the personal web page of one of the members, see [online]. [cit. 2016-12-20]. URL: <http://johanneskretz.bplaced.net/www-jk-neu/prisma.html>.)

45 BRESSON, Jean. Introduction. *The OM composer's book Vol.2.* Jean Bresson – Carlos, Agon – Gérard, Assayag (eds.). Paris: Editions Delatour France & Ircam, 2008, p. 1.

46 See BABONI-SCHILINGI, Jacopo. Les trois niveaux d'interaction de la Composition par Modèles Interactifs vers une musique hypersistémique. In *PRISMA 01*. Milano: Euresis Edizioni, 2003, p. 129–145. (The anthology *PRISMA 01* is currently unavailable. The individual papers are available upon request. A list of papers and members of the group is published on the personal web page of one of the members, see [online]. [cit. 2016-12-20]. URL: <http://johanneskretz.bplaced.net/www-jk-neu/prisma.html>.)

47 KRETZ, Johannes. Navigation of Structured Material in Second Horizon for Piano and Orchestra. In *The OM composers book Vol. 1.* Agon, Carlos – Assayag, Gérard – Bresson, Jean (eds.). Paris: Editions Delatour

Looking to the composer – machine relation again, we can schematically describe the creative process in this relation to the machine as mutual interaction between the first two steps:

CREATION OF A CONCEPT <--> FORMALIZATION INTO THE LANGUAGE OF THE MACHINE -> REALIZATION BY THE MACHINE

6. Conclusion

The nature of composition with the assistance of a computer has, over the course of its existence, transformed in relation to the development of the technology in current use. This can be seen as the "technological" nature of CAC. This makes CAC specifically a product of the second half of the 20th century.

The programs used represent tools that form the musical structure, which are formulated by compositional methods with varying degrees of precision. These finally have to be clearly formalized into the language of the machine, so that they can be realized entirely mechanically. In the first CACs, created in the late 50s and early 60s, the difference between the formulation of a method and the ensuing formalization into the language of the machine was recognizable, despite the fact that the knowledge that the piece would finally be realized by a computer could, logically, have an influence on the creation of the concept. Generally, however, we can say that the act of programming followed the compositional act, that is the choice or creation of a method and the ensuing creation of a detailed plan of the entire composition. The creation or choice of method was the main part of the work's conception. The entire concept had to exist as a whole even before beginning work with the computer itself. Then followed a formalization into the language of the machine, which encompasses a programming act and which was usually an extensively time-consuming process, taking up to several months.

The contemporary practice, extant since the late 1980s, is different, or rather, the ordering of the individual steps is unnecessary, thanks to technological developments. Work can begin without preparation, and directly with the computer. The formulation of the concept can take place within the interactive dialogue with the machine. Within this dialogue, one can verify one's steps throughout the process, and thereby correct the direction of the work. To this end, one can choose to display partial or other results in any representation or form, including conventional notation or playback. This process is simplified by the intuitive nature of contemporary CAC "object-oriented visual programming environments", which create a specific language on the borders of traditional representations of musical structures and a programming language. Discriminating between the act of "programming" and "composing" as a series of separate activities that follow one after another is impossible in the case of contemporary CAC.

France & Ircam, 2006, p. 107.

Bibliography

AGON, Carlos – ASSAYAG, Gérard (et al.). *Computer Assisted Composition at Ircam – PatchWork & OpenMusic* [online]. 1999. [cit. 2016-12-20]. URL: <http://recherche.ircam.fr/equipes/repmus/RMPapers/CMJ98/>.

AGON, Carlos – HADDAD, Karim – ASSAYAG, Gérard. Representation and Rendering of Rhythmic Structures. *International Conference of Web Delivering of Music – WedelMusic*,

Darmstadt: Computer Press, 2002, p. 109–116.

AGOSTINI, Andrea – GHISI, Daniele. A Max Library for Musical Notation and Computer-Aided Composition. *Computer Music Journal*, 2015, vol. 39, no. 2, p. 11–27.

ANDERS, Torsten. *Composing music by composing rules: Design and usage of a generic music constraint system*. PhD thesis. Queen's University, School of Music & Sonic Arts, Faculty of Arts, Humanities and Social Sciences, 2007.

ANDERS, Torsten – MIRANDA, Eduardo R. Interfacing Manual and Machine Composition. *Contemporary Music Review*, 2009, vol. 28, no. 2, p. 133–147.

ASSAYAG, Gérard. Computer Assisted Composition today. *1st Symposium on Music and Computers. "Applications on Contemporary Music Creation: Esthetic and Technical Aspects"* [online]. 1998. [cit. 2016-12-20]. URL: <http://www.ircam.fr/equipes/repmus/RMPapers/Corfou98/>.

ASSAYAG, Gérard – RUEDA, Camilo (et.alii). Computer – Assisted Composition at IRCAM: from PatchWork to OpenMusic. *Computer Music Journal*, 1999, vol. 23, no. 3, p. 57–72.

BABONI-SCHILINGI, Jacopo. Les trois niveaux d'interaction de la Composition par Modèles Interactifs vers une musique hypersistémique. In *PRISMA 01*. Milano: Euresis Edizioni, 2003, p. 129–145. The individual papers are available upon request. A list of papers and members of the group is published on the personal web page of one of the members, see [online]. [cit. 2016-12-20]. URL: <http://johanneskretz.bplaced.net/www-jk-neu/prisma.html>.)

BABONI-SCHILINGI, Jacopo (et alii). *PRISMA 01*. Milano: Euresis Edizioni, 2003. (The anthology *PRISMA 01* is currently unavailable. The individual papers are available upon request. A list of papers and members of the group is published on the personal web page of one of the members, see [online]. [cit. 2016-12-20]. URL: <http://johanneskretz.bplaced.net/www-jk-neu/prisma.html>.)

BRESSON, Jean. Introduction. *The OM composer´s book Vol. 2.* Jean Bresson – Carlos, Agon – Gérard, Assayag (eds.). Paris: Editions Delatour France & Ircam, 2008, p. 1–6.

BRESSON, Jean – AGON, Carlos – ASSAYAG, Gérard. OpenMusic – Visual Programming Environment for Music Composition, Analysis and Research. *Proceedings of the 19th ACM international conference on Multimedia*. Scottsdale: ACM MultiMedia, 2011 [online]. [cit. 2016-12-20]. URL: < https://hal.archives-ouvertes.fr/hal-01182394/document>.

CAMPBELL-KELLY, Martin. Not Only Microsoft: The Maturing of the Personal Computer Software Industry, 1982–1995. *The Business History Review*, 2001, vol. 75, no. 1, p. 103–145.

COPE, David. Preface. In *The OM composer's book Vol. 2.* Jean Bresson – Carlos Agon – Gérard Assayag (eds.). Paris: Editions Delatour France & Ircam, 2008, p. ix–xv.

DAUBRESSE, Eric – ASSAYAG, Gérard. Technology and creation – The creative evolution. *Contemporary music review*, 2000, vol. 19, Part 2, p. 61–80.

DAUBRESSE, Eric – HUREL, Philippe. Creating Polyphonic Sequences in the Form of Rhythmic and Melodic Canons. In *The OM composer's book Vol. 2.* Jean Bresson – Carlos, Agon – Gérard, Assayag (eds.). Paris: Editions Delatour France & Ircam, 2008, p. 55–64.

GERZSO, Andrew. Paradigms and Computer Music. *Leonardo Music Journal*, 1992, vol. 2, no. 1, p. 73–79.

HILLER, Lejaren. Music composed with computers: A historical survey. In *The computer and music*. Lincoln, Harry B. (ed.). Ithaca, NY: Cornell University Press, 1970, p. 42–96.

HILLER, Lejaren – ISAACSON, Leonard. *Experimental music – Composition with an electronic computer*. New York: McGraw-Hill, 1959.

HILLER, Lejaren A. – BAKER, Robert A. Computer Cantata: A Study in Compositional Method. *Perspectives of New Music*, 1964, vol. 3, no. 1, p. 62–90.

KOENIG, Gottfried Michael. *Composition Processes* [online]. 1978. [cit. 2016-12-20]. URL: <http://www.koenigproject.nl/indexe.htm>.

KOENIG, Gottfried Michael. *Working with "Project 1". My Experiences with Computer Composition* [online]. 1990. [cit. 2016-12-20]. URL: <http://www.koenigproject.nl/indexe.htm>.

KRETZ, Johannes. Navigation of Structured Material in Second Horizon for Piano and Orchestra. In *The OM composer's book Vol. 1*. Agon, Carlos – Assayag, Gérard – Bresson, Jean (eds.). Paris: Editions Delatour France & Ircam, 2006, p. 97–113.

LASKE, Otto E. Compositional theory in Koenig's Project one and Project two. *Computer music journal*, 1981, vol. 5, no. 4, p. 54–65.

LAURSON, Mikael – KUUSKANKARE, Mika – NORILO, Vesa. An Overview of PWGL, a Visual Programming Environment for Music. *Computer Music Journal*, 2009, vol. 33, no. 1, p. 19–31.

LOY, Gareth. Musicians Make a Standard: The MIDI Phenomenon. *Computer Music Journal*, 1985, vol 9, no. 4, p. 8–26.

MALT, Mikhail. Some Considerations on Brian Ferneyhough's Musical Language – Part I. In *The OM composer's book Vol. 2*. Jean Bresson – Carlos, Agon – Gérard, Assayag (eds.). Paris: Editions Delatour France & Ircam, 2008, p. 7–19.

POPE, Stephen Travis. A Smalltalk-80-based Music Toolkit. In *ICMC Proceedings 1987*. San Francisco: International Computer Music Association, 1987, p. 166–173.

PUCKETTE, Miller. Preface. In *The OM composer's book Vol. 1*. Carlos Agon – Gérard Assayag – Jean Bresson (eds.). Paris: Editions Delatour France & Ircam, 2008, p. ix–iix.

ROADS, Curtis (et alii). *The Computer Music Tutorial*. Cambridge: MIT Press, 1995.

ROTHSTEIN, Joseph. *MIDI: A Comprehensive Introduction.* Madison: A-R Edtion, 1985.

SANDRED, Örjan. Approaches to Using Rules as a Composition Method. *Contemporary Music Review*, 2009, vol. 28, no. 2, p. 149–165.

SANDRED, Örjan. PWMC, a Constraint-Solving System for Generating Music Scores. *Computer Music Journal*, 2010, vol. 34, no. 2, p. 8–24.

TAUBE, Heinrich. Common Music: A music composition language in Common LISP and CLOS. *Computer Music Journal*, 1991, vol 15, no. 2, p. 21–32.

TAUBE, Heinrich. *Notes from the metalevel – Introduction to algorithmic music composition*. London: Taylor & Francis, 2004.

TRUCHET, Charlotte (Ed. and Intro.) – ASSAYAG, Gérard (Ed. and Intro.). *Constraint Programming in Music*. London: ISTE, 2011.

TÜZÜN, Tolga. Maquette As Data Structure and Synthesis Agent in Metathesis. In *The OM composer's book Vol. 2*. Jean Bresson – Carlos, Agon – Gérard, Assayag (eds.). Paris: Editions Delatour France & Ircam, 2008, p. 263–277.

XENAKIS, Iannis. *Formalized Music, thought and mathematic in composition*. Rev. ed. Stuyvesant, NY: Pendragon Press, 1992.

XENAKIS, Iannis. Grundlagen einer stochastischen Musik/Elements of Stochastic Music. In *Gravesaner Blätter / Gravesano Review*, 1960, no. 18, p. 61–84. 1960, no. 19–20, p. 128–140. 1961, no. 21, p. 102–113. 1961, no. 22, p. 131–44.

YAVELOW, Christopher. *MacWorld music and sound bible*. San Mateo: IDG Books Worldwide, 1992.

Bach Project [online]. [cit. 2016-12-20]. URL: <http://www.bachproject.net/>.

Common Music [online]. [cit. 2016-12-20]. URL: <http://commonmusic.sourceforge.net/>.

Music Representations Team [online]. [cit. 2016-12-20]. URL: <http://repmus.ircam.fr/cao>.

OpenMusic [online]. [cit. 2016-12-20]. URL: <http://forumnet.ircam.fr/fr/produit/openmusic/?lang=en>.

PureData [online]. [cit. 2016-12-20]. URL: <https://puredata.info/>.

PWGL [online]. [cit. 2016-12-20]. URL: <http://www2.siba.fi/PWGL/index.html>.

Rhyth tree [online]. [cit. 2016-12-20]. URL: <http://support.ircam.fr/docs/om/om6-manual/co/RT1.html>.

The OM composer's book Vol. 1. Agon, Carlos – Assayag, Gérard – Bresson, Jean (eds.). Paris: Editions Delatour France & Ircam, 2006.

The OM composer's book Vol. 2. Jean Bresson – Carlos, Agon – Gérard, Assayag (eds.). Paris: Editions Delatour France & Ircam, 2008.

The OM composer's book Vol. 2. Jean Bresson – Carlos, Agon – Gérard, Assayag (eds.). Paris: Editions Delatour France & Ircam, 2008.

The well-tempered object – Musical applications of object-oriented software technologies. Pope, Stephen Travis (ed.). Cambridge: MIT Press, 1991.