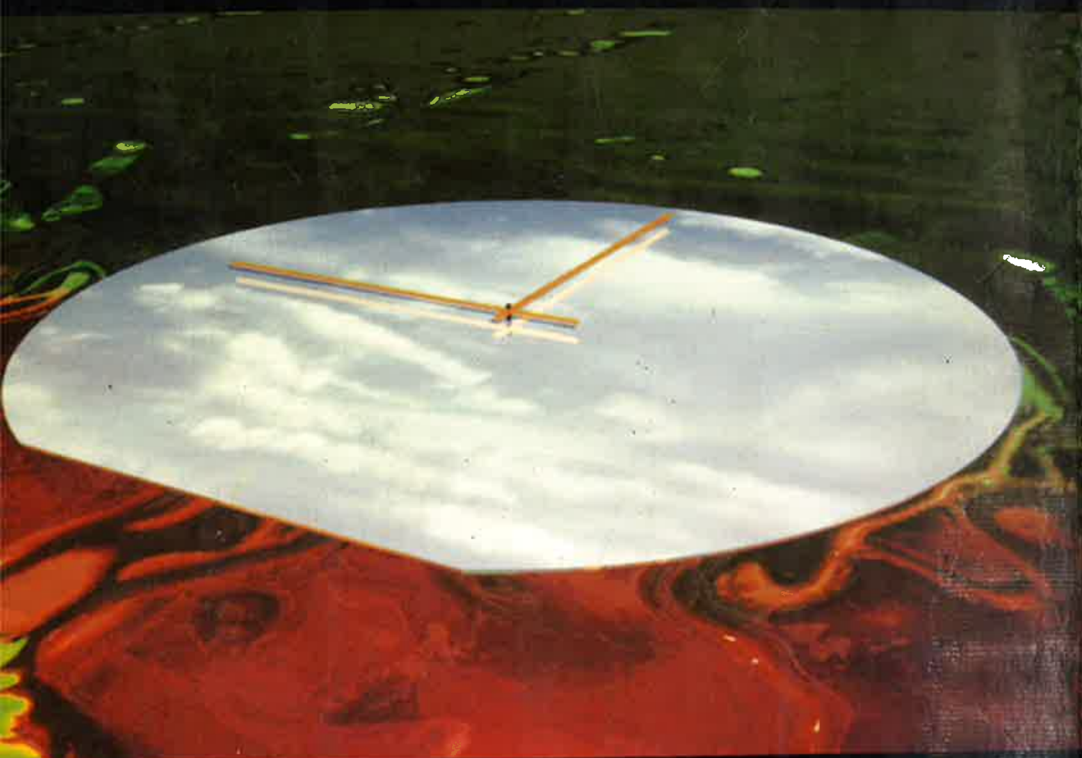


PATROCÍNIO:

CNPq
FINEP
IBM Brasil
CAPES

APOIO:

FAPERGS
FUNDAÇÃO ROCKFELLER
SOFTSUL
SOFTEX 2000
(NÚCLEOS STA.CATARINA)
SILICON GRAPHICS
CP ELETRÔNICA
RBS

PROMOÇÃO:

SBC
UFRGS

REALIZAÇÃO:

II / UFRGS

II SBC&M - 2ND BRAZILIAN SYMPOSIUM ON COMPUTER MUSIC

PROCEEDINGS

II SBC&M - 2ND BRAZILIAN SYMPOSIUM ON COMPUTER MUSIC

PROCEEDINGS

15TH CONGRESS OF THE BRAZILIAN COMPUTER SOCIETY

CANELA, RS - BRAZIL • 1995

SOCIEDADE BRASILEIRA DE COMPUTAÇÃO

Informática UFRGS

II Simpósio Brasileiro de Computação e Música

Canela - RS
29 de julho a 1º de agosto de 1995

ANAIS

Editado por
Eduardo Reck Miranda

Realização
Instituto de Informática
Universidade Federal do Rio Grande do Sul

Promoção
Sociedade Brasileira de Computação - SBC
Instituto de Informática da UFRGS

Capa: Flávio V. Cauduro - NID/Comunicação UFRGS

Impressão: Gráfica Editora Pallotti

Dados Internacionais de Catalogação-na-Publicação (CIP)
(Biblioteca do Instituto de Informática da UFRGS, Porto Alegre, RS)

Simpósio Brasileiro de Computação e Música (2. : 1995 jul. 29-Ago. 1 : Canela)
Anais / II Simpósio Brasileiro de Computação e Música ; editado por Eduardo Reck Miranda. - Porto Alegre : Instituto de Informática da UFRGS, 1995.
301 p.

Evento realizado no XV Congresso da Sociedade Brasileira de Computação.

1. Informática. 2. Música. I. Miranda, Eduardo Reck. II. Congresso da Sociedade Brasileira de Computação (15. : 1995 jul. 29-Ago. 4 : Canela). III. Título.

Cópias adicionais

Universidade Federal do Rio Grande do Sul
Instituto de Informática - Setor de Eventos
Caixa Postal 15064
91501-970 Porto Alegre RS Brasil
Telefone: (051) 336 83 99 ramal 6166
Fax: (051) 336 55 76
E-mail: eventos@inf.ufrgs.br

Prefácio

O Congresso da Sociedade Brasileira de Computação, que neste ano tem sua décima-quinta edição, é o mais tradicional evento científico da área de computação no Brasil. Como em outros anos, ele reune em 1995 diversos eventos paralelos, nos quais são discutidos temas técnicos, políticos, educacionais e profissionais.

Repetindo o que ocorreu há dez anos atrás, mais uma vez a comunidade de Informática da UFRGS traz para nosso Estado o PANEL - Conferência Latino-Americana de Informática, promovido pelo CLEI - Centro Latinoamericano de Estudios en Informatica, para ser realizado em conjunto com o Congresso da SBC. O PANEL está em sua vigésima-primeira edição, e é um dos mais importantes eventos científicos da América Latina, sempre reunindo grande número de participantes.

Na área científica, o XXI SEMISH - Seminário Integrado de Software e Hardware e o PANEL'95, integrados, têm a apresentação de 115 trabalhos de vários países da América Latina, Europa e Estados Unidos. Estes trabalhos foram selecionados dentre mais de 250 artigos submetidos.

Em paralelo com o SEMISH/PANEL, acontece a apresentação de 43 trabalhos no VII SBAC-PAD - Simpósio Brasileiro de Arquitetura de Computadores e Processamento de Alto Desempenho, 25 trabalhos no VI SCTF - Simpósio de Computadores Tolerantes a Falhas, e 39 trabalhos e 6 concertos no II SBC&M - Simpósio Brasileiro de Computação e Música. Todos estes eventos, inclusive o SEMISH/PANEL'95, oferecem ainda diversas palestras convidadas, proferidas por especialistas de renome internacional.

Na área de ensino e atualização em Informática, 11 cursos de curta duração foram selecionados para apresentação na XIV JAI - Jornada de Atualização em Informática. O VIII CTD/SBC - Concurso de Teses e Dissertações da SBC, realizado junto com o II CTD/CLEI, mesmo concurso do CLEI, e o XIV CTIC - Concurso de Trabalhos de Iniciação Científica, apresentam, também dentro do Congresso, trabalhos premiados de alunos de pós-graduação e graduação, respectivamente.

O III WEI - Workshop de Educação em Informática, realizado em conjunto com o IV EDUC - Congresso Ibero-Americano de Educação Superior em Computação, oferece aos interessados diversos painéis, grupos de trabalho, palestras convidadas e artigos selecionados, que tratam de temas como desenvolvimento curricular e formação profissional. Estes temas, aliados a questões de regulamentação profissional e mercado de trabalho, são também abordados no XIII ENECOMP - Encontro Nacional de Estudantes de Computação.

Na parte política do Congresso, o XXV SECOMU - Seminário de Computação na Universidade combina uma retrospectiva histórica da computação no Brasil nos últimos 25 anos com a elaboração, com ampla participação da comunidade, de um plano estratégico para os anos vindouros.

O XV Congresso da Sociedade Brasileira de Computação e o PANEL'95 - XXI Conferência Latino-Americana de Informática são organizados pelo Instituto de Informática da Universidade Federal do Rio Grande do Sul. O sucesso destes eventos é fruto do trabalho dedicado e competente da Comissão Organizadora global e dos responsáveis por cada um dos eventos paralelos. É preciso também agradecer o trabalho qualificado dos Comitês de Programa e de mais de uma centena de avaliadores de artigos. O Congresso pode ser realizado graças ao suporte financeiro do CNPq, FINEP, FAPERGS, CAPES e IBM Brasil e ao apoio da UFRGS, FAPESP, UNESCO e várias outras empresas.

Canela, julho de 1995

Flávio R. Wagner
Coordenador do XV Congresso da SBC

José Mauro V. de Castilho
Coordenador do PANEL'95 - XXI Conferência Latino-Americana de Informática

Foreword by the Chairman

Welcome to the II SBC&M! Today, computers are commonly used in every aspect of daily life; leading to a change in attitude towards them, from a mere scientific accessory, to a medium for creativity.

There is a tradition in the development of musical instruments to use technical advances and increased scientific understanding in order to produce the best sounds; computer music is therefore a natural progression. Computer music, however, does not involve only the design of new and better instruments. It is a discipline that embodies a wide range of musicological research, from speculative machine modelling of the mind's creative process, to pragmatic music analysis in the field of ethnomusicology. Today, computers may play a key role in several aspects of music-making, ranging from synthesis of complex sounds that are not possible to produce with acoustic instruments, to automatic composition of music for film and TV; even robots can play music now!

NUCOM, the Computer Music Commission of SBC, was created in 1993. The aim of NUCOM is to foster research in computer music in Brazil and to promote integration amongst researchers and musicians. Following the success of the I SBC&M, held in Caxambu in 1994, the II SBC&M has consolidated Brazil as a leading Latin-American country in computer music research. The programme this year includes 2 tutorials (Rick Taube, ZKM, Germany and Aluizio Arcela and co-workers, UnB-LPE, Brazil), 6 key lectures (Trevor Wishart, Marc Leman, Rick Taube, Steve Pope, Richard Moore and Chris Chafe), 8 sessions of papers and 6 concerts featuring worldwide composers.

The Proceedings of the II SBC&M include a CD with selected compositions. On the CD you will find works by experienced pioneers and by young, but rather promising composers. The compositions demonstrate a variety of styles; including algorithmic composition, the *acousmatique* approach, acoustic instruments plus live electronics, computer improvisation and note level-based works.

The II SBC&M organizers wish to thank all participants and the support of all the institutions and individuals, who in some way contributed to make this event possible.

Eduardo Reck Miranda
Canela, July 1995

II SBC&M

II Simpósio Brasileiro de Computação e Música

Coordenador Geral	
Eduardo Reck Miranda	UFRGS, Brasil
Vice Coordenador	
Eloi Fernando Fritsch	UFRGS, Brasil
Auxiliar de Coordenação	
Fábio Ghignatti Beckenkamp	UFRGS, Brasil
Comissão de Artigos	
Aluízio Arcela	UnB, Brasil
Carlos Cerana	LIPM, Argentina
Conrado Silva	UnB, Brasil
Geber Ramalho	Université de Paris VI, França/Brasil
Jamary Oliveira	UFBA , Brasil
Comissão Artística	
Celso Loureiro Chaves	UFRGS, Brasil
Conrado Silva	UnB, Brasil
Francisco Kröpfl	LIPM, Argentina
Maurício Álvares Loureiro	UFMG, Brasil
Robert Willey	University of Califonia San Diego, USA
Produção do CD	
Eduardo Reck Miranda	UFRGS, Brasil
Robert Willey	University of Califonia San Diego, USA
Coordenação dos Concertos	
Conrado Silva	UnB, Brasil
José Augusto Mannis	CDMC-Unicamp, Brasil
Maurício Álvares Loureiro	UFMG, Brasil

XV Congresso da Sociedade Brasileira de Computação

Coordenador Geral	
Flavio Rech Wagner	UFRGS
Secretária	
Lourdes Tassinari	UFRGS
Comissão Organizadora	
Carla Maria Dal Sasso Freitas	UFRGS
Carlos Alberto Heuser	UFRGS
Cirano Iochpe	UFRGS
Flávio Rech Wagner	UFRGS
Gabriela da Silva Conceição	SBC
Iara Pereira Claudio	PUC-RS
José Mauro Volkmer de Castilho	UFRGS
Luis Otávio Soares	UFRGS
Mara Abel	UFRGS
Maria Aparecida Castro Livi	UFRGS
Maria Janilce Almeida	UFRGS
Nina Edelweiss	UFRGS
Ricardo Augusto da Luz Reis	UFRGS
Roberto Cabral de M. Borges	UFRGS
Rosa M. Viccari	UFRGS
Silvia D. Olabarriaga	UFRGS
Tiarajú Asmuz Diverio	UFRGS
Tiaraju Wagner	UFRGS

Sociedade Brasileira de Computação

Diretoria

Presidente	
Ricardo Augusto da Luz Reis	UFRGS
Vice-Presidente	
Paulo Roberto Freire Cunha	UFPE
Vice-Presidente Adjunto	
Edson C. B. Carvalho Filho	UFPE
Secretário Geral	
Cirano Iochpe	UFRGS
Secretário Geral Adjunto	
Flávio Rech Wagner	UFRGS
1ª Secretária	
Cláudia Maria Bauzer Medeiros	UNICAMP
1º Secretário Adjunto	
Miguel Jonathan	UFRJ
2º Secretário	
Roberto da Silva Bigonha	UFMG
2º Secretário Adjunto	
Clarindo Isaías P. Silva e Pádua	UFMG
Tesoureira	
Therezinha Souza da Costa	PUC-RJ
Tesoureiro Adjunto	
Emmanuel Lopes Passos	IME
Conselho - Membros Titulares	
Cláudia Lage Rebelo da Motta	UFRJ
Clesio Saraiva dos Santos	UFRGS
Daltro José Nunes	UFRGS
Luiz de Castro Martins	LNCC
Marcos Roberto da Silva Borges	UFRJ
Nivio Ziviani	UFMG
Pedro Manoel da Silveira	UFRJ
Philippe Olivier Alexandre Navaux	UFRGS
Siang Wun Song	USP
Silvio Romero Lemos Meira	UFPE
Conselho - Membros Suplentes	
Alberto Henrique Frade Laender	UFMG
Cláudio Kirner	USP S. Carlos
Daniel Schwabe	PUC-RJ
Ricardo Anido	UNICAMP
Sérgio de Mello Schneider	UFU

Sumário

Palestras Especiais

Artigos I: Sistemas para Composição I

Artigos II: Sistemas Interativos para a Música & *Live-Electronics*

Artigos III: Música e Inteligência Artificial & Modelagem Física de Instrumentos Musicais

Artigos IV: Reconhecimento Automático de Partituras & Análise Musical Auxiliada por Computador

Artigos V: Processamento de Sinais e Síntese Sonora

Artigos VI: O Papel do Computador no Estúdio

Artigos VII: Sistemas para Composição II

Papers VIII: O Computador na Educação Musical

O CD do Simpósio

Fifteen Years of Computer-Assisted Composition

STEPHEN TRAVIS POPE
Computer Music Journal, and CNMAT, Dept. of Music, U. C. Berkeley.
P. O. Box 9496, Berkeley, California, 94709 USA
Electronic mail: stp@CNMAT.Berkeley.edu
World-Wide Web: http://www.cnmat.berkeley.edu/~stp/

Abstract

This paper describes several generations of computer music systems and the music they have enabled. It will introduce the software tools used in some of my music compositions realized in the years 1979-94 at a variety of studios using various software and hardware systems and programming languages. These tools use a wide range of compositional methods, including (among others): high-level graphical notations, limited stochastic selection, Markov transition tables, forward-chaining expert systems, non-deterministic Petri networks, and hierarchical rule-based knowledge systems. The paper begins by defining several of the terms that are frequently used in the computer music literature with respect to computer-aided composition and realization, and introduces several of the categories of modern models of music composition. A series of in-depth examples are then drawn from my works of the last 15 years, giving descriptions of the models, the software tools, and demonstrating the resulting music.

Background

Composers of music in the Western tradition have always used the highest technology that was available to them (Pope, 1991a). Examples of this can be found throughout music history, be it J. S. Bach's use and advocacy of equal temperament (remember that Bach was a contemporary of Euler—the inventor of logarithms, on which the equal-tempered scale is based), or the heavy use of digital technology by composers and performers today. Because composition involves many technical processes and typical "engineering" trade-offs, there is no contradiction for a composer to use a computer and learn a software user interface or programming language; common-practice Western music notation is, after all, a highly technical and abstract description language for performance gestures and has very little to do with the resulting music or personal expression.

There are several good surveys of the use of computers in music composition (Loy and Abbott, 1985; Loy, 1989); the field is actively growing, as new informatics techniques find applications in music composition—recently artificial neural networks, the mathematics of non-linear dynamical systems, and genetic algorithms have been seen here. Some abstract representations—e.g., the use of instrument definitions and note lists—and classes of compositional algorithms—e.g., common procedural or stochastic models—have been in common use since the "dawn" of computer music in the 1950s and 1960s. This paper will survey several of these popular techniques (and a few rather obscure ones), and present concrete examples of the musical results and graphical user interfaces.

Algorithmic Composition and High-level Composition Tools

In the last 40 years, computers and informatics technology have found a wide range of applications in music composition, realization and real-time performance. Note, however, that the kinds of abstractions and tools used in these three areas—composition, production, and performance—are very different. Composers need software support for the refinement of incompletely specified ideas and the capture of abstract information early in the process of composition, and for the generation of scores or the transition from composition to production or performance (Pope, 1986, 1993b). The wide variety of processes involved in composition, and the vague and abstract nature of high-level musical material, both place special requirements on the design of software tools to support composers.

There is a wide-spread misunderstanding that all uses of software tools in the composition or production of music imply algorithmic composition. It is important to differentiate between a composer who uses the computer models of musical structures in the process of composition and organization of musical material, and the performer or producer who uses digital tools for the "later" tasks of music performance, production, or recording.

What I wish to focus on here is the use of computer-based software systems in the capture and elaboration of musical ideas—the use of computers as "composer's assistants" rather than as simple scribes or digital instruments. In this category, it is important that the computer have some software-based model of the musical structure or process. I will discuss below the different families of compositional algorithms and software-based composer's tools, and present examples of their use in my musical compositions from the period 1979-94.

Procedural, Stochastic, and Knowledge-based Systems

There are many ways to model complex data structures or dynamical systems using contemporary mathematical and informatics-based techniques. *Procedural* descriptions such as those expressed in common programming languages are widely used to describe the processes and policies of real-world systems in computer-based simulations. It is also possible to describe musical processes in a procedural manner, and there are numerous software systems that allow one to construct "musical automata" using straight-forward imperative procedural programming techniques. Music-oriented function libraries exist for most common programming languages such as C, Pascal, BASIC, Lisp, and Smalltalk. These generally allow the user to create some kind of musical data structure (often using MIDI parameters in lieu of a higher-level abstraction for musical parameters) and to store it to a file or play it in real time. There are also several kinds of music-specific languages that take other languages as their models, e.g., Barry Vercoe's *Csound* is based on assembly language; F. Richard Moore's *cmusic* on C, and William Schottstaedt's *clm* on Common Lisp.

Very complex dynamical systems cannot always be modelled directly in a procedural programming language, and modelers often resort to *stochastic* models that use grammars or other structuring paradigms together with limited random selection of values. There are many kinds of stochastic models, the most common of which are based on relatively simple structures (described in a procedural or static manner) that are elaborated with bounded stochasm (randomness). This can be as simple as declaring a musical structure (e.g., a melody) and adding some amount of random variation to its parameters to make it seem more "life-like." (Indeed, some commercial synthesizers and sequencers have a "humanize" function that does just this.) One commonly used higher-level stochastic model is that of generative grammars, which can be used to describe the "rules" of a given musical structure. Given a small set of rules (each of which describes a transition between a set of starting tokens and a set of final tokens), it is possible to generate musical structures that have a high degree of "internal consistency" (on some linguistic level at least). There is an active debate among music theorists as to whether or not music is "linguistic" in terms of the applicability of generative grammars to higher-level musical structures. Other common models involve bounded stochasm, where the computer is used to make a selection from a given set or range of values.

Several kinds of *knowledge-based* frameworks are also popular today, including rule-based, logic-based, and neural-network-based models. These systems differ from the previous types in that the "knowledge" of the musical domain is programmed in a declarative style—you tell it *what* there is to know about the task, rather than *how* to solve it. There is a debate (that I would like to avoid) in the literature as to whether neural nets indeed constitute a "knowledge-based" model, since there is no direct knowledge model involved, but rather a complex statistical model serves as their base. The literature of artificial intelligence (AI) and music is very rich, and numerous kinds of knowledge-based frameworks for music composition have been developed and used in real-world situations by well-known composers.

There are, of course, other basic modelling paradigms, such as statistical models—which may be classified as procedural or stochastic, depending on their structure. The above list is not intended to be complete, but rather to provide basic terms for categorizing the algorithmic composition approaches I will introduce below.

Description Languages, Automata, and Graphics Tools

The various stages of the composition process often call for several different kinds of tools or languages. Sometimes, a composer wants to use a formal language to describe a specific musical structure or process, e.g., for the purpose of annotation. In this case, he/she can use any of a large variety of formal languages, ranging from logic declarations to programming languages. There is a rich literature of "music input languages" that includes music description formats based on just about every known computer science formalism (Loy and Abbott, 1985; Loy, 1989; Pope, 1993a). Some of these languages are based on the declaration of simple, linear "note list" structures, while others allow the user to describe higher-level musical structures such as chords, performance gestures, and musical forms as well (Pope, 1986, 1989).

"Algorithmic composition" implies that the composer builds a program (the algorithm) that then composes (all or only some features of) the music. As described above, this description can take several forms, including procedural, stochastic, or knowledge-based software tools that are "generative" in the sense that they take some data as their input and generate musical data as output.

Graphical tools such as score editors based on common-practice Western music notation are widely used in music processing, but composers rarely have access to higher-level (i.e., higher-level than the "note") notations. There has been some interesting research in the development of formal notations (i.e., those that are based on a strict encoding and mapping between graphical symbols and musical objects) for musical structures, and several composers have used these in the real musical settings (Loy, 1989). There is a continuum of options between complex score notations—pitch-time diagrams where the icons used have more information than traditional note heads and stems—and graphical programming languages for musical algorithms. This is a very interesting field of research in which there is much left to be discovered.

Examples

In this section, I will illustrate a variety of software-based composition techniques and composer's tools that I have used in my own compositions. In the live presentation, I intend to show examples of the software user interfaces used for each of these pieces, and to play short musical examples that demonstrate the compositional structures or algorithms that are most heavily used.

I will not present the in-depth reference list or "picture show" in this paper, but instead refer the interested reader to the publications that are cited below. (I have published detailed articles on each of the tools I present here.) The examples are presented in chronological order, rather than by the level of sophistication of the technology they use.

Graphical Notations: *WAKE*

WAKE: Ten Tangents for Dance was realized at the SSSP (Structured Sound Synthesis Project) studio within the computer systems research group (CSRG) at the University of Toronto during1979/80. This piece originated as a work for organ solo, and had several generations of graphical scores including common-practice notation and a "piano-roll-style" score (i.e., a pitch vs. time plot) where color was used to denote organ voicing. This graphical score was "transcribed" for use by the SSSP's digital synthesizer rather easily. The further revisions to the score used several graphical notations that were specific to the SSSP system (Buxton et al. 1979), and for which tools were written in C. These notations went beyond the simple pitch/time diagrams and included structure trees, timbral notations, and other non-traditional graphical representations for which computer editors were built.

This is an example of realizing an "ad hoc" composition—where the compositional strategy was implemented without computer assistance—with a formal software-based tool. In this case, the score existed before-hand in a non-computer-oriented format, and was transferred to the computer by "mousing" it all in (as is commonly done today by composers who use MIDI sequencers). The computer tools played an important role in the "arrangement" of the original organ piece for a new "ensemble" of voice-like synthetic sounds.

Rule-based Forward-chaining Expert Systems: *Bat out of Hell*

Bat out of Hell: Stories for Dance was realized at the CMRS studio in Salzburg, Austria in 1983 using the "ARA" expert system in Lisp. ARA was developed to implement a specific model of the composition process in a tool that had machine-learning capabilities. It divided the process into the stages of generation of rhythmical patterns, selection of pitch sets and melodic phrase structures, the mapping of compositional properties to timbral control parameters, sound generation, and mixing/spatialization. The ARA tools allowed the composer to define a "language" in each of these domains where a set of adjectives was specified using musical examples, for example, "this rhythmic pattern is red, that other one is blue, and this one is very blue." As the user assigned adjectives and scales to musical material, ARA "learned" which attributes of the material were being influenced. It prompted the user in cases of ambiguity (which were plentiful).

After "teaching" ARA how to classify a set of materials and apply transformations based on user-defined adjectives, the user could then make requests such as "make the spatialization in this section rounder" or "use a smooth orange rhythm here." Due to the technology of the day (twelve years ago), ARA was a non-real-time system that generated Music-11 instrument definition and note-list files. These had to be compiled in "batch" mode. The piece *Bat out of Hell* can be viewed as a transition from order to chaos and back in which ARA started with a well-organized consistent set of material and was then directed to carry out a series of transitions into and out of less-organized areas. The timbre was kept very simple (a single-modulator FM bell sound) for both musical considerations and compute-time constraints.

Logic-based Petri Nets: *Requiem Aeternam Dona Eis*

Requiem Aeternam Dona Eis was realized at the CMRS and PCS GmbH in Munich during1984/85 using the "DoubleTalk" Smalltalk system. DoubleTalk used Petri nets (which are similar to state transition automata) where the states and transition rules were defined in a Prolog-like declarative logic format. Compositional processes were defined as Petri nets with a static structure (the layout of the net), pre-defined state capacities (how many of which kind of token can inhabit a given state at a time), and an initial "marking" (what tokens are where at time zero). There were a suite of interactive editors for "drawing" nets and "programming" their transitions, defining token types, and placing tokens in a net (marking the net). After this was done, one could "run" the net by letting a simulator step through the possible transitions like a rule base. Transition rules could have side-effects such as playing musical notes, or changing the rules of transitions or the marking of the net. DoubleTalk was also initially

connected to a non-real-time software synthesis engine (cmusic), though later versions could generate MIDI output in real time. In *Requiem Aeternam Dona Eis* the DoubleTalk system was used to build musical structures that bear some similarity to the sounds of three sections of the text of the Latin requiem mass. These structures were then the subject of a set of variations that were generated by using the same net and initial marking but changing the transition rules.

Stochastic Procedural Systems: *Day*

Day: Installation was realized at Xerox PARC in 1987 using "EventGenerators" (Pope 1989) in Smalltalk. This is a real-time system that lets the composer/performer interact with a number of mixed procedural/stochastic "smart performers" or "hyper-instruments" that I call "event generators." Examples of simple event generators are chords, trills, and scales; one can, for example, define a chord given its root, type, and inversion, or a trill given its notes, duration and frequency, or a scale given its base, goal, and gamut. Stochastic event generators select at random from a given set of rata or data range, as in the example of a "cloud" that repeats the notes of a given chord according to a given rhythmic pattern. Higher-level procedural event generators use algorithms such as Markov tables or bell peal rules (as in the ringing of the changes) to generate their output. In *Day*, these event generators were configured ahead of time, and the performer could interact with them live. For example, there is one section that is defined as a set of fixed rhythms based on arpeggiated chords, but the actual chord notes are to be played by the performer in real-time.

Grammar-based Rule Systems: *Kombination XI*

Kombination XI is the first movement of *Celebration: Laments and Simple Truths for a Quiet Spiritual Place*. It was realized at the Vienna Music Academy, CCRMA, STEIM, and elsewhere from 1978-90 using "TR-Trees" and the "extended generative theory system" in Smalltalk (Pope 1991b, 1992). This system is loosely based on the seminal work by Fred Lerdahl and Ray Jackendoff on a general-purpose theory of any style of music based on generative linguistics. Their theory defines several kinds of hierarchies, from low-level event grouping up to high-level formal structures based on tension and relaxation (hence the name TR trees). For each of these hierarchies, they define rules for recognizing structures and defining the well-formedness of alternative structures. The TR-Trees system is essentially a set of knowledge bases that implement a specific music theory (my style, as it were), and allows one to use this knowledge in generating (rather than analyzing) music. The expert system engine was the same for all knowledge bases. In *Kombination XI* these rules govern the processing of spoken text as musical material so that a tight symbiosis is achieved between the text and musical structures. Both the musical and text-like sounds in the piece are derived from the sounds of the spoken voice.

Hybrid Multi-tool Environments: *Paragraph 31*

Paragraph 31: All Gates Are Open was realized at the Swedish Institute for Computer Science (SICS), and the EMS Studio in Stockholm during 1992/93 using a wide variety of tools written in Smalltalk and C. My current feeling is that a composer needs to be able to use many different abstractions, notations, and algorithmic tools within any one composition, so I have built a framework that includes all of the components mentioned above, and more. The current hybrid system—implemented within the Musical Object Development Environment, or MODE—includes several kinds of extended graphical notations, a rich set of procedural, stochastic, and algorithmic event generators, the TR-Trees knowledge bases, and a DoubleTalk-like network-based tool. The system can also import data from and export data to a variety of external programs such as MIDI sequencers and real-time digital sound file mixers. This system was used along with various third-party C-language tools in the composition and realization of *All Gates Are Open*, in which the basic structures were described using TR-Trees that generated the specific gestures represented in terms of event generators. These generators were edited using graphical notation tools. The Petri net subsystem was used for the mapping of the actual performance rules that governed the system's "interpretation" of the raw musical data. The final production was described using several Smalltalk- and C-based tools that generated and edited mixing scripts for a real-time digital signal processors at EMS.

Conclusions

My current development effort is aimed at extending the MODE to include still more tool paradigms for new notations, and alternate procedural, stochastic, and knowledge-based algorithmic composition abstractions. My works-in-progress (of which there are several) involve new tools based on context-free generative grammars, other kinds of rule bases, and performance mapping using fuzzy logic.

I strongly believe that, in the future, broad multi-paradigm composition environments will replace the current crop of single-purpose, single-representation tools, and will provide musicians with ever more flexible, scalable, and multi-level facilities for music composition, realization, performance, and production.

References

Buxton, W. et al.,1979. "The Evolution of the SSSP Score Editing Tools." *Computer Music Journal* 3(4): 14-25. Reprinted in C. Roads and J. Strawn, eds. 1985. *Foundations of Computer Music*. Cambridge, Massachusetts: MIT Press.

Loy, D. G., and Abbott, C. 1985. "Programming Languages for Computer Music Synthesis, Performance, and Composition." *ACM Computing Surveys* 17(2): 235-266.

Loy, D. G. 1989. "Composing with Computers—A Survey of Some Compositional Algorithms and Music Programming Languages." in M. V. Mathews and J. R. Pierce, eds. 1989. *Current Directions in Computer Music Research*. MIT Press. pp. 291-396.

Pope, S. T. 1986. "Music Notation and the Representation of Musical Structure and Knowledge." *Perspectives of New Music* 24(2):156-189.

Pope, S. T. 1989. "Modeling Musical Structures as EventGenerators." *Proceedings of the 1989 International Computer Music Conference.*

Pope, S. T. 1991a. "Music Representation, Compositional Methodologies, and Computers." *Proceedings of the Second Symposium on Music and the Sciences*, Seattle: University of Washington.

Pope, S. T. 1991b. "A Tool for Manipulating Expressive and Structural Hierarchies in Music (or: T-R Trees in the MODE: A Tree Editor Based Loosely on Fred's Theory)." *Proceedings of the 1991 International Computer Music Conference.*

Pope, S. T. 1992. "Producing *Kombination XI*: Using Modern Hardware and Software Systems for Composition." *Leonardo Music Journal* 2(1): 23-28.

Pope, S. T. 1993a. "Music Composition and Scoring by Computer." (Invited chapter) in G. Haus, ed. *Computer Music and Digital Audio Series, Volume 9: Music Processing*. A-R Editions. pp. 25-72.

Pope, S. T. 1993b. "Real-Time Performance via User Interfaces to Musical Structures." *INTERFACE* 22(3): 195-212.

Cognitive Modelling in Ethnomusicology: Challenges, Caveats and its Relevance for Computer Music

Marc Leman
University of Ghent
Institute for Psychoacoustics and Electronic Music
Blandijnberg 2
B-9000 GHENT
Belgium
Email: Marc.Leman@rug.ac.be

June 23, 1995

Abstract

A distinction is made between cognitivism and materialism, and between constructivism and ecologism and it is argued that a materialistic and ecological approach may provide an original step towards a characterization of the cultural environment in which music is embedded. The argumentation is based on the notions of causative model and auditory images. The conclusion is that a profound immersion in a culture may allow one to map out the conceptualizations of music at a semantic level, while the modelling approach may allow one to map out the mental representations and cognitive dynamics at the level of auditory images.

1 Introduction

Does computational modelling of music cognition contribute to the study of music as a cultural system? According to the Finish ethnomusicologist P. Moisala, cognitive modelling appears to be in contradiction with the ethnomusicological approach because of its adherence to materialistic epistemology. In reaction to [10] she says [19, p. 9]: "According to materialistic epistemology, it is possible to believe that physiological musical material digitally transmitted into an expert system includes all the information involved in human musical processes. On the other hand, the elimination of cultural and situational contextual factors from the computational modelling is also a natural result of the limitations of such a study method." Accordingly, the relationship between music and culture is problematized in ethnomusicology, and "computer simulations do not fit into the field of "cognitive ethnomusicology" because they either eliminate cultural and contextual influences (symbolism) or do not problematize the interaction between culture and music (subsymbolism)" [19, p. 17].

The claim that cognitive modelling does not fit in ethnomusicology, because it fails to integrate cultural aspects, is unjustified. In this paper we show that cognitive modelling may indeed contribute much to the study of music as a cultural phenomenon. More specifically, we argue that (i) the materialistic epistemology associated with so-called subsymbolic modelling paradigm may provide a step towards a better definition of the cultural environment in which music is embedded and (ii) that such an enterprise may cope well with some old and recent tendencies in ethnomusicological research. The general tenet of this paper is that cognitive modelling provides a challenge to ethnomusicology, which should be understood within the well-defined caveats of an ecolo-cognitive theory of music.

2 Cognitive Issues in Ethnomusicology

To start with, it is useful to make a distinction between the interest in cognition on the one hand, and cognitive modelling on the other. The latter is indeed a rather recent phenomenon associated with the advent of the computer as a tool for simulation. Furthermore, the concept of cognition has different interpretations in ethnomusicology. In general terms, it makes sense to distinguish between:

- cognition as process (related to perception and perceptual learning),
- cognition as conceptualization of world views (related to the musical reflections of myths, nature and society),
- cognition as instrument in observation (related to the emic-etic problem).

Cognition as process was one of the central issues of the early German *comparative musicology*, while both cognition as conceptualization of world views and cognition as instrument in observation was one of the concerns of the anthropological oriented American ethnomusicology.

3 The Use of Computers in Ethnomusicology

According to Nettl [20, p. 391], the effects of technology have been surprisingly small in ethnomusicology. The field is nevertheless growing and the availability of computer techniques for sound processing has been a source for new and interesting developments.

Although Lomax [14] used already computers for classification and statistical analysis, the interest of the ethnomusicological community in computers is more recent phenomenon. Its application, however, is manyfold. Computers have been used for archiving melodies [22], for sound recording and psychoacoustic analysis [15, 2, 3], and in rare cases also for cognitive modelling.

Some interesting attempts to use computer modelling by means of *interactive* field work have been undertaken by J. Kippen and B. Bel. [7] describes a expert system, known as the Bol Processor, that has been used to make a sort of linguistic analysis that is built up during interaction with Indian drum players. In later reports about this project, it was acknowledged that the use of interactive computer modelling is not that simple [8]. "... we succeeded in implanting our own preconceived ideas about musical structure which ultimately biased the analysis".

If cognitive research is indeed making sense in ethnomusicology, then an analysis of the basic assumptions with respect to the current state of the art in cognitive research and the possible role of cognitive modelling is at its place. Would cognitive modelling be of any help in the determination of the physiological-based or cultural-based status of "inherent patterns"? Can cognitive modelling be of any help in clarifying the distinction between musical (cultural determined schemata) and non-musical (purely physiological) perception? Would it be of help in clarifying the notion of schemata? What kind of modelling should be conceived of if one wants to avoid the pitfalls of preconceived ideas? All these questions should be answered with

thorough consideration of past attempts and failures. In what follows, we give an analysis of the basic assumptions of cognitive modelling. Aspects such as representation, dynamics and relationship to the environment are central.

4 Modelling Music Cognition

The above survey shows that cognitive issues in ethnomusicology have been around since the very beginning of this discipline.

In the second half of the 1980ies a revival of the cognition-as-process paradigm can be noticed, based on a renewed psychoacoustical and Gestalt theoretical framework. This is an important development which is related to the availability of new methods (computer simulations) and a more detailed knowledge of the auditory system. Within this context, it makes sense to make a distinction between two developments in cognitive ethnomusicology. One approach is inspired by cognitive anthropology and the idea that a profound immersion in a culture may allow one to map out the conceptualizations of music at a semantic level. The other approach is inspired by cognitive science and the idea that psychological research, combined with cognitive modelling may allow one to map out aspects of the auditory imagery. Both approaches are not exclusive, but the analysis in this and the following sections will focus exclusively on the hard approach. The question is: how far does it go in modeling music cognition, and to what extend can it give an account of cultural influences on music?

Although we intend to explore the consequences of materialist modelling to ethnomusicology in the next sections of this paper, it is instructive to start with a short description of cognitivism—which is often associated with functionalism.

4.1 Cognitivism

Cognitivism has its roots in the philosophy of Descartes, Locke, Hume, and many other philosophers that deal with the human mind and related issues in epistemology. It is based on the idea that cognition has an autonomous status which is independent from the physical carrier (the human brain). Mental states are thereby considered to be *functionally* related to other mental states, irrespective of the actual physical carrier. Fodor [4] has much contributed to the modern version of this philosophical viewpoint. According to his *symbol-based* interpretation of functionalism, music cognition would have a formal nature in that processes, which we call cognitive, are operating on properties of the musical representation. The notion of *representation* is important to understand this position: a representation is something that stands for something else. Like notes in a score are representing the sounds of music, this theory assumes that the mental representation of music is pointing to features and properties of the outer world. Like notes in a score are symbolic entities, it is assumed that the entities of the mental representation are symbols. A typical characteristic of symbols is the separation between form and content: the referent of the symbol is its content, the actual appearance of the symbol is its form. Thus, according to cognitivism, cognitive processes operate on the formal properties of the representation, nl. the *symbols*. Hence the idea that human cognition is symbol-manipulation [21].

In addition to symbol-manipulation, the cognitivists assume an *interpreter* that is operating on the contents of the symbol and that is assigning *meaning* to the representation. The status of this interpreter has never been very clear but this notion is related to rather ill-defined concepts such as "self", "individual" and "intention". To summarize: the cognitivist theory of mind is very similar to an interpreted logical theory, where all inferences are based on formal properties of the representation, and where an interpretator is making a mapping to the outer world.

Much of the work which is done in the field of "Artificial Intelligence applied to Music" underscribes this approach. Music cognition has been modelled in terms of a rule-system that is operating on notes [13, 5, 6, 18, 16]. Other approaches, in which rule-based systems have been replaced by connectionist models, have often adopted a similar position with respect to representation in that the units of the model are assigned a meaning by the programmer of the connectionist model. In Bharucha's model of chord facilitation [1], a connectionist model is specified in which node-units stand for notes, chords and keys. The connections between the different node-units stand for constraints between different types of node-units. In this model the basic units have been "hand-made" in the sense that it was the programmer himself who first attributed meaning to the formal unit in the connectionist network. In a second stage, he assigned the constraints between the units in terms of connection efficacies.

The approach assumes the existence of small units (epistemological atoms) that form the basis for the description of the represented concepts. The method, called "methodological solipsism" [4] is characteristic for the cognitivist: the maker of the model assumes an interaction between the world and the musical mind, but he/she does not feel the need to model this interaction in the praxis of his/her cognitive research. He/she is not denying the existence of the outer world (and adopts realism), but he/she is in between the outer world (or musical environment) and the model. Cognition is believed to be autonomous: it can be modelled independently from the constraints of the environment in which it developed. For methodological reasons, he/she acts as a gateway between model and musical environment. Summarizing, the cognitivist approach has a high degree of freedom in modelling the cultural environment because the interpreter (programmer) acts both as a filter and interpreter: he/she selects the categories, and defines their relationships. The role of *interpreter* is therefore essential.

4.2 Materialism

The alternative position has less degrees of freedom but it allows an alternative account to representation and musical imagery in particular.

The materialist position holds that the autonomous status of cognition—associated with the notion of epistemic atoms and methodological solipsism—does not give a good picture of what cognition is about. It holds that cognition is *causally* related to the physical world by means of an auditory system whose representational states can be associated with the notion of auditory *images*. Rather than *pointing* to objects in the musical environment, images are *reflecting* properties of the physical signals. By *learning processes*, invariant information in the images are extracted and stored in more stable knowledge structures, which are called *schemata*. Auditory *grouping* at the peripheral level segregates the acoustical input into streams, leading to images of "sonorous objects". Concepts at higher levels are thus formed by self-organization and association. In this conception, the causal chain from environmental stimulus to auditory image is critical. Materialism assumes that these is no reason to assume that the images have a status which is different from physical systems. Even a schema is a mere response structure which is itself to be described as a physical system. In order to know the structure of the schema, one must test the responses of the organisms with signals of the environment. Such an organisms can be modelled provided that also the environment is modelled. As such, finding out the response structure of the model is similar to the way in which mental representations are recorded from humans. This is done by presenting signals and recording the responses. The responses are then analysed and particular structures can be inferred. The materialist position assumes that there is some sort of isomorphism between the schema-responses or responses of knowledge-structures to stimuli from the environment, and brain states. In the context of this paper, however, it is not necessary to go deeper into this discussion (See e.g. [23]).

5 Modelling in relation to the cultural environment

The difference is best understood by looking at how both approaches are dealing with the musical environment:

- In the *constructivist* approach, the environment is assumed to be independent from the model, and the connection with the environment is mediated through the programmer.
- In the *ecologist* approach, the environment is taken to be an integral part of the model.

5.1 Constructivist Modelling

In constructivist modelling, the environment is not necessarily restricted to a purely musical environment. As the programmer is mediating between the model and the environment, he/she is responsible for the categorical selection. To some extent this situation reflects the etic position in the context of antropological studies. Applied to modelling, however, the position becomes explicit in that a limited set of well-defined categories must be specified *a priori*. This attitude is similar to Lomax's coding technique. In Lomax's study [14], the listener has to rate some 37 parameters, among which the "overall rhythmic scheme (one-beat, simple meter, complex meter, irregular meter, free rhythm,...)". The listener has to choose a value for the parameter which is then translated into a number. Apart from the conceptualization of sound, cultural premises, including religious belief, conception of space and time, can be associated with particular units of a model, and it is furthermore possible to define correlations between these units and a representation of the musical environment (pitch-distribution in time, timbre, rhythm). The constructivist approach thus allows flexibility in modelling: nearly anything can be modelled provided that the appropriate categories are related to each other. It is up to the programmer to interpret the input as well as the output to the model.

5.2 Ecologist Modelling

Compared to constructivist modelling, the ecologist approach seems to be far more restricted in that the cognitive model depends on the representation of an environment. In modelling terms, this means that the computer environment needs to reflect the real environment as good as possible. The use of apriori categories and symbolic representation is thus not appropriate, categorical structures are developing by self-organization on the basis of exposure to the environment. In a former paper [9], we have related this viewpoint to a so-called *psychomorphological* approach of music cognition. It assumes that the cognitive processes are based on an adaptation of the brain structures to the environment both ontogenetically (neural architecture has been developed on an evolutionary basis) and philogenetically (fine-tuning of the neural architecture is achieved by exposure to the environment).

In what follows, we discuss a framework in which the materialistic and ecological viewpoint is related to the study of cultural aspects. What we want to exclude in modelling are:

- the use of *apriori* categories,
- the distinction between *form* and *content* of a representation,
- the *solipsist* attitude in modelling, ...

The approach is based on:

- causative models
- direct perception
- perceptual learning by self-organization

- context-dependent processing of information

In the absence of an overall multi-modal model of the human senses, this approach implies a number of limitations. The most severe limitation is that the current causative models are restricted to only one input-modality which is the auditory input modality.

6 A Framework for Modelling in Cognitive Ethnomusicology

Given the constraint of mono-modality, it is clear that any answer to the question whether computer modelling can say something about non-musical cultural influences (religion, world view, action patterns) will be indirect. It is then useful to decompose the modelling environment into the following levels:

- The Human Information Processing System. Similar to the laws of physics, we may assume that the causative input systems (auditory system, visual system, olfactive system) as well as the brain dynamics which carries the representation of the outer world at higher cognitive levels are universal and thus in principle the same for the members of different cultures. The following levels are then distinguished:
  - Sensory Information Processing: eyes, ears, and other sense organs are the same for the members of all cultures.
  - Perceptual Information Processing: the *dynamic principles* which underlay grouping and perceptual mapping are the same for the members for all cultures.
  - Cognitive Information Processing: the *dynamic principles* which underlay perceptual learning, categorization and association are the same for the members of all cultures.
- The Physical Environment. The nature of the acoustical environment can be described at the level of the waveforms. This is the same for all cultures. The properties of the waveform can be studied at a purely physical level, e.g. by making Fourier analysis or any kind of signal manipulation technique. The physical environment is also identical for all cultures over the world and the choice of representation can be related to universal human constraints. For example, a sampling rate of 44.1 kHz and 16 bit resolution is sufficient to capture all relevant signals of the musical domain within the computer environment.
- The Musical Environment. From the present mono-modal point of view, the musical environment is made of acoustical signals. The acoustical analysis of these signals may show differences between cultures. For example, in some cultures the musical environment is based on instruments with inharmonic overtones, while in other cultures, only harmonic instruments are used. Some cultures use polyphonic settings, while other cultures use monophonic settings, and so on... Obviously, these differences will be reflected at the level of the auditory images, and these images may ultimately lead to quite different categorical schemata.

  The question of what kind of musical environment should be included into the model varies from culture to culture and it will ultimately depend upon the choice made by the programmer. The obvious choice is to represent the musical environment as accurate as possible. The fact that tones, played in a particular succession, or at the same time, contribute to a context, may be an important factor. The perception of that context may be a central cue for meaning formation in a particular culture. Other applications may center on isolated aspects of the musical environment, such as tones [2].

In the ecological approach, the selection of the musical environment is an intricate question which should be considered with great care. But assume that a representative selection of music can be made, how does the model relate to influences of the non-musical cultural environment? It is clear than an answer to this questions will be indirect: by showing which aspects are determined by auditory mechanisms it is possible to obtain information about aspects that are non-acoustical as well. For example, by means of learning models, one may show that perceptual categories are emerging from the acoustical environment and that they correspond with psychological categories (e.g. obtained by means of cognitive structuralist methods) [11]. By relying on principles of auditory grouping, one may show the emergence of musical objects at the level of auditory imagery [24]. Such categories can be called cultural (since they are determined by the musical environment) but they are inherently acoustical because they dependent on (a) acoustical properties and (b) properties of the auditory system. By showing that such a categorization process is possible at a purely auditory level, one obtains an indirect argument that certain characteristic properties are not non-musical–in other words: that they are constrained by the feedback of auditory principles. Typical examples are the use of scales and rhythm patterns: their properties are largely dependent on the use of particular instruments and the constraints of auditory perception.

Of course, one may argue that the use of this or that particular instrument is determined by a religious influence, so that the categorization process is ultimately non-musical. We believe that such an argument misses the point in that the development of the musical environment is always also non-musical. What constitutes the musical environment is dependent on a historical development in which the non-musical influences have found an acoustical sediment. The present approach is the first to acknowledge such influences. The relationship between auditory and non-auditory constraints are often intricate, but that is exactly the reason why this materialistic and ecological approach is a well-come addition to the antropological approach: musical conceptualization is not limited to semantic categories, but involves a number of constraints that are imposed by auditory principles.

7 Ethnomusicology and Music Research

The relevance to computer music is that ethnomusicology may provide keys to build up cognitive reference frames that can be used in music research as tools for analysis and composition. Of particular relevance is the idea that the musical environment in different cultures has lead to schemata or cognitive representations which are very different form those of the Western culture (in which harmonic tonality plays an important role). With computer tools, it is now possible to explore new musical environments and schemata which may have an appeal that is founded in perceptive and cognitive principles. Once such a cognitive frame is known, it can be used as a generating device or as a control mechanism for feedback in interactive computer systems.

8 Conclusion

In this paper we have explored the challenges and caveats of cognitive modelling in ethnomusicology. Contrary to the claim that the materialist epistemology is neglecting the role of cultural and situational contextual factors in music, we found that this approach may provide fruitful insights in the way in which people of different cultures have organized their musical imagery. Acoustical contextual dependencies are well preserved in acoustical recordings and although this mono-modal representation implies a limitation of the inter-modal musical premises (including dance or self-movement and visual aspects), the proposed approach makes it nevertheless possible to clarify some intricate questions of musical performances in different cultures.

References

[1] J. Bharucha. Music cognition and perceptual facilitation: a connectionist framework. *Music Perception*, 5(1):1–30, 1987.

[2] W.A. Deutsch and F. Födermayr. Tonhöhe versus Frequenz–zur Frage der indonesischen Tonsysteme. *Musicologica Austriaca*, (6):25–33, 1986.

[3] F. Födermayr and W.A. Deutsch. Musik als Geistes- und Naturwissenschaftliches Problem. In W. Gratzer und A. Lindmayr, editor, *De Editione Musices-Festschrift Gerhard Croll zum 65. Geburtstag.* Laaber-Verlag, Wien, 1992.

[4] J. Fodor. *Representations.* The MIT Press, Cambridge, MA, 1981.

[5] J. Jones, D. Scarborough, and B. Miller. An expert system for music perception. In Balaban et al., editor, *Proceedings of the first workshop on artificial intelligence and music*, Minneapolis/St. Paul, 1988. AAAI-88.

[6] J. A. Jones, B. O. Miller, and D. L. Scarborough. A rule-based expert system for music perception. *Behavior Research Methods, Instruments, and Computers*, 20(2):255–262, 1988.

[7] J. Kippen. An ethnomusicological approach to the analysis of musical cognition. *Music Perception*, 5(2):173–196, 1987.

[8] J. Kippen. Music and the computer - some anthropological considerations. *Interface*, 21(3-4):257–262, 1992.

[9] M. Leman. The theory of tone semantics: concept, foundation, and application. *Minds and Machines*, 2(4):345–363, 1992.

[10] M. Leman. Symbolic and subsymbolic description of music. In G. Haus, editor, *Music Processing.* Oxford University Press, A-R Editions, Madison, Wisconsin, 1993.

[11] M. Leman. *Music and Schema Theory - Cognitive Foundations of Systematic Musicology.* Springer-Verlag, Berlin, Heidelberg, 1995.

[12] M. Leman. A survey of subsymbolic paradigms in music research, 1995.

[13] F. Lerdahl and R. Jackendoff. *A generative theory of tonal music.* The MIT Press, Cambridge, MA, 1983.

[14] A. Lomax. *Folk Song Style and Culture.* Transaction Books, New Brunswick, New Jersey, 1968.

[15] E.H. Lubej. Die Stimmgebung in den Gesängen der "tenores" aus sardinien. In E.Th. Hilscher and T. Antonicek, editors, *Beiträge zu Methode und Problematik der systematischen, ethnologischen und historisschen Musikwissenschaft.* Verlegt bei Hans Schneider, Tutzing, 1994.

[16] A. Marsden and A. Pople, editors. *Computer representations and models in music.* Academic Press, London, 1991.

[17] S. McAdams. Recognition of sound sources and events. In S. McAdams and E. Bigand, editors, *Thinking in Sound–the Cognitive Psychology of Human Audition.* Oxford University Press, Oxford, 1993.

[18] B. O. Miller, D. L. Scarborough, and J. A. Jones. On the perception of meter. In M. Balaban, K. Ebcioglu, and O. Laske, editors, *Understanding music with AI: perspectives on music cognition.* The AAAI Press, Cambridge, MA, 1992.

[19] P. Moisala. Cognitive study of music as culture–basic premises for cognitive ethnomusicology. *Journal of New Music Research*, 24(1):8–20, 1995.

[20] B. Nettl. Recent directions in ethnomusicology. In H. Myers, editor, *Ethnomusicology—An Introduction.* W.W. Norton & Company, New York, 1992.

[21] A. Newell. Physical symbol systems. *Cognitive Science*, 4:135–183, 1980.

[22] H. Schaffrath. The retrieval of monophonic melodies and their variants: Concepts and strategies for computer-aided analysis. In A. Marsden and A. Pople, editors, *Computer representations and models in music.* Academic Press, London, 1992.

[23] E. Scheerer. Psychoneural isomorphism: historical background and current relevance. *Philosophical Psychology*, 7(2):183–210, 1994.

[24] U. Wegner. Cognitive aspects of amadinda xylophone music from buganda—inherent patterns reconsidered. *Ethnomusicology*, 37(2):201–241, 1993.

MaxAnnealing: A Tool for Algorithmic Composition Based on Simulated Annealing

Fernando Iazzetta
Center for New Music and Audio Technologies
University of California at Berkeley
1750 Arch St. - Berkeley, CA - 94709
also from
PUC-SP - Communication and Semiotic Program
fernando@CNMAT.berkeley.edu

Fabio Kon
Department of Computer Science
Institute of Mathematics and Statistics
University of São Paulo
Cx. Postal 66281 - 05389-970
05389-970 - São Paulo (SP) Brazil
kon@ime.usp.br

Abstract

Musical composition can be roughly viewed as a search for the best solution among a finite - although huge - universe of possibilities. Some of the algorithmic compositional techniques try to simulate the act of composing doing this search automatically. However, this approach has two major problems. The first one is the hardness of depicting aesthetic concepts through mathematical rules. The second problem is the low efficiency of the exhaustive search among all possible solutions.

The "Simulated Annealing" algorithm - first proposed in [1] - presents very good results on finding the optimal solution for many combinatorial problems efficiently (in polynomial time). In this paper we present an adaptation of this algorithm to the problem of algorithmic composition. We then discuss some possibilities regarding goal functions to this algorithm and describe MaxAnnealing, a tool designed to help composers and musicologists study the possibility of defining aesthetic concepts through mathematical rules. The system is implemented in the MAX programming environment.

1 Introduction

Musical composition involves many interrelated parameters which interact with each other creating a complex structure of musical possibilities and constraints. As Minsk has said "the problem of creating a good piece of music is a problem of finding a structure that satisfies a lot of different constraints" [2]. To deal with this complexity the composer uses different strategies, some of which are very well defined and can easily be formalized while others remain so flexible and context dependent that resist to any kind of formalization. These strategies include intuition, chance, adaptation, and trial and error based choices. However when the compositional task is transferred to a computer, these strategies are not always available since they are hard to implement as a computer program.

Usually, the compositional task consists of defining a musical goal to be reached by a compositional project. In most cases, not only the compositional project is determined by the goal, but also the latter is sensitive to the constraints imposed by the former. Thus, music arises from the balance of the composer's intentions (goal) and the compelling processes of organizing musical ideas (project). In other words, composition is a constant exercise of adaptation and interaction between project and goal.

However, in the case of computer generated compositions, sometimes it is hard to come up with an efficient project which can be translated into a compositional algorithm. Although the composer can have a clear idea of his musical intentions, he may be incapable of formalizing these ideas in order to build a program. Actually, for certain kinds of musical problems it is hard, or even impossible, to delineate an algorithm to solve them using the contemporary Computer Science tools. Usually, it can be due to the diversity of elements involved in the problem, to the complexity of parameters that affect the problem, or even to the lack of knowledge about some aspects of the system.

In his piece *Protocol* for solo piano, Charles Ames [3] proposes a compositional method that goes beyond the traditional processes strictly based on random selection or rigid determinism used in previous computer music programs. Ames formalizes his ideas in a "protocol" which is a "collection of tests where each test has been ranked according to one's preferences [...] Then by having the computer evaluate a substantial repertory of alternatives, one can direct it to search for the alternative best fitting these criteria. If there is a choice passing all tests, then systematic evaluation must find it; otherwise, the search will provide the best of imperfect choices" [3, p. 215]. The caveat of this method is that the computer must evaluate all in an enormous range of possible musical configurations to find the best one.

Another problem involving an extensive search of compositional solutions is presented by Schottstaedt in his "Automata Counterpoint" [4]. Schottstaedt implements a program that generates five species of counterpoint based on the rules exposed by J.J. Fux in the Gradus ad Parnassum (1725). Although the rules which govern the counterpoint are very clear and well defined, there are many different solutions for the same counterpoint problem and the computational time to check each possible solution becomes a significant constraint. "If we make an exhaustive search of every possible branch of a short (10-note) first-species problem, we have 16 raised to the 10th power possible solutions (there are 16 ways to move from the current note to the next note). Even if we could check each branch in a nanosecond, an exhaustive search in this extremely simple case would take 1,000 sec (about 20 minutes)" [4, p. 203]. The solution Schottstaedt presented to this case was to start with the best first result found for each interval, that is, the one to which the program assigns the smallest penalty. But as Schottstaedt recognizes, "the first such solution may not be very good. By accepting the smallest local penalty we risk falling into a bad overall pathway" [4, p. 203].

Some computational tools have been applied in music in order to solve this kind of problem where it is necessary to find the best configuration in a large space of possibilities. Examples are the back-propagation training algorithms used to weight the connections in a neural network, or the genetic algorithms which, inspired in the selective processes that occur in a natural environment, apply successive transformations to a system that evolves toward its environmental fitness. Those tools are based on the search of the best solution of a problem through iterative processes. In these processes, a possible solution is generated and evaluated by a particular function. Then, the program generates a new possible answer which again is evaluated. The results of these successive "guesses" are compared in order to direct the search for the best result. After doing a number of these iterations the system can reach a solution which is sufficiently close to the optimal.

2 The Simulated Annealing Algorithm

Simulated annealing (SA) is a probabilistic algorithm used in the search for optimum solution first described in [1]. It has been developed after the analogy with a Condensed Matter Physics thermal process named annealing and can be used to find near optimal configurations in very large and complex systems.

The annealing process consists on heating a piece of metal until it reaches a temperature slightly above its critical homogenization temperature and then carefully decreasing the temperature until the molecules are arranged in a way so that the metal reaches its thermodynamic equilibrium. The thermodynamic equilibrium is the state in which the molecules form a structure strictly organized and the energy of the system is minimal. If the heating and cooling processes are not correctly done, the metal does not reach the thermodynamic equilibrium.

As a combinatorial optimization process, the purpose of the SA is to find the minimum or maximum values of a cost function for a specific system. The cost function or goal function measures how good a specific configuration is. The SA starts with an initial structure S which can be randomly generated and has a method for modifying this structure generating a neighbor structure. For each step in this process, the quality of the new structure is determined and, if the neighbor solution S' is better than the current solution, then S' becomes the current solution. But if the new solution does not represent an improvement in the goal function, it still can be accepted with probability

$$e^{\frac{Quality(S')-Quality(S)}{T}}$$

This condition, known as Metropolis criterion, helps the SA algorithm to escape from local minima. Unlike traditional local search algorithms, SA can make occasional moves in the search space which can decrease the value given by the goal function *Quality*(). The probability of acceptance of the new structure is greater if the difference between the cost of S and S' is small, even if it would represent a decrease in quality. Also, the probability of acceptance of a structure that decreases quality gets smaller as the temperature T decreases providing that the algorithm gets stabilized under a certain temperature.

It is possible to demonstrate that if the goal function and temperature lowering functions meet some constrains, the SA ends its running in polynomial time and finds the optimal solution with probability almost one [6].

3 Applying SA to Musical Composition

Figure 1 presents a version of the algorithm applied to the musical composition problem.

```
Procedure Simulated Annealing
Begin
  S ← random initial song
  T ← 1 /* initial temperature*/
  L ← L0
  While (Quality is increasing) do
        Repeat L times
              S' ← Neighbor(S)
              If Quality(S') > Quality(S) Then S ← S'
              Else S ← S' with probability e^((Quality(S')-Quality(S))/T)
        T ← T × 0.9 /* decreases the temperature */
        L ← reduce(L)
  X ← S, the final song given by the algorithm
End.
```

Figure 1: *Simulated Annealing Algorithm*

3.1 Basic Data Structure

Each configuration of the solution space is called a "song" and its structure is defined by the following C language declarations.

```
typedef struct { char pitch, velocity, start; } note;
typedef note song[MaxVoices][MaxTimeUnit];
```

Thus, we see a song as an array of `MaxVoices` rows and `MaxTimeUnit` columns where each entry is a triple (pitch, velocity, start). Pitch may contain either a MIDI pitch value (from 0 to 127) or a rest (represented by -1). Velocity contains a MIDI velocity value (from 0 to 127). Start may contain either 0 or 1 representing that there is a note or rest starting at that time unit (1) or not (0).

3.2 Neighbor Function

The job of the neighbor function in the SA algorithm is to receive a configuration in the solution space and to return another configuration in the same space being slightly different from the first one.

Our implementation of this function returns a song identical to the one it receives except for one note which is randomly added.

3.3 Goal Function

The purpose of our work is to develop an algorithmic tool to be used by composers and musicologists. Our idea is to offer a simple way for composers or musicologists with programming experience write their own goal functions in C, link them to our system, and then use the resource created as a MAX external object. Any goal function which receives a song as described above and returns an evaluation for its quality can be linked to our basic system.

For those with no programming experience, we have written a goal function which can have some of its parameters defined through the cells and tables of a friendly MAX patch. In the next section we describe the implementation of this goal function which can be taken as an example by other people interested on using our system with their own goal functions.

4 Implementation of MaxAnnealing

We have created a program called MaxAnnealing which uses the SA algorithm in order to find the optimal solution for a compositional problem. The program was implemented in the OPCODE's MAX 2.5.2 environment [5] since MAX offers a series of handful tools for manipulating all the music and graphic data needed by the program. An external MAX object called **annealing** which performs the optimization was created using the ThinkC 5.0.3 compiler.

The program has three basic modules: 1) the parameters interface, which allows the user to set the parameters that will be used for the song evaluations ; 2) the simulated annealing object, which performs the search for the optimal solution; 3) the player, which receives the song produced by the annealing algorithm and plays it through a MIDI output.

The current implementation of MaxAnnealing generates a song of sixteen bars, each bar consisting of four beats, each of which having up to four subdivisions, which we call "time units". The piece is distributed by four different tracks or voices which can be assigned to four different MIDI channels. All these values can be changed through modifying the annealing object source code.

4.1 The Parameters Interface

The program starts with the user providing tree general parameters for the piece. The first one is a tension curve which determines the pitch tension at each time unit of the piece. The pitch tension measures how dissonant are the simultaneous intervals that occur at each point of the music. This parameter is set by drawing a curve where each point represents the tension for each time unit in the piece as shown in Figure 2.

A second curve determines the density parameter. In this case, each point in the curve corresponds to the number of notes that should sound at each beat (group of four time units). The range can vary from 0 (no sound at all) to 16 (4 time units × 4 voices).

Finally, the user assigns weights to every pitch class of the chromatic scale. Here, a weight 0 means that the note should never occur. This brings a generic character in the pitch domain since the user can determine which scales he wish to use and create a pitch hierarchy by assigning high weights to certain pitches and low weights to others.

4.2 The Simulated Annealing Object

All data set by the user is then given to the simulated annealing module which starts the optimization process. It begins with a piece of music composed by just one arbitrary four-note chord where the voices are distributed from the highest to the lowest note through the tracks one to four (any other initial song

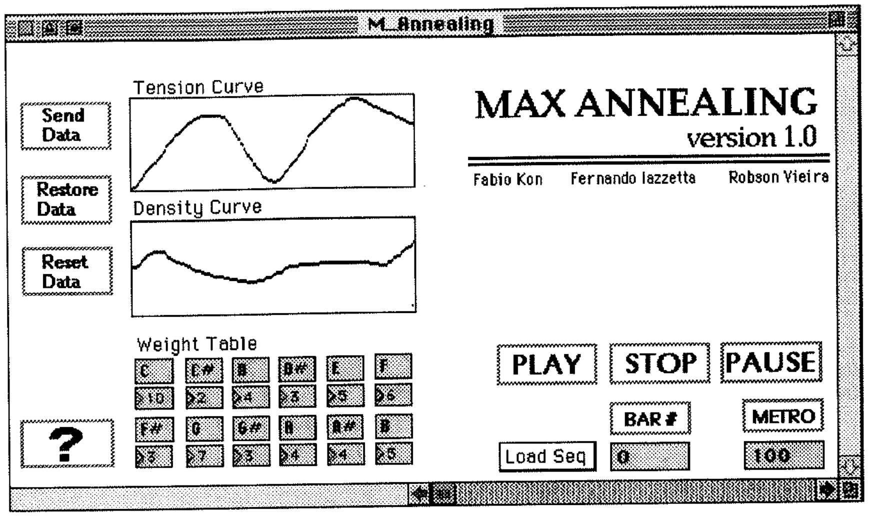


Figure 2: The MaxAnnealing Patch

would fit here). This is the first song evaluated by the algorithm. Given a song, its quality is evaluated by the goal function which is the weighted average of the five following criteria.

1. For each time unit, it calculates the tension among the intervals generated according to a previously established table of dissonance. To calculate the tension, MIDI notes are transformed in pitch classes and then the algorithm verifies all intervals that occur among the four voices for each time unit. These intervals are translated into previously established values which represent a dissonance level. The pitch tension for each time unit is calculated by adding these values.

   The closer is the pitch tension to the tension determined by the user for that time unit, the higher is its evaluation. The final evaluation for the whole song is the average of its time unit evaluations.

2. For each bar, it calculates how many notes would sound and compares the result with the values established by the density curve giving a higher evaluation for the bar if the result gets closer to the value established by the user.

3. For each of the four voices, it evaluates the leaps between each note and its antecedent. Leaps close to a minor third receive higher evaluation than bigger and smaller ones. This will assure that the melodic contour of each voice will not have too many large leaps nor too many small intervals.

4. The more crossings between voices a song has, the lower is its evaluation. By doing so we try to avoid too many voice crossings.

5. Lower pitches receive higher evaluation if they have longer durations than higher pitches. This would lead to a music structure in which lower pitches will be associated to long durations and higher pitches will be associated to short durations.

As we have seen in section 2 and 3 the value given by the goal function is compared with the quality of a previous song. Then the program decides if it takes the new song as the new temporary solution, or if it keeps the last value.

4.3 The Player

This is a simple module which receives the best result obtained by the simulated annealing and translates this data so it can be played as a four-channel MIDI sequence. It presents some standard control buttons including play, pause, stop, metro, and save and load sequence.

5 Results

We have run MaxAnnealing for several times, using different parameters to test its performance. Initially, we have set the program to search through an average of 9,200 pieces of music for each new parameter configuration. This process took about 30 seconds in a shared SUN SPARCserver 1000 and one minute in a Macintosh Performa 630. Further tests have shown that, in some cases, MaxAnnealing was able to find very good solutions after a search through only 2,000 songs, which lowers considerably the necessary time to run the program. It is worth noting that even searching through 9,200 different songs, the algorithm runs almost 2,000 times faster than that which searches through all the 2^{24} possible solutions for our particular problem.

For each test we have set different parameters in order to generate specific kinds of musical output. By assigning certain weights to the notes in the pitch table we were able to generate pieces of music based on different modes and scales. The curves also provided an easy method of control over the tension and density of events which happened at each point of the song. After only a few experiments with these controls we could make satisfactory predictions about the general characteristics of the music MaxAnnealing was going to generate as the best solution.

Although our intention was only to demonstrate the validity of using the SA technique in the solution of musical problems, and despite the simplicity of the compositional rules applied, MaxAnnealing has produced some interesting musical results.

6 Conclusion and Further Work

We have introduced the use of simulated annealing algorithm as a powerful tool in the fields of musical composition and musicology. The Simulated Annealing has shown to be very effective in the search for a satisfactory solution for problems involving a large number of possible musical configurations in a very reduced time span.

With modifications in its objective function, the system would be very useful for finding the solution of many other compositional problems. Moreover, one can conceive the utilization of the SA as a practical tool in the field of musicology, which would enable the verification of relations between the formal rules which govern a piece of music and the actual effects - in terms of auditory experience - those rules generate.

As a further step on this work, we intend to test the use of other objective functions and develop the MaxAnnealing user interface to allow the generation of more complex compositional systems, which, we believe, will lead to more interesting musical results. These developments include new configuration options to be set by the user and the introduction of more elaborated compositional constraints in the program functions.

The MaxAnnealing source code and binaries are available by anonymous FTP at ftp.ime.usp.br, directory pub/macintosh/MaxAnnealing .

7 Acknowledgments

The authors gratefully acknowledge the help provided by Robson Feichas Vieira (IME/USP) throughout the development of the computational system.

This work was supported by CNPq (process # 200124/94-3), FAPESP (process # 93/0603-1), and CNMAT.

References

[1] Kirkpatrick S., C.D. Gelatt Jr. and M.P. Vecchi. "Optimization by simulated annealng", *Science*, 220, pp. 671-680, 1983.

[2] Roads, Curtis, "Interview with Marvin Minsk", *Computer Music Journal*, 4 (3), 1980.

[3] Ames, Charles, "Protocol: Motivation, Design, and Production of a Composition for Solo Piano", Interface, 11, pp. 213-238, 1982.

[4] Schottstaedt, W. "Automatic Counterpoint", *Currents Directions in Computer Music Research*, Max Mathews and John Pierce (Eds.). Cambridge: The MIT Press, 1989.

[5] Puckett, M., D. Zicarelli, *MAX - An Interactive Graphic Programming Environment*, Opcode Systems, Menlo Park, CA, 1990.

[6] Mitra, D., Romeo, F., Sangiovanni-Vicentinelli, Alberto, "Convergence and Finite-Time Behavior of Simulated Annealing", *Advanced Applied Probability*, 18, pp. 747-771, 1986.

AREM2: a composition tool

Francisco Kröpfl, Miguel Calzón
Laboratorio de Investigación y Producción Musical (LIPM)
Junín 1930 - (1113) Buenos Aires, Argentina
fk@morfo.filo.uba.ar
mcalzon@cnea.edu.ar

Abstract

In recent years an interactive software named AREM the goal of which is to produce reinterpretation in real time of sequences input through a MIDI interface was developed. This software is based on a specific composition methodology by Francisco Kröpfl implemented by Miguel Calzón. AREM2, a non interactive program, takes the same starting point and it is oriented towards the organization of musical structures including pitch, rhythm, syntactic units and texture. Both programs are complementary and may be used simultaneously.

1. Introduction

This paper introduces AREM 2, a program that creates musical structures which are reinterpreted successively with regard to their qualities of pitch, rhythm, texture and syntax. It is based on a composition methodology designed and taught by Francisco Kröpfl (described in Kröpfl 1987).

This paper will cover three areas: firstly the relationship between AREM1 and AREM2, secondly the way in which AREM2 produces musical sequences, and thirdly how both programs are applied in different fields of music.

2. AREM 2 and its precursor AREM 1

AREM - *Algoritmos para la Reinterpretacion de Estructuras Musicales* - **or ARMS** - *Algorhythms for the Reinterpretation of Musical Structures* (described in Calzón 1992 and 1993) is an interactive program which allows the transformation of the structure of the musical input by a performer or composer. From now on we will refer to this program as **AREM 1.**

AREM 2, by contrast, is **not** interactive. It consists of a group of patches that produce musical sequences following patterns derived from the subjacent methodology of composition. Each patch generates information on pitch, intensity or duration of notes, while the articulation remains unaltered. Each patch is activated by either the computer keyboard or by MIDI switches. The modularity within the program allows for the programming of new generators according to the needs of the user. The generators can operate successively or simultaneously, as well as working while AREM 1 is running.

AREM 1 and **AREM 2** complement each other. The idea behind writing this second program was to develop more complex structures than those that an instrument performer is able to carry out, as well as a tailored computer aid for the design of structures. Because of the modularity of its conception, the user can personalize the program within the possibilities allowed by the underlying method of composition, by rewriting the tables the usage of which is described below.

3. Groups and phrasing in AREM 2

As far as syntax is concerned, **AREM 2** offers at least two levels of generation: the group (a sequence of few notes whose construction function is to act as motif cells) and the phrase (a sequence of several groups). The group is a motif element that controls micromodes, pitch networks and accent prototypes. The separating and linking factors relate the groups to each other.

In the version of **AREM 2** we are describing there are five modules:
1) Sequences composed of groups of four elements.
2) Sequences composed of groups of two, three and four elements.
3) Uniform rhythm in an ample pitch register: uniform durations within each group, which produce the neutralization of the agogic factor; the scope of the pitch register is over an octave and a half.
4) Uniform rhythms in a narrow pitch register: the same usage of the uniform field of durations as explained in the case above but in a pitch register not wider than two octaves at the most.
5) Groups whose trajectory show a strong direction (melodic arc) produced by a periodic variation in their translation values that positions the groups in the pitch register.

Each group uses tables containing a selection of micromode permutations to determine the relevant pitches, which are arrived at by either random selection or by following predetermined rules for their sequence. The agogic or dynamic accentuation is arrived at in much the same way, by selection of the accentual prototypes which are applied to the relative duration and intensity. The restrictions or laxity allowed the elements of the three tables will have important consequences in the ensuing phrases.

4. Separation and Linkage factors

Each group is articulated to the others by global registers that constitute the factors of separation and linkage which are responsible for their positioning within the register of the groups. The factors involved are:
1) Pitch register: a constant additive in the pitch field determines a translation of the tabulated interval structures (simple and complex micromodes).

2) Intensity register: a different additive constant in the intensity field carries the relative intensity values (agogic rhythm).
3) Speed register: a multiplying constant modifies the the relative duration values (agogic rhythm) varying the tempo or speed of the groups by contracting or dilating their durations. The numeric values which control these factors are selected randomly within a calibrated acceptable range.

5. Texture and Syntax

As occurs with AREM 1, this program also produces reinterpretations of texture and syntax.
As far as texture is concerned, the program produces monody (one active module) and polyphony (several active modules). It is also possible to create chords, the function of which is to add a new factor of accentuation by increasing the vertical density.
With regard to syntax, the program is able to control the reiterative or evolutionary nature of a structure according to which values have been selected from the tables that determine the material in each group.

6. Applications

The program may be used to assist in the composition of music, inasmuch as it enables the user to audibly check a priori configurations referring to pitch, rhythm, texture and syntax. Because of its complementary nature with AREM 1, it is applicable to live electronics. It can also be applied to assist in music teaching.

7.Aknowledgments

We would like to thank to Lorraine Smith for her help in preparing the English version of this paper.

8. Bibliography

Calzón, M.(1992). A program for the reinterpretation of musical input structures. *Proceedings of the International Computer Music Conference 1992 - San Jose*. The International Computer Music Association.
Calzón, M.(1992). AREM / ARMS - Reinterpretation of musical structures. *Proceedings of the International Computer Music Conference 1993 - Tokio*. The International Computer Music Association.
Kröpfl, F. (1987). Methodology of rhythm analysis. *National Convention of Musicology*, Buenos Aires, 1986. Centro Cultural Recoleta Edition, Buenos Aires, 1987.

SINAPSIS: A self-generating system of musical discourses.

Eleazar Garzón

Centro de Investigaciones de la Facultad de Filosofía y Humanidades (CIFFyH) de la Universidad Nacional de Córdoba.

Manuel Carles 3280. Poeta Lugones. Córdoba 5008 ARGENTINA.

ABSTRACT

SINAPSIS is a composition software whose more significant feature is its capability of "INTERACTION" with the user at the level of production of "Musical Discourses". For this purpouse, a setting similar to that of staging has been designed. On this basis, the system provides two structures: ACTORS and STAGE DIRECTORS.

By grouping and relating actors in networks, by asigning them to stage directors, and by overlaping and/or juxtaposing these latter, the user creates the structural plan of the work over which the system will generate (with each performance) a particular discourse (a SPECIMEN belonging to the universe of probabilities that such a model admits).

INTRODUCTION

The SINAPSIS programme appears in the same instant of creating a system of composition by means of a computer where this should have the decision in the preparation of the musical Discourse, but at the same time

this intervention should be userful in the development of the sonorous action in terms of UNITY AND DYNAMIC PROCESS of the discourse.

COMPOSITIONAL MODEL

Taking into account what has been expressed before a hypothesis of work was stated considering the development of a system simulating an analogous atmosphere to the one of a setting on the stage. In order to obtain this, two entities were created; the ACTORS and the DIRECTORS OF SCENE - the first ones are considered time sonorous structures and variables that determine potential margins of behaviour of themselves.

The second ones are in charge of selecting ACTORS, determine their behaviours (within the rank the actor admits) and conduct the sonorous flow.

ACTORS AND DIRECTORS OF SCENES

An ACTOR is:

- A list of temporary data and of MIDI messages.

- A set of variables that fix limits and characteristics of behaviour to the actor in:

- Time:

- Establishing if the temporary relationships between events should be kept constant or should admit modifications.

- Or determining if the speed in which the actor is played, will be altered in an increasing or decreasing way.

- Notes:

- Fixing the scale in which the actor is played. (Diatonic, pentatonic, chromatic, etc.).

- Establishing the measured transpose capacity in terms of interval (ascending and descending energy jump).

- And sonority: Determining the potential rank of modification of the amplitude for each actor's event.

It also has other properties:

- Behaving like mutant in the notes plane and timbre (procedure of cumulative selection).

- Establishing contacts with other actors or their own field.

- Autoinserting other lists.

With the capacity of mutating an actor as the possibility of establishing contacts among them (Markov Chaining), the system provides the user with the resource of a mechanism with theme evolution.

A DIRECTOR OF SCENE is a SELECTOR of ACTORS, their MODULATOR and the DIRECTOR of the temporary flow (sonorous action).

As selector he performs a recurrent activity calling the same actor or a different one each time, belonging to an associated field to this director.

As Modulator his activity consists of:

- Awakening the actor's behaviour concerning temporary in tone (transpose) and sonority, deciding what magnitude should be applied on each stage.

- Make an instrumental selection (selection of MIDI channel).

- Decide in case of an actor's insertion and/or fragmentation of the same. Establish in case of the application of an operation of deformation in the scale associated to it.

The selection of the actor, modelating instrumentation and deformation of scales is done in a probabilistic way.

As conductor orientates the dynamic tendencies of the discourse in the following levels:

- Space:

1- Tendency to the high sound.

2- Tendency to the low sound.

3- Browning movement (each transpose ascending or descending transpose of the tone is done on the previous step).

The chosen percentage for the two first operations will determinate the strength of the tendency.

- Amplitude:

- Progressive increase of the sonority itself.

- Progresive decrease of the sonority.

- Fluctuation between opposite terms.

- Time:

- Progresive increase of lineal density (acceleration).

- Progresive decrease of lineal density (desacceleration).

METHODOLOGY OF COMPOSITION

The methodology of composition with Sinapsis can be sinthesized as a set of steps consisting in:

1. Construction of actors (list of data time and MIDI) and establish their potential behaviour.
2. Grouping in actors' fields.
3. Construction of the directors of scenes establishing:

- Temporary fluctuation margins.
- General sonority.
- Register.
- Basic tone.
- MIDI channel on which timbric selections could be made.
- Types of dynamic conduction in time, sonority and space.

By means of the superposition and/or juxtaposition of Directors of scenes, the User can create a STRUCTURED PLAN where the system creates generates (in each performance) a particular discourse, A SPECIMEN that will belong to the universe of probabilities admited by that model.

Simultaneously to the process of elaboration of this the computer will activate the generators of sounds connected to the system and store the information in the form of a data list time and MIDI in the memmory zone for a further edition performance of the version of the storage in file (EGM or MIDI).

IMPLEMENTATION

Sinapsis was conceived as a Pascal Turbo language 5.5 to be run in a PC AT 386 or following using an interface MIDI, CPU-401 as a means of communication with peripheral generators.

Data MIDI sending of messages, including those for the Yamaha line (DX7, TX81Z, TG55 and TG77 have been implemented in Sinapsis together with the remote activation (messages of exclusive system).

CONCLUSIONS

A lot of experiments have been carried out, some of them have been oriented to the development of a complete composition, others have been oriented to generate partial fragments, having been observed that the system is apt to guarantee its efficiency as regards dynamic process and unity of discourse. In spite of that Sinapsis is very far to be in position to replace the creative role of the composer. Its possibilitties are limited to make contributions to a structural plan, a multiplicity of constructive proposals (simulations), some of them will be able to be used in due time.

REFERENCES

Bolognesi Tommaso. Automatic Composition: Experiments with Self-similar Music. Computer Music Journal. Vol. 7 Nº 1. p 25-36.

Cuevas Agustin Gonzalo. Teoria de la Información, Codificación y Lenguaje. Ed. SEPA. 1986.

Dawkins R. El Relojero Ciego. Ed. LABOR. 1989.

Dewdney A.K. Un Relojero Ciego Supervisa el País de los Biomorfos. Investigación y Ciencia. Abril 1988 Nº 139. p 88-92.

Holtzman S.R. Using Generative Grammars for Music Composition. Computer Music Journal. Vol. 5 Nº 1. p 51-63.

Englert Giuseppe. Automated Composition and Composed Automation. Computer Music Journal. p 30-35.

Eigen Manfred. Cuasiespecies Viricas. Investigación y Ciencia. Septiembre 1993. Nº 204. p 14-22.

Jones Kevin. Compositional Applications of Stochastic Processes. Computer Music Journal. Vol. 5 Nº 2. p 45-61.

Lorrain D. A Panoply of Stochastic "Cannons". Computer Music Journal. Vol. 4 Nº 1. p 53-81.

McAdams Stephen and Bergman Albert. Hearing Musical Stream. Computer Musical Journal. Vol. 3 Nº 4. p 26-60.

Prata Stephen. Vida Artificial. Ed. ANAYA. 1994.

Rock Irving y Palmer Stephen. El Legado de la Psicologia de la Forma. Investigación y Ciencia. Febrero 1991 Nº 173. p 50-57.

Roads C. Grammars as Representations for Music. Computer Music Journal. Vol. 3. Nº 1. p 48-55.

Sumner Richard A. Patter in Music From B Kleinmuntz ed. Formal Representation of Human Judmen. New York. WILEY. 1968. p 327-343.

SINTESE SONORA COM HARMONICOS ESCORREGADIOS

Jorge Antunes
Departamento de Música da Universidade de Brasília
Campus Universitário, 70910-900 Brasília , DF - Brasil

Resumo

Com o sistema UPIC é possível o desenho de *arcos* com o *mouse* ou com uma caneta eletromagnética sobre um grande quadro digitalizado. Este método gráfico permite a exploração da altura, do andamento, da intensidade, da duração e do timbre no largo e completo espectro audível, de modo não discreto.

O autor construiu, com ondas senoidais, um espectro harmônico de 24 sons parciais, em busca de novos timbres evolutivos. Ele estava interessado em trabalhar com aquele pacote de sons como se fosse um átomo com seu núcleo e elétrons em suas respectivas órbitas.

Inspirado no fenômeno físico da isomeria, quando elétrons saltam de uma órbita para outra, o autor trabalhou com o espectro sonoro de modo a evitar a existência de sons sustentados: a cada momento um harmônico salta, com um glissando, para um outro nível de harmônico (órbita).

Desta forma passa a ser construída uma contínua variação tímbrica, com o mesmo fundamental e a mesma altura, em que as mudanças dos harmônicos com rápidos glissandos criam uma textura complexa e imbricada.

SINTESE

A partir dos trabalhos de François Marie Charles Fourier (1772-1837) a consciência sonoro-musical de acústicos e músicos deu lugar à construção de um grande e novo tesouro cultural. Qualquer estudo aprofundado acerca dos progressos científicos e estéticos influenciados pela série harmônica e pela transformada de Fourier daria lugar a prolixas olas homenageando não apenas R. Strauss, Varèse, Koenig e Helmholtz, mas também Eimert, Schaeffer, Max Mathews, Risset, Chowning e Gabor.

Qualquer compêndio de música ou de acústica, desde o mais elementar ao mais aprofundado, não deixa de abordar a questão da síntese sonora sem abordar a questão da análise sonora. O entendimento de um é feito através do entendimento do outro. A regra dos contrários é a regra pedagógica : analisar é "desmontar"; sintetizar é "montar". A indefectível analogia com o prisma, a luz branca e o disco de Newton está sempre na ordem do dia.

Mas, depois da informação livresca, onde o músico pratica a famosa síntese sonora? Tal como todos os compositores de música eletroacústica, tivemos sempre acesso à síntese "empacotada", *prête-à-porter*. Buscamos durante quase três décadas a oportunidade de realizar uma verdadeira síntese sonora, artesanal, em que pudéssemos, usando a série de Fourier, construir um pacote de sons em que cada harmônico fosse disposto, um por um, a nosso bel-prazer.

Dos livros e dos museus, conhecemos todos as experiências do século XIX com os ressoadores de Koenig para a análise e as sirenes de Helmoltz para a síntese. Estas experiências as lemos, mas não as escutamos.

Para os ouvidos restam-nos alguns exemplos raros, mas insuficientes e precários: a faixa de exemplos do início do primeiro disco, em 45 rpm, dos trabalhos de Ernest Krenek, e um dos discos ilustrativos do *Traité des Objets Musicaux* de Pierre Schaeffer.

Desde os anos 60, por mais sofisticados que sejam os Estúdios de Música Eletroacústica, a síntese meticulosa e científica sempre foi impossível. Os estúdios de Milão, Paris, Utrecht, Bourges, Buenos Aires e New York sempre estiveram, em suas versões clássicas, bem servidos de uma boa quantidade de geradores de onda senoidal. Mas qualquer experiência no domínio da síntese aditiva com sons harmônicos sempre foi feita "de ouvido", de modo amadorístico e imperfeito, por falta de recursos de medição perfeita. Dentre os estúdios por que passamos, o único que dispunha de um freqüencímetro era o do CLAEM do Instituto Torcuato Di Tella. Assim mesmo, o seu índice de erro era enorme, com uma flutuação de 5 Hz no dígito das unidades.

O sonho de realizar uma síntese científica e auditivamente perfeita foi, assim, longo. A peregrinação de estúdio em estúdio, durante os últimos 33 anos, foi enfim gratificante, no sentido da realização desejada, quando de nosso trabalho no UPIC em 1993 e em 1995.

Crítica às Sínteses Granulares

Síntese Granular é moda. Ela é antiquíssima, pois que foi inventada por Dennis Gabor em 1946. Iannis Xenakis a praticou, sem saber, no Centre Bourdan do GRM, em Paris, quando em 1952 construiu uma micro-montagem com pedacinhos de fita magnética ordenados aleatoriamente, os retirando ao acaso de um chapéu.

Hoje a expressão é moda nos círculos musicais e científico-musicais. Inúmeras pesquisas e vários artigos são divulgados a respeito. A Síntese Granular é estudada e praticada por toda a comunidade de jovens pesquisadores e compositores que se embrenham na pesquisa sonora de controles MIDI e "comandos gestuais". A nova geração fascinada com o *Power Glove* e com as ferramentas de espacialização e tratamento sonoro em tempo real se dedica à Síntese Granular.

Mas a expressão "síntese", atualmente, se generaliza de modo indevido e incorreto. Confunde-se "síntese sonora" com "composição de objetos sonoros". Muitos dos programas desenvolvidos atualmente, anunciados como "Síntese Granular", têm como objetivo e resultado a construção de "nuvens de pontos", e não a construção de um "som sustentado".

A expressão "Síntese", em nosso entender, deve continuar a ser usada exclusivamente para designar o trabalho de construção de um único som sustentado, de parâmetros fixos ou variados. A construção de seqüências de sons sai do domínio da "síntese sonora" e entra no campo da "composição".

Precisão de Medidas

Em geral são poucos os *zooms* sucessivos que podemos realizar nos programas disponíveis (Pro-Tools, Sound Designer, Sample Cell, etc), quando trabalhamos com grandes objetos. A possibilidade de "ver" o signo gráfico representativo do som, em seu detalhe interior, através do "*zoom* de *zoom* de *zoom* de *zoom* de...", nem sempre é satisfatória. No UPIC, com a escolha adequada do tamanho de enquadramento de um objeto, é possível a realização sucessiva de dezenas de *zooms*. Com este procedimento torna-se possível a leitura ou a escritura de um "arco", cuja freqüência tenha precisão de medida até a segunda casa decimal.

Foi com o uso destes vários *zooms* sucessivos, aliado à escolha adequada de uma tabela de freqüências, que nos foi possível a medição perfeita, até a segunda casa decimal, dos 24 harmônicos de nosso "som sintético". Assim, o primeiro espectro construído, o de um Fá# grave, tinha um fundamental de exatamente 92,48 Hz.

Este trabalho minucioso de desenho de um espectro harmônico incomodou bastante os técnicos assistentes do UPIC porque este preciosismo não é comum entre compositores, e nunca havia sido praticado naquela máquina. Este detalhismo triplica o tempo de uso normal do estúdio. Em geral os compositores convidados a trabalhar durante apenas uma ou duas semanas no UPIC, ali realizam seus trabalhos "pintando" curvas variadas diretamente no monitor, através do uso do *mouse*, em arroubos criativos mais orientados pela intuição do que pelo cálculo meticuloso.

É importante ressaltar o número de harmônicos por nós utilizados: 24. Os exemplos clássicos de síntese aditiva, divulgados nas antigas gravações de Eimert, Krenek e Schaeffer, não passavam de 16 harmônicoss, no máximo.

Transformação tímbrica

Construído um largo espectro harmônico podemos, e assim foi feito freqüentemente nos estúdios clássicos, realizar variações tímbricas através da variação contínua do parâmetro intensidade. Em resumo, o velho procedimento tem as características de uma filtragem, na medida em que o aumento ou a diminuição gradual da intensidade dos harmônicos resulta na construção de um pacote de sons parciais de diferentes formas dinâmicas.

A variação tímbrica não discreta de uma forma de onda a outra é bem conhecida de todos aqueles, egressos do estúdio analógico, que trabalharam com o EMS ou o Moog, em que um simples botão de "balanço" transforma contínua e gradualmente, uma onda triangular em uma onda quadrada, ou em uma onda dente-de-serra.

Nosso objetivo era outro. Queríamos realizar, com o largo espectro harmônico, o procedimento que havíamos usado em 1970 na obra Isomerism para orquestra de câmara. No início desta peça as cordas realizam um *cluster* de 24 sons com os 12 sons do sistema temperado e suas oitavas. O *cluster* é permanente, imutável no que concerne às notas que o compõem, mas tem seu timbre variado continuamente: os instrumentos trocam de notas entre si, com glissandos contínuos. Assim, por exemplo, o violino que tocava um Lá 3 "glissa" até um Sol 2, enquanto a viola que tocava um Sol 2 "glissa" até o Lá 3.

Desenhando a Síntese

Com o UPIC podemos "desenhar" (e ouvir) até 128 arcos simultâneos. A cada uma destas linhas, retas ou curvas, que se alongam horizontalmente evoluindo no eixo do tempo, podemos associar qualquer parâmetro. A prática usual de se utilizar o eixo vertical para as freqüências nos permite, portanto, a construção de um *cluster* de 128 notas se a cada linha associarmos uma diferente forma de onda. Mas também podemos construir um "som complexo" de 128 parciais, no caso de a cada linha associarmos uma onda senoidal.

O nosso trabalho de síntese, que pretendia se voltar à construção de um som de altura determinada, optou pelo uso da Série de Fourier com a simultaneidade de 24 sons harmônicos, senoidais. O objetivo final era a construção de uma variação tímbrica contínua, nos moldes da técnica usada com o *cluster* de Isomerism. Assim, todo o espectro foi reconstruído com o "desenho" de glissandos de uma horizontal a outra, transformando o conjunto de paralelas horizontais em um novo emaranhado de retas oblíquas que esporadicamente se cruzam.

Poder-se-ia afirmar que tal procedimento não deixa de ser uma nova espécie de filtragem, na medida em que cada harmônico, a cada momento, deixa de existir, com sua "ida" em glissando à freqüência de um outro harmônico. A "filtragem" contínua, entretanto, em nada se assemelha às filtragens conhecidas: passa-alto, passa-baixo, passa-banda, de oitava, de terça, polifônica, em pente, em varredura, etc. Todos os harmônicos "abandonados" voltam a existir de momento a momento. Por outro lado, também de momento a momento o espectro deixa de ser, instantaneamente, harmônico, na medida em que glissandos (variações contínuas de freqüência) ocupam o espaço de tempo entre o "abandono" de um harmônico e a "chegada" a outro harmônico. Nesta última afirmação é preferível ser usada uma linguagem simbólica e se falar em "órbita" ao invés de "harmônico". Assim, um harmônico abandona sua "órbita" para, através do glissando, alcançar outra "órbita". O modelo simbólico se reporta à analogia entre as organizações internas de um som harmônico complexo e de um átomo: ambos com suas órbitas fixas, com níveis de energia quantizados. O som fundamental da série harmônica corresponde ao núcleo do átomo, ambos com seus campos gravitacionais. Os sons harmônicos correspondem às diferentes órbitas de cada elétron. O harmônico que salta a outro nível através de um glissando corresponde ao elétron que salta de uma órbita a outra.

A velocidade de leitura, no eixo horizontal do tempo, no UPIC, é variável à vontade do utilizador. Assim os "saltos", em glissandos, podem ocorrer de modo muito veloz, como um *portamento*, ou de modo muito lento e gradual.

O resultado sonoro é um som sintético, harmônico, complexo, de altura fixa e clara, pois que todos seus harmônicos estão, bem afinados, constantemente presentes. Mas o som sintético adquire um rico desenvolvimento interno graças à progressiva variação tímbrica e às periódicas intervenções de tramas de glissandos internos.

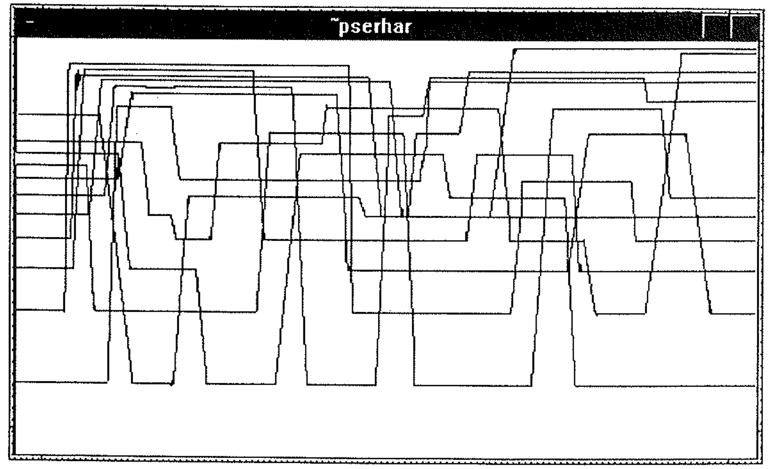
Transposições

No *menu* "Frequency Table", no UPIC, podemos ter acesso ao *window* "Define" onde fixamos o "ambitus" da tabela de freqüências. Assim, com um simples cálculo de "regra de três" podemos, modificando o âmbito da tabela de freqüências, transportar o "espectro de harmônicos escorregadios", livremente para cima (agudo) ou para baixo (grave).

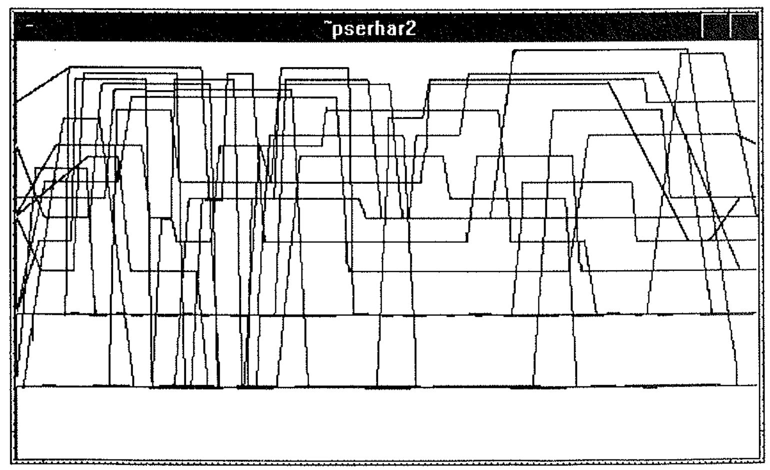
A segunda experiência realizada consistiu na construção de um novo "espectro de harmônicos escorregadios" com freqüência de fundamental igual a 100 Hz (harmônicos: 200 Hz, 300 Hz, 400 Hz, 500 Hz, ..., 2400 Hz). A partir deste som complexo que corresponde a um Sol# desafinado para baixo, foi possível realizar transposições a todas as alturas do sistema temperado através da modificação do "ambitus" da tabela de freqüências. A construção com fundamental de 100 Hz não invalida o trabalho minucioso de medição exata até a segunda casa decimal, pois que aqui tivemos o cuidado de desenhar os parciais com 100.00 Hz; 200.00 Hz; 300.00 Hz; etc, isto é, também com exatidão até a segunda casa decimal. O tecnicismo apressado sempre nos estará tentando lembrar que o ouvido humano não dá conta deste preciosismo, mas não concordamos com esta falsa acertiva psicoacústica. Ou será que o ouvido do autor deste trabalho não é humano?

O cálculo que o computador realiza na mudança de uma tabela de freqüências para outra, pode ser ouvido em tempo real. Dependendo da velocidade da máquina, os saltos graduais, de um espectro a outro, de cada som parcial, dá lugar a verdadeira "cascata" de sons que interligam dois espectros.

Estes procedimentos realizamos para a fabricação de rico material sonoro utilizado nas obras compostas no UPIC: "Interlude Nº 1 para Olga" (1993) e "Agenda pour un Petit Futur" (1995). O mesmo material sonoro foi usado em "Ballade Dure" (GRM, 1995), "Vitraux MCMXCV" (GMEB, 1995) e "La Beauté Indiscrète d'une Note Violette (GMEB, 1995).



Síntese sonora com harmônicos "escorregadios": 24 harmônicos, com saltos em glissando de uma órbita a outra, com o desaparecimento momentâneo e esporádico de alguns harmônicos.



Síntese sonora com harmônicos "escorregadios": 24 harmônicos, com saltos em glissando de uma órbita a outra, com a presença constante dos harmônicos 1 e 2 (o fundamental e sua oitava).

MWSCCS: A Stochastic Concurrent Music Language

BRIAN J. ROSS
Brock University
Department of Computer Science
St. Catharines, Ontario, Canada L2S 3A1
bross@sandbanks.cosc.brocku.ca

Abstract

The paper describes a music composition language MWSCCS – the Musical Weighted Synchronous Calculus of Communicating Systems. MWSCCS is a stochastic language based on Tofts' WSCCS and Milner's SCCS algebra. Main features of MWSCCS are its straight-forward approach to probabilistic execution, the ability for processes to generate musical events autonomously or to communicate amongst each other, the ability to write prioritized reactive processes, and its concise, hierarchical set of operators. MWSCCS diverges from the pure WSCCS algebra by introducing various devices useful for music, for example, a MIDI mappable event space, and stream repetition for the generation of repetitive melodies and rhythms. MWSCCS's execution engine consists of a search mechanism, which replaces the pure algebra's denotational semantics over streams. The implementation of the language benefits from the formal semantic definition of the original process algebra. The language is implemented in Prolog. Work is under way in creating compositions with MWSCCS that exhibit interesting chaotic and self-organizing behaviours.

1 Introduction

The perspective of music as a concurrent activity is well established. Music models using Petri nets (Haus & Rodriguez, 1988; Haus & Sametti, 1992) and algebraic interleaving (Chemillier & Timis, 1988; Chemillier, 1990) are rooted in the concurrent computation paradigm to different extents. This papers contribution is to suggest a music programming language based on a process algebra. The process algebra model considers music as comprised of discrete, concurrent behaviors that can be observed and reacted upon by musical processes. This can be intuited by considering a small improvisational jazz band. A competent musician does not play independently with respect to the rest of the band, but rather, reacts to musical events and cues from the other musicians. Such cues can range from maintaining step with subtle tempo drifting, to compositional cues such as the playing of notes that fulfil melodic or harmonic obligations with the notes played by other instrumentalists. We therefore identify a *musical event* as being an observable discrete phenomena that can be observed by others, and which can ellicit reactions. We identify the ability to observe, recognize, and react to such events as being an important ingredient to human and computer music processing and composition.

The music programming language discussed in the paper is the Musical Weighted Synchronous Calculus of Communicating Systems, or MWSCCS. It is an implementation of the MWSCCS process algebra, enhanced with features useful for music composition. Some benefits of the language include a concise yet powerful set of basic primitives, the ability to create abstract hierarachical musical systems, and an intuitive mechanism for specifying complex stochastic behaviours of musical processes.

The format of the paper is as follows. A brief discussion of the process algebra foundation of MWSCCS is in section 2. Section 3 discusses the MWSCCS language, and section 4 gives some example MWSCCS compositions. A discussion concludes the paper in section 5.

2 Process Algebra

An *agent* or *process* is an abstract mechanism whose behavior is characterised by discrete actions. Process algebra are mathematical formalisms for modelling processes. There are many process algebra in the literature (eg. (Hoare, 1985; Hennessy, 1988; Milner, 1989)), each of which establish different perspectives and styles for modeling concurrency. Process algebra are effective models of concurrency because of the high degree of abstraction possible with them, as well as their intuitive "programming language" feel. Along with process algebra, two other algebraic models of concurrency include Petri Nets (Peterson, 1977) and trace theory (Aalbersberg & Rozenberg, 1988). Although all three formalisms share much in common – they all define finite automata – they do differ in the nature of their semantics and subsequent analyses. For example, process algebra tend to be more abstract and specification-oriented than Petri nets, while the latter describe concurrency at a more intricate structural level.

The MWSCCS process algebra is described in detail in (Ross, 1995a; Ross, 1995b). MWSCCS is a musically-extended version of Tofts' WSCCS algebra (Tofts, 1990), which is in turn a stochastic version of Milner's SCCS (Milner, 1989). SCCS (synchronous calculus of communicating systems) is a process algebra in which processes contribute their visible activity synchronously, or in other words, in unison with a global clock. The algebra also contains operators for structuring process definitions, renaming and inhibiting actions, and permitting nondeterministic choices of behaviour. WSCCS adds to SCCS a mechanism useful for stochastic and reactive control of nondeterministic choice. Finally, MWSCCS adds to WSCCS some denotations useful in a musical context, for example, an event space mappable to MIDI. The algebraic semantics of all these algebra are defined in terms of transitional rules of inference. Each rule describes the behaviour of an operator in terms of the relation it defines with respect to a transition over the stream of observable events. A discussion of these rules is beyond the scope of this paper (please see the aforementioned references). However, it is worth noting that a transitional inference rule semantics is invaluable for creating a correct implementation for the algebra as done in this research.

MWSCCS treats music as a stream of observable discrete events. When applied to music, this stream denotes the horizontal structure of music – the relative sequential order of events with respect to each other. The vertical component of music – event simultaneity – is naturally denoted by the event domain of multi-particle actions. A *particle* is the smallest visible atomic event definable . Multiple particles can coalesce to produce an *action*. When multiple processes synchronously communicate their actions, their constituent actions are combined to form new actions. The process $a\overline{b}.cd$ represents two actions, $a\overline{b}$ followed by cd. Particles have two polarities (eg. a and $\overline{a}$), which cancel themselves if they coincide in one action. '1' is the identity action, and represents silence. '0' represents termination. Then the expression

$$a.b.c.d.0 \# a\bar{b}.cd.1.0 \# ab.\bar{d}\bar{e}.f.0$$

represents three processes communicating concurrently via the composition operator #. The expression reduces to $aa\bar{b}ab.bcd\bar{d}\bar{e}.c1f.d00$, which in turn simplifies to $aaa.bc\bar{e}.cf.0$. Note how the d in the first term is not generated, because the other two processes have already terminated. The power of this representation is that it naturally models musical activity. Each particle above can be considered to be a musical note, and composite notes in turn denote chords. We can colour the notes to denote particular voices or instruments if desired. A ramification of this musical interpretation is that a whole composition quits as soon as one process (instrument, voice, musician) quits.

Another useful feature of the algebra, inherited from WSCCS, is its representation of stochastic processes. In the choice expression,

$$2\omega^2 ab \ + \ 3\omega^2 cd \ + \ 4\omega^1 ef.g \ + \ 5\omega^1 h$$

the first two terms are considered before the last, as their priority value ω^2 is higher than ω^1. The first and second terms are selectable with probabilities of $2/(2+3) = 2/5$ and $3/5$ respectively, and the last terms with probabilities $4/9$ and $5/9$ respectively.

3 The MWSCCS Language

3.1 The event space

Section 2 introduced particles and actions, and how they coalesce and cancel during synchronous communication. MWSCCS supports the following event space $\mathcal{A}$:

$$\mathcal{A} \ = \ \{ \ \textit{Generic Actions} \ \} \ \cup \ \{ \ \textit{Music Notes} \ \} \ \cup \ \{ \ \mathbf{1}, \surd, -\surd \ \}$$

Generic actions are vanilla process algebraic communications, and normally are lower-case constants. *Musical Notes* uses notation for identifying standard 12-semitone multi-octave notes. For example, *as5* is *A-sharp octave 5.* In addition, notes can be coloured with a channel (between 1 and 16) and velocity (between 0 and 255), as in *as5 ch 3 vel 155.* In order to map to MIDI events, note actions can be interpreted as *Note On* and *Note Off* messages. In this interpretation, $as5$ and $as5\dagger$ denote *A-sharp On* and *A-sharp Off* respectively. The action **1** denotes silence. All actions (except **1**) have positive and negative polarities, for example, a and $-a$. The reserved particle $\surd$ denotes "active termination" of a process. This termination differs from absolute termination (**0**) in that the process does not end, but remains silent throughout the rest of the composition, having become equivalent to a process *Silent*:

$$Silent \ =d= \ \mathbf{1}.Silent$$

A process that emits $\surd$ just before becoming silent is called *well-behaved.*

Composite actions are denoted by tuples. The term $(c3,\ e3,\ g3)$ represents the chord C-major. The language permits simple function expressions over particles, which is useful in concert with parameter passing. The term $f(X+2)$ denotes the musical note two semitones above note X.

Probability and priority codes may prefix terms. In $(F,P)\Delta(Action)$, F is the relative frequency and P is the priority. If P is missing as well, the priority is taken to be 1. If F is missing, then F is 1. The expression

$$(2,2)\Delta(as5,\ -bf) \ + \ (3,2)\Delta(a5,\ -bf) \ + \ (4,1)\Delta(c5) \ + \ (5,1)\Delta(d5)$$

therefore treats the first two terms as higher priority than the other two. Probabilities are assigned as described in section 2.

3.2 Operators

The language $\mathcal{E}$ of agent expressions over an event space $\mathcal{A}$ is defined by the grammar

$$\begin{aligned} \mathcal{E} \ ::= \ & \mathbf{0} \mid X \mid (F,P)\Delta\alpha.E \mid E\lceil A \mid E\backslash A \mid E[f] \mid E_1 + E_2 \mid E_1 \# E_2 \mid \\ & E_{label}(\tilde{t}) \mid !E \mid N \ rep \ E \mid inf \ rep \ E \mid (N) \end{aligned}$$

where $E, E_i \in \mathcal{E}$; F,P,N are integers > 0; A are particles; α is an action; f is a particle renaming function; $\tilde{t}$ is a parameter list; and E_{label} is a process name. Processes definitions are defined by:

$$E_{label} \ =d= \ E$$

The *null* process **0** is the process that has absolutely terminated. The *prefix* operator in $\alpha.E$ represents the process that can perform the action α, thereafter becoming the process E. The expression $\alpha.\mathbf{0}$ abbreviates to α.

The *permission* operator in $E\lceil A$ denotes the process that performs the particles that are members of the set A. In effect, the permission operator prunes all actions not in A. The *restriction* operator in $E\backslash A$ is the converse of permission, except that it lists the actions which cannot be generated by E. In fact, $E\backslash A \ = \ E\lceil(\mathcal{A} - A)$.

The *relabelling* operator in $E[f]$ renames particles according to f, while leaving weights alone. For example, $E[b/a]$ replaces particle a with b in each action emitted by E.

The *choice* operator $E_1 + E_2$ represents the choice of execution of a set of processes according to priority and probability prefixes on the terms, as described above.

The *parallel composition* operator in $E \# F$ forces concurrent processes E and F to synchronously communicate with one another. We let $E \# F \ = \ F \# E$. If both E and F have no probability or priority information, then a new action is formed by their combined actions:

$$a.E \# b.F \ = \ ab.(E \# F)$$

When prefixes are involved, rules from the MWSCCS algebra describe how to combinate them to create new prefixes for the results.

Wherever a process variable X is found, the $\mathcal{E}$ expression bound to it is used. A reference to a process E_{label} defined with an expression $=d=$ causes that process definition to be invoked, possibly with parameter passing.

$!E$ denotes the infinite invocation of an expression. E is invoked until it absolutely terminates, at which time it is reinvoked. On the other hand, $inf\ \ rep\ \ E$ invokes E until it terminates, and then infinitely repeats the *stream* generated by E. Therefore, $!E$ generates in an indefinite stream of varying behaviour, while $inf\ \ rep\ \ E$ gives an indefinite stream of repetitive behaviour.

Finally, the notation (N) represents a delay of N clock ticks. For example (4), is equivalent to $\mathbf{1.1.1.1}$. Because compositions usually require lengthy pauses between notes or between note on/off messages, this notation is very convenient.

3.3 Useful Extensions

Sometimes it is useful to model asynchronous processes. Unlike the processes above, asynchronous processes may nondeterministically wait or stall before elliciting their observed behaviour. We can represent an asynchronous process by the following meta-operator:

$$\bullet(P) \ =d= \ P \ + \ \mathbf{1}.\bullet(P)$$

Here, P can either execute, or wait. As can be seen in this recursive definition, an asynchronous process may very well stall forever, although this is statistically unlikely.

The strict sequential execution of musical events is often required. To play sequentially, one musician's musical part should end before the next musician commences. A sequential composition operator ";" is useful for this purpose:

$$X \ ; \ Y \ =d= \ (X[c/\surd] \ \# \ \bullet(-c.Y)) \setminus \{c\}$$

where c is a new particle not generated by X or Y. The ; operator takes two processes X and Y as arguments, where X must be well-behaved. Here, X will play until it is done, at which time it generates the termination signal $\surd$. This is renamed to c when it is seen, and Y has this same c prefixed to it. When used with the restriction of c, the expression disallows c from appearing. Fortunately, when $\overline{c}$ is finally generated by the left-side, it synchronizes with c on the right, and that particle disappears, ie. $(-c, c) = \mathbf{1}$. Then Y may proceed to play. Note that we do not want X to literally die (reduce to $\mathbf{0}$), or else the whole expression will quit, since $\mathbf{0}\#E = \mathbf{0}$. Rather, X should be well-behaved, remaining silently active in order for the whole composite expression to execute to completion.

3.4 Other implementation issues

MWSCCS has been implemented in MacProlog 32 and SICSTUS Prolog under IRIX 5.3 Unix. The implementation uses a meta-interpreter over MWSCCS expressions. The implementation of various operators were directly derivable from the transitional inference semantics of the original algebra. The

utility of this formal semantic specification for the language cannot be overstated, as it permitted the the quick derivation of a correct and functional implementation.

However, the denotational semantics of concurrent composition $\#$ in the process algebra is defined over an exhaustively complete universe of behaviour. This definition is not suitable for use in the implementation, since exponential space and time is required for its construction. Therefore, concurrent composition is implemented with depth-first inferential search. The advantage of this is the avoidance of exponential resource usage; the possible disadvantage is inefficient run-time execution search should an MWSCCS program be ill-structured. This implies that MWSCCS programs should be structured for efficient execution in mind, rather than rely on the execution engine to find results. The use of search is, of course, naturally supported by Prolog.

4 Example

4.1 Grammatical composition structure

Using the notion of well-behaved termination, WSCCS can duplicate the expressiveness of regular grammars, and therefore permits grammatical description of composition (Roads, 1979) For example, we might like to structure a composition as:

$$\begin{array}{rcl}
Tune & =d= & Prelude \ ; \ Main \ ; \ Finale \\
Prelude & =d= & 2ccdce.\surd.Silent \ + \ cdcee.\surd.Silent \\
Main & =d= & FirstPart \ ; \ SecondPart \ ; \ Climax \ ; \ Resolution \\
FirstPart & =d= & 2ccdce.FirstPart \ + \ cdfee.FirstPart \ + \ g.\surd.Silent \\
\ldots & & \\
Finale & =d= & d.d.b.d.ceg.\surd
\end{array}$$

Here, there are one of two possible preludes possible, followed by the main body and finale. *FirstPart* has the regular expression form $(X+Y)^*Z$.

4.2 A more complex example

$$\begin{array}{rcl}
Music & =d= & ((Melody \ [x/e5, x\dagger/e5\dagger, b/\surd] \\
& & \# \ Acc_7_9(x, e5, b) \setminus \{\alpha, x, x\dagger, b\}) \ [y/ef5, (y\dagger)/(ef5\dagger), b/\surd] \\
& & \# \ Acc_7_9(y, cf5, b)) \setminus \{\alpha, y, y\dagger, b\} \\
Melody & =d= & (M1 \ ; \ M2) \ ; \ (M2 \ ; \ M1) \ ; \ \mathbf{0} \\
M1 & =d= & (2,1)\Delta(c5, e5, g5).\mathbf{1}.(c5\dagger, e5\dagger, g5\dagger, \surd).Silent \\
& & + \ (1,1)\Delta(cf5, ef5, g5).\mathbf{1}.(cf5\dagger, ef5\dagger, g5\dagger, \surd).Silent \\
M2 & =d= & (2,1)\Delta(c5, ef5, g5).\mathbf{1}.\mathbf{1}.(c5\dagger, ef5\dagger, g5\dagger, \surd).Silent \\
& & + \ (1,1)\Delta(c5, ef5, gs5).(c5\dagger, ef5\dagger, gs5\dagger, \surd).Silent \\
Acc_7_9(\alpha, \beta, \pi) & =d= & (1,2)\Delta(-\pi, \surd).Silent \\
& & + \ (3,1)\Delta(-\alpha, \beta, f(\beta+7)).Rel_7_9(R, X, 7, \pi) \\
& & + \ (2,1)\Delta(-\alpha, \beta, f(\beta-9)).Rel_7_9(R, X, -9, \pi) \\
& & + \ (1,0)\Delta\mathbf{1}.Acc_7_9(\alpha, \beta, \pi) \\
Rel_7_9(\alpha, \beta, \gamma, \pi) & =d= & (1,2)\Delta(-\pi, -\alpha\dagger, \beta\dagger, f(\beta+\gamma)\dagger, \surd).Silent \\
& & + \ (1,1)\Delta(-\alpha\dagger, \beta\dagger, f(\beta+\gamma)).Acc_7_9(\alpha, \beta, \pi) \\
& & + \ (1,0)\Delta\mathbf{1}.Rel_7_9(\alpha, \beta, \gamma, \pi)
\end{array}$$

This example uses both grammatical structure, stochastic nondeterminism, and reactive communication. It also uses the *Note On* and *Note Off* notation suitable for MIDI generation. The main musical process is *Music*, which plays a basic melodic line (*Melody*) along with two accompaniment processes (*Acc_7_9*). Melody uses two subprocesses *M1* and *M2*, which stochastically generate simple chords. *Music* then uses particle renaming when invoking *Acc_7_9*, and restricts the renamed particles so that they will be invisible to the audience. This style of relabelling of events and restricting their emmission is a technique used by process algebras to control communication. The calls to *Acc_7_9* make generous use of argument passing, to make that process as general as possible.

Acc_7_9 does three things, which are prioritized as follows. Firstly, the $(1,2)\Delta$ term checks if *Melody* has terminated, and if so, it quits as well. Otherwise, it checks if *Melody* has generated the event α. If so, it chooses to generate either a note 7 semitones higher, or a note 9 semitones lower. The process *Rel_7_9* is used to release the accompanied tone previously generated. Finally, if neither of the above cases have occurred, *Acc_7_9* waits until the next communicated event.

The use of priorities in *Acc_7_9* and *Rel_7_9* serves two purposes. Firstly, they allow the processes to behave correctly. For example, if we used the same priority value in all the terms of *Acc_7_9*, it is possible that the silent waiting term might be selected when in fact *Melody* has terminated, which prevents the generation of the note off and termination signal actions. Secondly, from an efficiency point of view, priorities greatly reduce the need for search. The most critical actions to take have higher priority and are tried first.

5 Conclusion

Work is under way in using MWSCCS for serious music compositions. One composition being undertaken uses MWSCCS to simulate self-organizing behaviour. In particular, Tofts' WSCCS simulations of ant colony behaviours in (Tofts, 1992) have interesting applications in music. One such MWSCCS composition exploits autosynchronizing behavior which, in a stochastic setting, generates interesting cyclic musical activity. MWSCCS will also be used to formally analyze compositions written in it.

A concurrent music language similar to spirit to MWSCCS is the Petri net (PN) language in (Haus & Sametti, 1992). The similarities and differences between it and MWSCCS are reflected in the inherent differences between process algebraic and Petri net models of concurrency (Nielsen, 1987). Possible advantages of MWSCCS over Petri nets are its more abstract view of musical behaviour, it's lean but powerful set of operators, and its intuitive execution semantics. It is also worth mentioning the similarity between MWSCCS and other conventional music programming languages (Loy & Abbott, 1985), especially those using object-orientation. The main advantage shared by WSCCS and the PN language in comparison to other music languages are their well-defined mathematical foundation, which permits direct formal analyses of systems built with them.

Acknowledgement: Support through NSERC Operating Grant 0138467 is gratefully acknowledged.

References

Aalbersberg, I.K., & Rozenberg, G. 1988. Theory of Traces. *Theoretical Computer Science*, **60**, 1–82.

Chemillier, M. 1990. Solfege, Commutation Partielle et Automates de Contrepoint. *Math. Inf. Sci. Hum.*, **28**(110), 5–25.

Chemillier, M., & Timis, D. 1988. Toward a theory of formal musical languages. *Pages 175–183 of: ICMC 95.*

Haus, G., & Rodriguez, A. 1988. Music Description and Processing by Petri Nets. *Pages 175–199 of:* Rozenberg, G. (ed), *Advances in Petri Nets (LNCS 340).* Springer-Verlag.

Haus, G., & Sametti, A. 1992. ScoreSynth: A System for the Synthesis of Music Scores Based on Petri Nets and a Music Algebra. *Pages 53–77 of:* Baggi, D. (ed), *Computer-Generated Music.* IEEE Computer Society Press.

Hennessy, M. 1988. *Algebraic Theory of Processes.* MIT Press.

Hoare, C. A. R. 1985. *Communicating Sequential Processes.* Prentice–Hall.

Loy, G., & Abbott, C. 1985. Programming Languages for Computer Music Synthesis, Performance, and Composition. *Computing Surveys*, **17**(2), 235–265.

Milner, R. 1989. *Communication and Concurrency.* Prentice Hall.

Nielsen, M. 1987. CCS - and its Relationship to Net Theory. *Pages 393–415 of:* Brauer, W. (ed), *Petri Nets: Application and Relationship to Other Models of Concurrency (LNCS 255).* Springer-Verlag.

Peterson, J.L. 1977. Petri Nets. *Computing Surveys*, **9**(3).

Roads, C. 1979. Grammars as Representations for Music. *Computer Music Journal*, **3**(1), 48–55.

Ross, B.J. 1995a (February). *A Process Algebra for Stochastic Music Composition.* Tech. rept. CS-95-02. Brock University, Dept. of Computer Science.

Ross, B.J. 1995b. A Process Algebra for Stochastic Music Composition. *In: Proc. International Computer Music Conference.*

Tofts, C. 1990. A Synchronous Calculus of Relative Frequency. *In:* Baeten, J.C.M., & J.W.Klop (eds), *CONCUR 90.* Amsterdam, The Netherlands: Springer-Verlag LNCS 458.

Tofts, C. 1992. Describing Social Insect Behavior Using Process Algebra. *Transactions of The Society for Computer Simulation*, **9**(4), 227–283.

Composing with Chaos; applications of a new science for music

DAVID CLARK LITTLE
Sweelinck Conservatory Amsterdam
Cornelis Springerstr. 14-2
1073 LJ AMSTERDAM
the Netherlands

In this paper the author shows where concepts and mathematical models derived from the developing field of Chaos Science can be applied to electroacoustic and instrumental composition. Examples of non-linear dynamics include Lorenz's model of fluid behaviour, Verhulst's model of population growth, Hénon's analysis of the multiple celestial body problem, Barry Martin's Algorithm which produces quasi-organic forms, and the 'Baker' mixing function. Besides broadening the numerical techniques available for electronic music generation, concepts such as fractal structure, feedback process and iterative function can be applied to 'ordinary' composition as well. For example, in designing melodic curve, defining meter, planning instrumentation, manipulating symbols, creating ornamentation and elaboration, etc. Some suggestions as to mapping are made, the critical boundary between science and art. Musical examples are used from the following works by the author: Harpsi-Kord for harpsichordist and tape, Fractal Piano for computer-guided pianola, The Five Seasons for 6 percussionists and tape, Brain-Wave for recorder-players, Modi-Fications for marimba & tape, and Hyperion's Tumble for tape.

300 years ago Newton formulated the laws of motion which laid the ground-work for a clockwork view of the universe. By the late 18th century the French astronomer Laplace optimistically stated that intelligent creatures could know any past or future state of the universe, if they only knew well enough its present state, what direction it was heading towards, and had powerful enough calculating methods. This deterministic world view has proved to need revision. Scientific and mathematical developments of the last 30 years have led to new insights into subjects, which because of their complexity, had previously been swept under the rug by the scientific establishment. Intractable problems in weather forecasting, the modelling of wildlife populations, the geometry of nature, the understanding of turbulent flow and bio-rhythms gave startling new results when revolutionary methods of analysis were applied. As a result, words such as "chaos", "order", "simple" and "complex" have been redefined; and a new concept formed: "fractal".

Ironically, it took the advent of the deterministic tool *par excellence* namely the computer, to cause many contemporary scientists to rethink the whole matter. With mathematical models they had been able to make accurate predictions of planetary motions and tides, for example. Everyone had thought that long-range weather prediction should also be possible; you just had to make much more calculations. In 1961 the meteorologist Edward Lorenz managed to model the Earth's weather on a computer; one could follow recurrent "rain storms", or "cyclones", etc. Only there was a problem: if he started the program with slightly different initial conditions of wind speed and temperature, the artificial weather would be the same as in a previous run only in the beginning. After a while, the "weather" would diverge from the previous run, and eventually end up completely different! (See fig. 1.) To appreciate what this means, one must remember that the computer model was using proven physical laws of gas and water behaviour; and the computer ran completely deterministically with no additional input after it was started. With Lorenz dawned the idea that long-range forecasting was impossible. Small errors in measurements would multiply, cascading upwards in the scale of turbulence: from a puff of wind to continent-sized spirals. Lorenz called it the "Butterfly Effect"- theoretically, a butterfly stirring its wings in Peking could start a storm over New York the next month!

Lorenz later developed a more general mathematical model of fluid behaviour. It describes the flow of heated fluid, called convection. For example, when a pan of water is heated, the hotter water at the bottom tends to rise, because it is less dense. At the top of the pan it comes into contact with air which cools it off somewhat. Then the cooled denser water sinks back to the bottom of the pan. This circulation of fluid is called a convection cell, and remains smooth and orderly as long as the heat under the pan is moderate. However, if the heat is high, the

water moves to fast to cool off very much; the convection cell breaks up and flow is turbulent, as portions of the water compete with each other to get to the top. Lorenz took the Navier-Stokes equation which describes fluid flow, and simplified it to get an equation to model convection, using three variables in non-linear relationship. (A linear relationship is where a change in one variable is mirrored by a proportional change in another variable: its graph is a straight line. A graph of a non-linear relationship, on the other hand, might show breaks, reversals, bends, etc.)

$$x_{new}=x+d*a*(y-x)$$
$$y_{new}=y+d*(x*(c-z)-y)$$
$$z_{new}=z+d*(x*y-b*z)$$

a, **b**, **c** and **d** are constants with the values 10, 8/3, 28, and .003 respectively. A new value is calculated for each variable, dependent on its previous value and the other variables in various proportions. A loop is set up by plugging the new values from a calculation into the variables for the previous state (e.g. $x=x_{new}$). Then we can run the calculations all over again. The change in values of the variables with time can be traced out in what's called a phase diagram (See fig. 2.) A point on the diagram represents the physical state of a system, actually in three dimensions. If a system heads toward a stable final state, its phase diagram would tend to localise to a point, called the attractor. For a periodic system, the phase diagram would tend to be a closed loop of some kind. Lorenz's model appears to be chaotic, with a kind of infinite complexity; it has a *strange* attractor! The trace of the model loops endlessly without repeating or crossing itself, flipping unpredictably from one side to the other. It does remain within bounds, however, and is not random; a pattern emerges resembling butterfly wings. Indeed, a new kind of order was discovered which was to reveal itself in analysis of many different natural phenomena; order within chaos. Lorenz's work started the revolution which was, like his "Butterfly Effect" to spread to many fields outside of meteorology.

At this point I'd like to describe pieces of mine which use some of the ideas just described. **Harpsi-kord** for tape and harpsichordist was composed in 1988. In this piece the central idea is order within chaos. Compositionally, it swings between the poles: regular/irregular, loud/soft, atonal/harmonic, the use of timbre from an ancient instrument or electronically generated. The middle ground is sought for by transformations sometimes possible only through new techniques: 'samples' of harpsichord sounds were adapted electronically. Sometimes techniques were turned on themselves; having sampled a tone-cluster, it was available on each note of a synthesizer. Clusters of clusters were made. Similarly, rhythmic or melodic structures were nested in several layers at times. For example, one samples not a single tone, but a melodic motive, and loops it. By holding several keys with the same sample, one generates a polymetric texture, because the same loop at higher pitch plays faster, hence is shorter. One can sample this whole texture, and repeat the process, achieving very soon the limits of human perception regarding detail! The harpsichordist relates to the tape in a quasi-improvisational manner. Although the timing and pitch material is exactly notated, he/she is given considerable freedom in performance. For example, only the pitches were notated in a square, with the rhythm and ordering "randomly" improvised. (See fig. 3.) In this way a "feedback loop" is created; the improviser must use his/her ears and think fast in order to create a proper "dialogue" with the tape.

The next two pieces, **Shuffle** and **Fractal Piano 6** (both from 1988) were realised with the help of the "Vorsetzer". The Vorsetzer is a new form of pianola developed by the technicians of the Electronic Studio of the Sweelinck Conservatory, Amsterdam. It has 88 electromagnets mounted over the keys, which can be triggered with varied degrees of force by a computer. The obvious advantage of this system over the old method of punching out rolls of paper is the inherent flexibility and compactness of data storage with computers. In addition, the use of the computer offers new compositional possibilities.

To make **Shuffle**, an 88-note chromatic scale was produced and manipulated by a computer. The scale accelerates smoothly from relative note values of quarters in the lowest register to 32nds in the highest. It has a dynamic curve of p<f>p with the loudest part occurring in the middle of the piano. MIDI-data for each note is stored in three separate memory-allocations: for pitch, timing/length, and loudness. The data for some notes are then "shuffled" around a bit by a computer program I made: the contents of two randomly chosen (but nearby) memory units for pitch, for example, are exchanged. Likewise, length or loudness data for a few other different pairs are exchanged. Then the memory-allocations are combined, and "performed" by the Vorsetzer: one hears a slightly flawed chromatic scale. The memory-allocations of this flawed scale are then subjected to the same process; data pairs are exchanged. Output is used for input for many cycles in a kind of feedback process. With each cycle, the scale becomes audibly more diffuse and irregular: notes migrate slowly away from their original position in the scale. The original perfectly ordered chromatic scale slowly "degenerates" into a "super-serial" shuffled mix. The final state is complex, and much dependent on the cumulative effect of many small random choices. In chaos theory, one would say that there is sensitive dependence to initial conditions. As such, **Shuffle** is a musical model of the butterfly effect. Here it should be added that it also resembles a 2-voiced canon in contrary motion: a descending chromatic scale enters with the 1st "shuffled" version of the ascending scale, and receives similar treatment.

Fractal Piano 6 is one of a series of studies in which a computer program I developed was used in combination with the Vorsetzer. The heart of this program is a mathematical model of population growth, first derived in 1845 by P. F. Verhulst. (It is often referred to as the logistic equation.) I'd like to describe it in some detail because although simple, it contains profound implications.

The Malthusian Model describes the unbounded growth of a population (of fruit flies, for example) with $\mathbf{x_{new}=a*x}$. This formula tells us that we can find the population of a new generation by multiplying the number in the last generation with a productivity factor. Suppose the population doubles each generation; then **a=2,** and starting with 2 parents, we'd get the series: 4 children, 8 grandchildren, 16 great-grand children, etc. It's easy to see that before many generations have been bred, we have a gigantic number. By the 10th generation that is 1024 siblings!

In order to make a more realistic model, Verhulst considered that in nature, the larger a population grows, the less productive it becomes, perhaps because of lack of food or other overpopulation problems. So in creating his (abstract) model, he says, let's set the upper limit of a population at 1. (Think of it as 100% of the room available for growth). Then the room left over by the environment for a new generation is **1-x.** This can be seen as a correction factor to unbounded growth. The Verhulst Model for limited population growth then becomes: $\mathbf{x_{new}=a*x*(1-x)}$. The population of a new generation is equal to the malthusian growth factor times the old population, and scaled down by the amount of room available for growth. In spite of its simplicity, it proves to be a fair model of what happens in nature. If the productivity factor **a** is 2, then starting the formula with a low seed value like 0.001, we see the population **x** rise and level off at 0.5. This is what we might expect in nature with animals with a healthy productivity. After an initial period of fast growth, the population stabilises.

If we set the productivity factor **a** to higher values, strange things happen. If **a** is 3.2, **x** grows rapidly at first, but then doesn't stabilise to one value; rather it alternates between two values endlessly. (See fig. 4a.) It doesn't matter what the seed value was, **x** ends up alternating between the same two values. If **a** is set a little bit larger than 3.4495, we find the values for **x** orbiting between four values eventually. Carefully increasing the value of **a** for still more trials, we find that the number of values that **x** seems to land on keeps bifurcating (to 8, and 16) until there is a value for **a**, 3.569946, just beyond which **x** fluctuates chaotically from one value to the next. Sometimes it bounces back and forth between a couple of values for a while, only to spin off again. (See fig. 4b.)

This type of chaotic behaviour is also observed in nature, for example by an animal with a productivity so high that it overreaches the ability of the environment to support it. The population crashes, only to build up again. The interesting thing about the model is that it does show a kind of regularity, with **x**-values jumping up and down, but it *never repeats itself exactly.* This simple, deterministic mathematical formula can be just as erratic as measurements of real populations in nature!

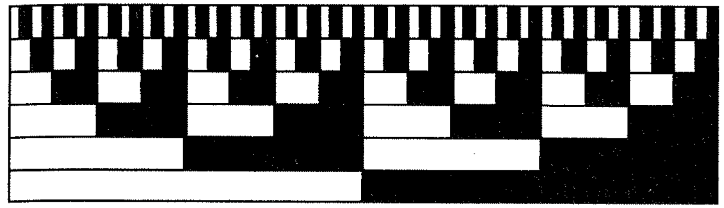
There are more mysteries lurking here. While searching for the exact values of **a** where the behaviour of the model changed-where **x** values would settle down eventually to one, two, four or eight values-the physicist Mitchell Feigenbaum recently discovered a constant ratio between the **a** values. Still more astonishing was the discovery that other quite different mathematical formulas (still using an output-input loop to calculate a new value from an old value), and also experimental data exploring the onset of turbulent flow, also showed the doublings, *and the same ratio between them,* 4.6692... In short, a new universal constant was discovered by Feigenbaum, like the constant of gravity, the speed of light, or the weight of an electron.

We're not yet finished with Verhulst's model. If **a** is increased to 3.83, the chaotic behaviour eventually stops, and **x** circles eventually between only three values. (See fig. 4c.) Increasing **a** in small amounts for new trials results in period doubling of the values where **x** eventually settles down to- 6,12; and again chaotic behaviour sets in, up to **a**=4. (See fig. 4d.) (We cannot set **a** to a number greater than 4, because that would produce **x** values greater than 1, or exceeding our original definition of the maximum population). With the help of a computer, a graph can be made of how Verhulst's formula behaves for all settings of the **a** value. (See fig. 4e.) We see the doublings of **x** at so called bifurcation points, followed by chaotic regions, then windows, where **x** again has a low number of stable values. We get a shock of recognition when we magnify the region where **x** splits up again; the whole pattern reveals itself in miniature! (See fig. 4f.) Indeed, it seems that the pattern contains nested within itself, its own replica! This kind of nested pattern is now called a "fractal". The Polish-born mathematician Benoit Mandelbrot derived the term from the Latin adjective *fractus,* meaning irregular or broken. Fractals are characterised by intricately nested patterns within patterns, with self-similarity on any scale. Fractals can be recognised in a wide range of natural phenomena and shapes, such as trees and clouds. Analysis of Indonesian Gamelan music reveals fractal structure. (See fig. 5). The rhythmic punctuation fits a pattern based on the series 2,4,8,16, etc.; and the nuclear melody is performed simultaneously on several instruments at different speeds.

To compose **Fractal Piano 6,** values obtained from iteration of the Verhulst formula where encoded

using non-linear mapping (with a partially "shuffled" scale, or with a selected or weighted element set) into pitches, lengths and loudnesses. This MIDI-data was edited with the help of a commercial sequencer program; stretched and squeezed time-wise, and layered in various ways, using "fractal" structures. For example, the on/off pattern shown in fig. 6 was used as a mask to create fragmented density in one part of the piece. Say the upper register of the piano turns on and off at intervals of 2 sec. When this register is "on", material is audible within this register. When it is "off", it is silent. In a register just under the highest one, the mask turns on and off every 4 sec.; in a register just under it, every 8 sec.; and so on. By applying such a mask over the (potentially endless) chaotic material, I find a kind of musical tension is generated. Notice that the whole mask pattern produces all possible on/off combinations for the chosen number of registers. It is related to the I Ching, with its 64 possible combinations of 6 solid or broken lines.

FIGURE 6 Combinational Scheme for Fractal Piano 6



Flocking animals co-ordinate in a remarkable and still incompletely understood fashion. The reaction time of a group in danger, or in making turns is considerably faster than the reaction time measured of isolated individuals. In order to maintain the proper distances between neighbours without collision, some sort of multi-sensory positive-negative feedback mechanism is in operation. Neural physiology has revealed that the massively interconnected neural network in the individual brain operates with feedback processes. A neuron cell has a main body, an axon from which it receives signals, and treelike extensions called dendrites which branch off in hundreds to make contact with other cells. Connections between axons and dendrites are effected across gaps, called synapses. Neurons send out impulses spontaneously at a rate of about 10 per second. However, the rate of firing changes, and depends on the sum total number and strength of the impulses it receives. There are both excitatory and inhibitory synapses: signals from the former tend to increase the firing rate of a cell, while the latter tend to reduce the firing rate of a cell. The picture of ceaseless electrical activity; signals amplifying, muting, modulating, crossing each other, and returning in loops; all in incredibly complex and indecipherable wave-like patterns: this picture gives us an idea how thought and memory are possible. Recent investigation of the physiology of perception has led to the discovery of chaos in the brain: complex behaviour which seems random, but has a hidden order. Vast collections of neurons shift quickly from one complex pattern to another, in response to the smallest of inputs (remember the butterfly effect). An organism as a whole acts in its environment with feedback mechanisms. The brain seeks information, and sends signals to muscles to place sensory organs in position, and to sensitise parts of the brain which will process signals. A burst of collective patterned activity from all sensory organs is combined to form a gestalt. Then a fraction of a second later, another search for information is demanded. It seems that chaos in the brain is not pathological, as one might expect; but instead is the basis of healthy functioning, indeed explains how the brain can respond quickly and flexibly to an ever-changing outside world. Even what we experience as a original idea (brain-storm) may be derived from a chaotic neural firing pattern triggered in an ever-widening cascade from a small initial impulse.

In my piece **Brain-Wave** for at least 3 recorders of any kind, (1989), I wanted to set up a self-regulating musical situation. All musicians improvise on the same basic material, which is arranged in four cycles, each with four events. (See fig. 7). Performers should stand or sit spread out around the hall, possibly on different levels. Each player should face an arbitrarily chosen direction. Emphasis is placed on influences which performers take from their neighbours. Players(s) in front of an individual give positive influence, and player(s) in back give negative influence. Here is a table summarising these influences:

	Positive influence	Negative influence
If:	player(s) in front are playing,	player(s) in back are playing,
one may:	start to play. play louder.	stop playing. play softer.

play more repeats of an event.	stop, or go to next event.
attempt to imitate style of other player(s).	attempt to play in a style opposite to other player(s).
try to match the speed(s) of other player(s).	play fast when other player(s) are slow, and vice versa.

The object is not to synchronise exactly with the other players, but to correlate what a player does with what the others do. Since there is no conductor, each player must partly assume that function, being attentive to the ensemble sound, and taking initiative to lead that sound were he/she thinks it should go. The aim is to create an interactive situation such as is found in nature, among flocks of birds, or in brain neurons, for example.

In 1989 I completed **The Five Seasons** for 6 percussionists and tape. My inspiration source was an ancient Chinese theory, in which the Seasons, Emotions, Colors, Elements and Directions were grouped as follows:

Spring	Anger	Green	Wood	East
Summer	Joy	Red	Fire	South
Aftersummer	Sympathy	Yellow	Earth	Center
Fall	Grief	White	Metal	West
Winter	Fear	Black	Water	North

This piece incorporates several techniques which were derived from Chaos Science. My adaptation of the Verhulst Model was used to derive some of the rhythmic material for the piece, as in **Fractal Piano 6.** Fractal structures define the form of several sections. Some of the electronic sounds on the tape were made using feedback loops for frequency modulation. The performers are called on to improvise in one section. I'd like to go into more detail with this piece to show how these techniques work.

The first part, **Spring**, begins with accelerandos of accelerandos. First a pulse plan was worked out; the distance between pulses starts large, with successive pulses scaled by a ratio such as 2/3rds down to small (fast) intervals. Then an accelerating figure was fitted to each of the pulses. (A similar slowly accelerating roll is found in Chinese opera and Korean ceremonial music.) There are three layers, played by wood blocks, temple blocks, and log drum. (See fig. 8). Later on, a 16-note theme in quarter-notes is introduced in the bass marimba. The melodic curve of this theme is a projection of a fractal graphic design I made. Here a 4-note melodic motive is fitted or transposed into a blown-up version of itself (See fig. 9). Such nested patterns with scaled elements are characteristic of fractals, as already described. There follows a metric canon; the theme enters in eighth notes, then triplet eighths, and finally sixteenths. (See fig. 10). One can consider the whole construction as a fractal of a fractal, since the theme pattern (itself a fractal) occurs simultaneously at different speeds and octaves. After another metric canon and a section with controlled improvisation, this theme returns with a different treatment. It is split up into 4-note fragments, and given a peculiar "doubling": not parallelism, but an exaggeration of the melodic curve, using multiplication. Again, such "scaling" is a common method of constructing geometrical fractals (See fig. 11).

In the second part, **Summer**, a wiring scheme which includes two feedback loops was used for electronic FM synthesis: the output of any generator provides input control voltages for two other generators. In one loop, when the voltage output of a Low-Frequency-Oscillator (LFO) is high, it causes its neighbour LFO to oscillate at a higher rate. In the other loop, inverted signals are sent out: in this case a high output voltage of a LFO causes a lower rate of oscillation in a cross-connected LFO. The output of all LFO's was used to control other electronic devices, to synthesize a sound. Because of the interconnectedness, and the complex interaction of positive and negative feedback loops, the results of such a circuit can be unpredictable and chaotic.

The **Summer** is divided into four sections, each with a clearly defined instrumentation. Each section and all instruments have similar material rhythmically, generated from the Verhulst Model. There are different scaling factors applied to the material for different instruments playing together in an ensemble, controlling the relative densities of attack.

The third part, **Aftersummer**, uses an on/off masking scheme like that used in **Fractal Piano 6.** Here, not registers on the piano, but six different percussion timbres (all with a sharp decay) are "turned on or off". As in **Fractal Piano 6**, we get all 64 possible combinations of the six elements, and a kind of fractured crescendo. Rolls and repeated notes of various tempi, but always in decrescendo, provide "thematic" self-similarity. (See fig. 6 again.)

The fourth part, **Fall**, depicts musically the "Butterfly Effect", previously described. In the last measures of **Aftersummer**, all six players have finally come together. In the first measure of **Fall**, they play all together again (this time on metal instruments); and then disperse. Sometimes 2 or 3 players synchronize for a while, but small deviations lead to larger separations, and this part ends fragmented and scattered. Loosely spoken, this part is an inversion of how **Spring** begins: **Fall** contains a ritardando of ritardandos.

For the fifth and last part, **Winter**, I used a technique I call "nested repeats" to create the metric structure. Difficult for a human, perhaps, but a computer can easily carry out the following set of commands:

j+[h+[f+[d+[b+[a]+c]+e]+ g]+ i]+k
=jhfdbaacbaacedbaacbaacegfdbaacbaacedbaacbaacegihfdbaacbaacedbaacbaacegfdbaacbaacedbaacbaacegik.

Here, the [ ] signs indicate a simple repetition. **[a]=aa**, for example. The nesting of the repeats makes that **a** gets printed 32 times, **b** and **c** 16 times, **d** and **e** 8 times, **f** and **g** 4 times, **h** and **i** 2 times; and **j** and **k** get printed only once each. Whether you look at a small or large part of the list above, it displays the self-similarity typical of a fractal. For **Winter**, I desired a metrical structure with many changes but internally consistent. First I decided what was going to happen in each measure, in terms of instrumentation and so on, and then let the length in eighth-notes of a measure be determined by substituting the numbers **2-12** for **a-k** in the fractal pattern.

Modi-fications for large marimba and tape (1990) makes use of what I call "transposing modes". These are constructed like fractals, with an interval structure repeated indefinitely. For example, take the interval cell [1,4,2] (a semitone=1). Starting with a low E and repeating the cell, we get E,F,A,B,C,E,F#. Notice that because the elements of the cell add up to 7, two cells don't complete an octave, but overreach it. Indeed, we must repeat the cell 12 times before we get the same pitch-names. Playing "scales" up and down through this mode, we get continuous transposition through the cycle of fifths.

The ordering of much of the material in this piece was achieved with a computer program I worked out called "statistical feedback". A weighted random choice between a string of elements; only the order of preference among the elements is always changing, depending on previous choices. What it does is make a "chaotisation" of serialism. I have used this program on several different element sets in composing the marimba part as well as the tape part of this piece. Pitch, note length and dynamics; as well as larger structures: mode, section length, and electronic timbre; all material and forms are subject to *modification.*

In 1993 I made **Hyperion's Tumble** for tape, using computer algorithms. Observations of Hyperion, a small, irregularly-shaped moon of Saturn, provided some of the first evidence that celestial motion is not merely giant clockwork. With an eccentric orbit phase-locked in 3/4 ratio with Titan (Saturn's largest moon), Hyperion tumbles end-over-end in sometimes periodic, sometimes chaotic fashion, subtly influenced by gravitational forces.

Newton had solved the problem of 2 bodies interacting gravitationally: depending on their energy and mass, they move in perfect curves: a circle, ellipse, parabola or hyperbola. The problem of 3 bodies interacting gravitationally has proved to be surprisingly difficult, and mathematician Henri Poincaré has shown that in the long term, their motion can only be approximated, and is in essence unpredictable. He invented a method to visualise the complicated behaviour of such a system, now called a Poincaré map. A 2-dimensional slice of a three-dimensional phase space will show either one or a few points if the system is periodic, and a complicated figure if it is chaotic. An object with a chaotic phase space might have a degenerate or unstable orbit, causing it to crash into another body, or fly off into infinite space. Close examination of these figures, called strange attractors, proves that they are fractals. Curves are folded into themselves, with infinite regress: increasing magnification shows evermore detail, but with recurring proportional patterns (see fig. 12).

French astronomer Michel Hénon has also demonstrated the theoretical possibility of chaos in the cosmos, when he modelled stellar orbits in galaxies, with the computer. Depending on how the model was set up, stellar orbits would show different behaviours: at low energy levels, orbits were regular ellipses. Higher energy levels gave more complicated orbits, which never exactly repeated themselves, and beyond a certain energy level, the orbits became unstable and unpredictable. He wrote a simple equation to explore the folding and remapping of an oval onto itself, which produces an archetype of strange attractors (see fig. 13).

$x_{new}=y+1-1.4*x^2$
$y_{new}=0.3*x$

Blowing up a strand of the attractor reveals tiny strands within it, spaced from each other in the same ratios as the parent strands.

Two computer programs I wrote based on chaos theory enabled me to generate voltage fluctuations for synthesis. A formula discovered by Barry Martin generates chaotic orbits, two-dimensional plots of which resemble organic structures such as cells under a microscope (see fig. 14).

$x_{new}=y-SQRT(ABS(b*x-c))*SIGN(x)$
$y_{new}=a-x$

Different initial values for the constants **a**,**b**, & **c** result in different patterns and periodicities.
I call the second program the "Baker function". A string of integers is folded into itself recursively, mixing the integers completely. However there are strange periodicities occurring, and eventually the original string mysteriously re-occurs. For example, imagine picking up the integer-string in the first row, below, by the middle. You have the '5' between your fingers, and two ends dangle below. Read off the numbers, starting at your fingers and alternating between the string ends as you move down, to get the second row. Repeat the process to generate the other rows. This process is similar to the one a baker uses to mix dough: flatten with a rolling pin, fold over a half, flatten again, fold again, etc.

1	2	3	4	5	6	7	8	9
5	6	4	7	3	8	2	9	1
3	8	7	2	4	9	6	1	5
4	9	2	6	7	1	8	5	3
7	1	6	8	2	5	9	3	4
2	5	8	9	6	3	1	4	7
6	3	9	1	8	4	5	7	2
8	4	1	5	9	7	3	2	6
9	7	5	3	1	2	4	6	8
1	2	3	4	5	6	7	8	9

Irregular motoric sounds resulted from the transcription of the output of these algorithms. A chaotic sound lies in a spectrum between a sinus-tone, which is perfectly periodic, and white noise, which is perfectly aperiodic. It can have a maximum of complexity, always producing more detail, or information, within certain limits. In principle, computer chaos can be used to model on any level, from musical structure to musical sound.

What holds our attention in listening to music? Music from Bach to Bartók, Josquin to Xenakis, from Bali to Bolivia has some special kind of pattern which hovers in a phase space between repetition and randomness, between association and breaking-away, between order and chaos. Does a *strange attractor* underlie a piece which gives us a feeling of anticipation & resolution, of simplicity within complexity? Do its patterns show resemblance to the fractal geometry identified in nature? Does the new definition of chaos- that dynamics can be paradoxically both deterministic and unpredictable- help us understand how a particular sequence of sounds gives us the feeling both of inevitability and surprise? Perhaps composers as well as scientists may do well to take a new look at Chaos.

June 2, 1995

David Clark Little (1952, USA). After receiving a BS in chemistry, he studied harpsichord, finishing with Gustav Leonhardt, and composition with Ton de Leeuw, in the Netherlands. He has been a finalist and prize-winner in several composition competitions, including in the USA, Germany, France and Greece; and has been given many grants, for example to attend festivals and workshops in Germany, Holland, and the Soviet Union; and has received many commissions for compositional work. Since 1988 he has worked on compositional methods using the computer and based on the new "chaos science" and "fractals". Scores of his music are available from Donemus, Paulus Potterstr. 14, 1071 CZ Amsterdam, the Netherlands.

References:
Dewdney, A. K.: **Wallpaper for the Mind;** Computer Recreations, September 1986 Scientific American
Freeman, Walter J.: **The Physiology of Perception**, February 1991 Scientific American
Gleich, James: **Chaos, Making a New Science**; 1987 Viking
Hawking, Stephen W.: **A Brief History of Time;** 1988 Bantam Books
Hofstafter, Douglas R.: **Godel, Escher, Bach;** 1980 Vintage Books
Hofstafter, Douglas R.: **Metamagical Themas (Mathematical Chaos and Strange Attractors)**; November 1981 Scientific American, and 1987 Penguin Books
Jones, Jesse: **Fractals for the Macintosh**; 1993 the Waite Group, Inc.
Lauwerier, Hans: **Fractals**; 1987 Aramith
Mandelbrot, Benoit: **The Fractal Geometry Of Nature**; 1983 W. H. Freeman and Company
Manzolli, Jonatas: **FracWave Sound Synthesis**; (unpublished paper)
Peterson, Ivars: **Newton's Clock, Chaos in the Solar System**; 1993 W. H. Freeman and Company
Prigogine, Ilya: **Order out of Chaos;** 1985 Fontana Paperbacks
Russell, Peter: **The Brain Book;** 1979 Routledge
Stewert, Ian: **Does God Play Dice? The Mathematics of Chaos;** 1989 Basil Blackwell Ltd.
Watson, Lyall: **The Dreams of Dragons;** 1987 Morrow, New York

Figure 1 The Butterfly Effect

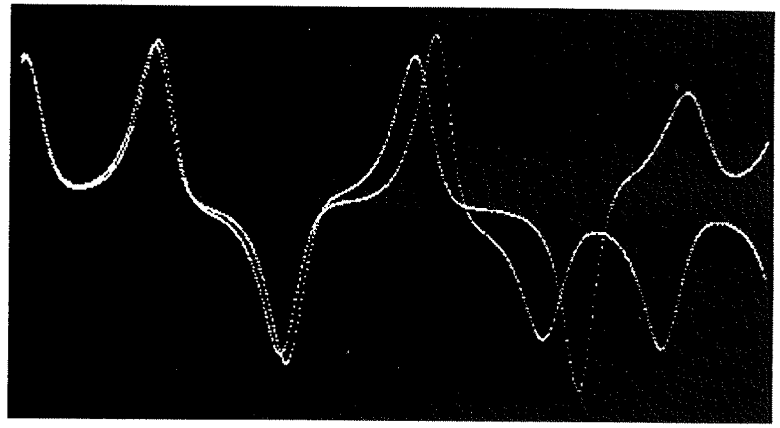


Figure 2 Lorenz Function

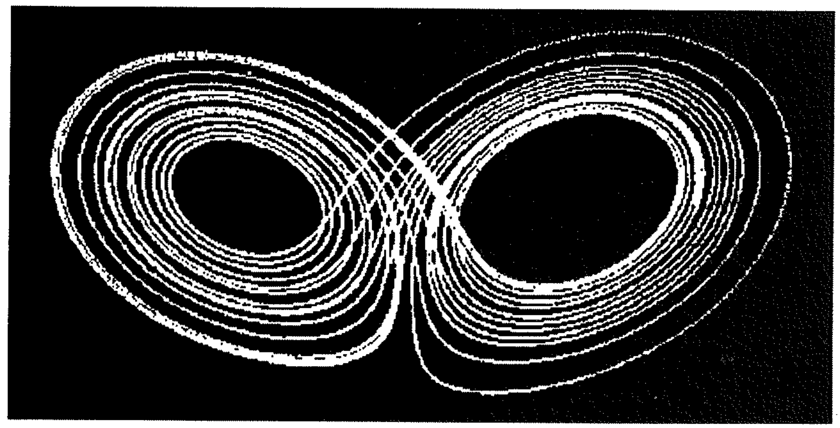


FIGURE 3 Harpsi-Kord p.1

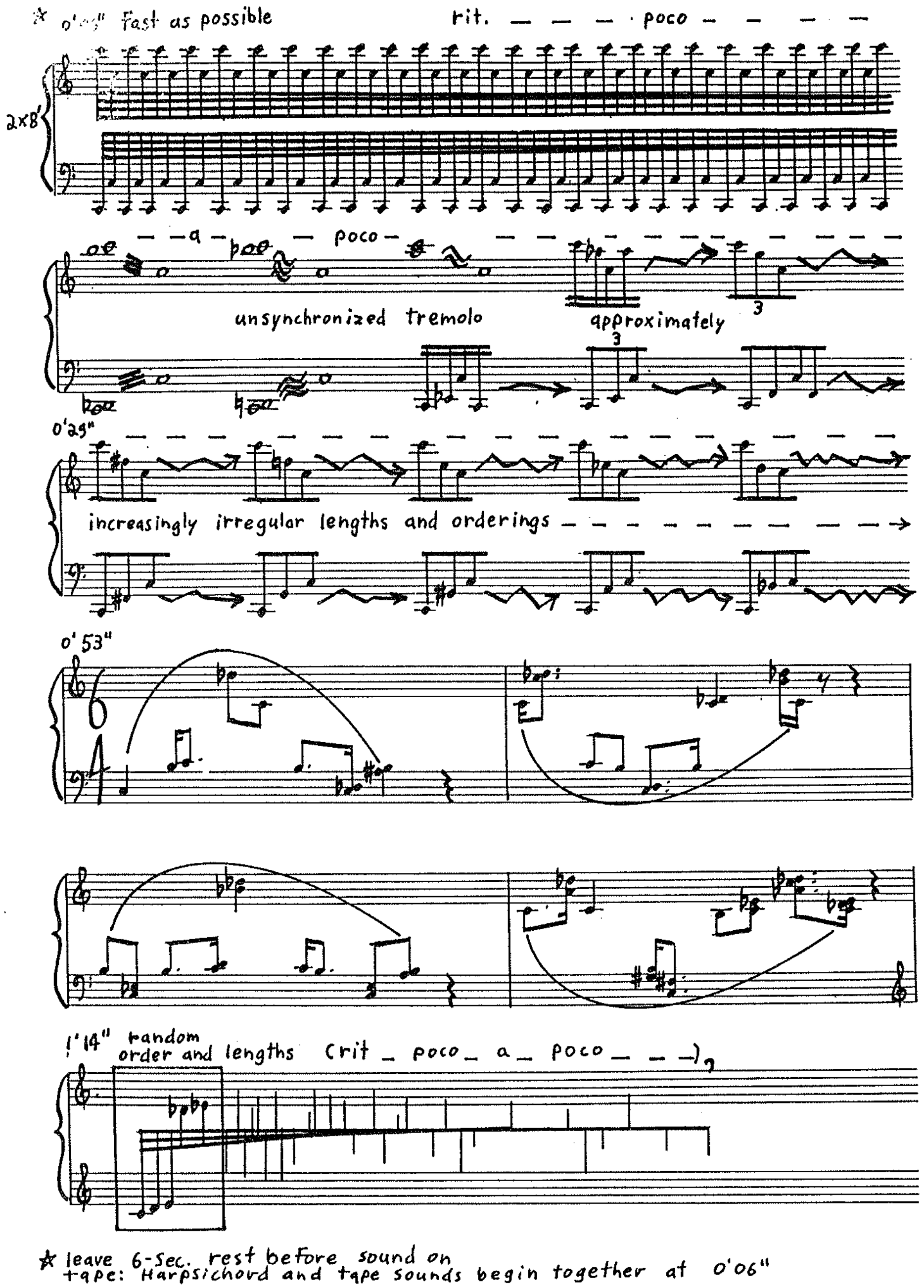


FIGURE 4a-d Verhulst Model

Figure 4a, Verhulst model, a=3.2

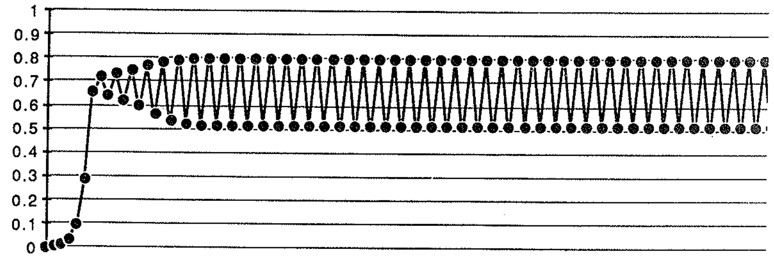


Figure 4b, Verhulst model, a=3.68

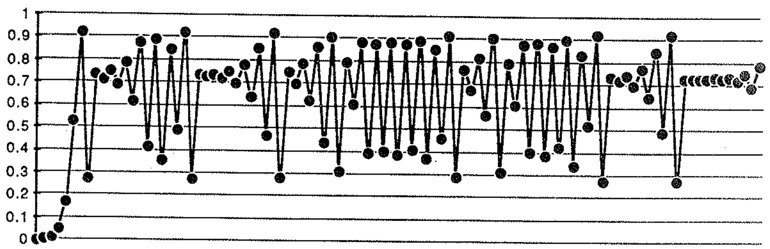


Figure 4c, Verhulst model, a=3.83

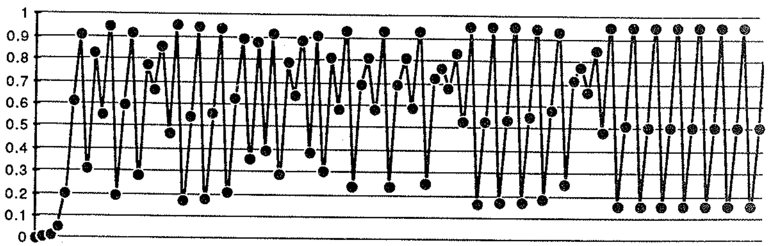


Figure 4d, Verhulst model, a=4

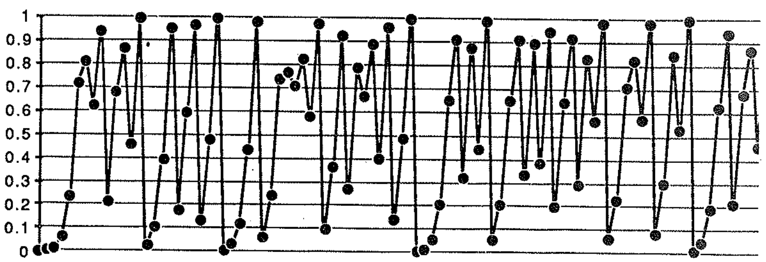


Figure 4e Verhulst Model: all "a" values along x-axis

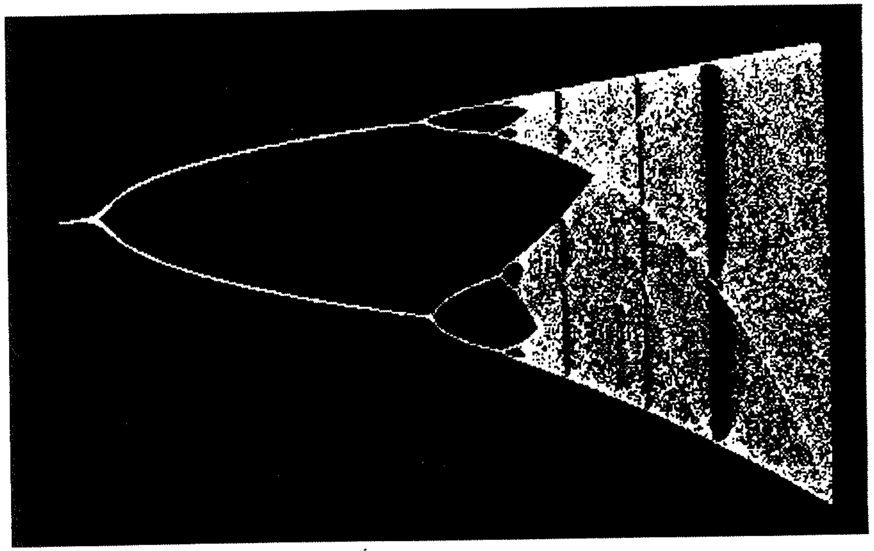


Figure 4f Verhulst Model: all "a" values along x-axis (detail)

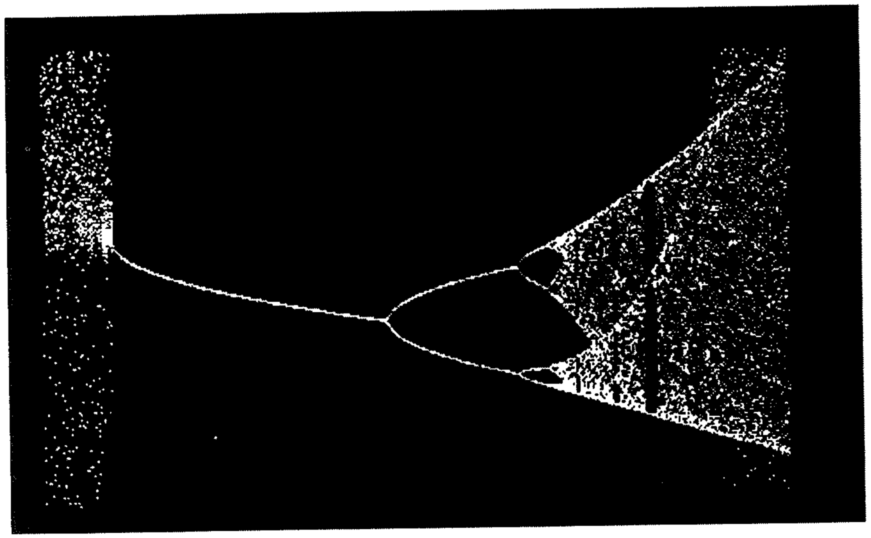
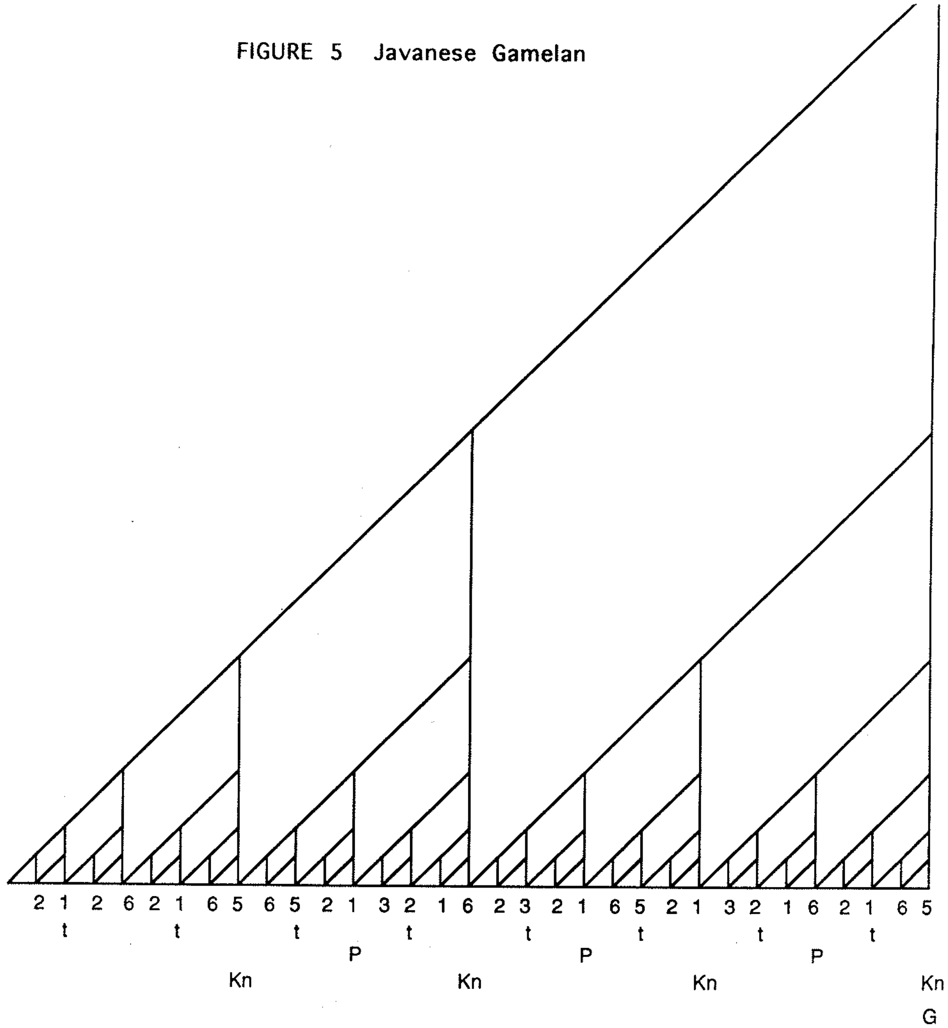
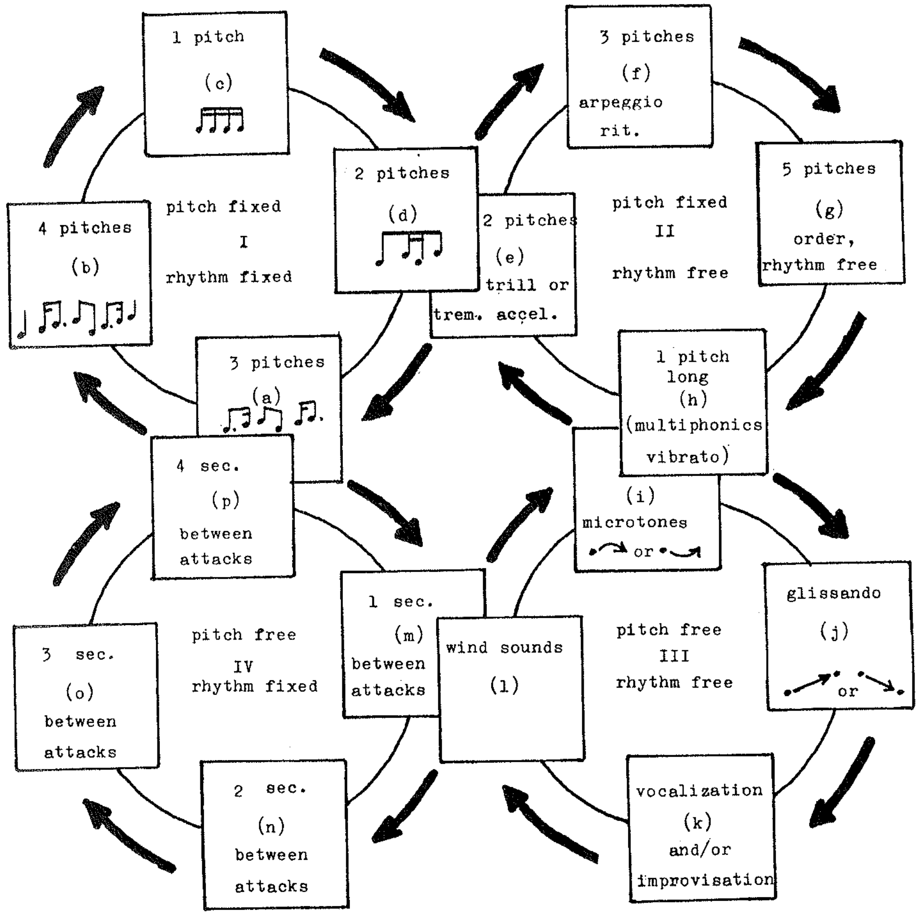


FIGURE 5 Javanese Gamelan



2 1 2 etc. =Balungan (nuclear melody) played on Saron/Slentem
t =Ketuk
P =Kempul
Kn =Kenong
G =Gong

FIGURE 7 Brain-Wave



1990 David Little

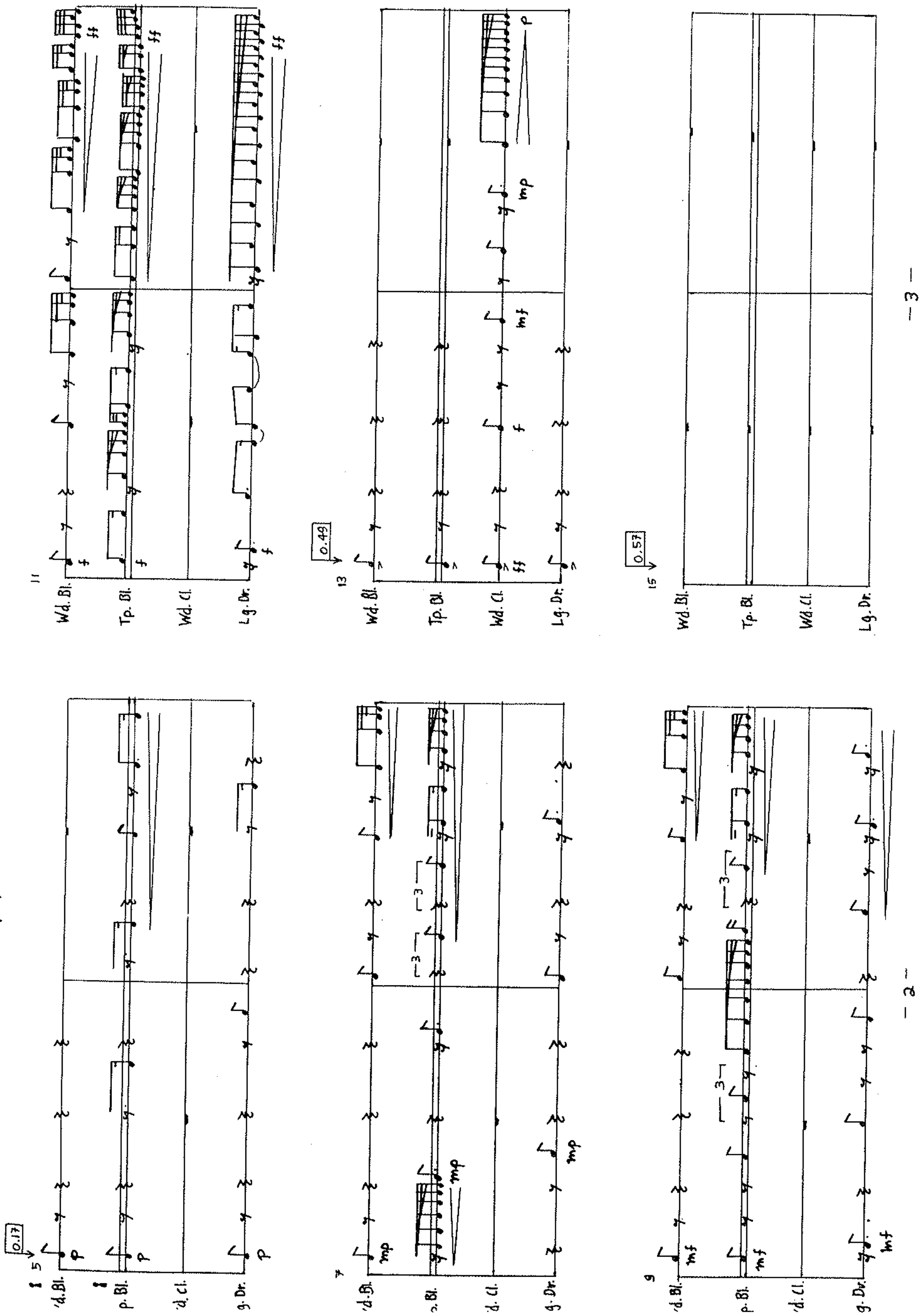


FIGURE 8 The Five Seasons p.2,3

FIGURE 10 The Five Seasons p.7

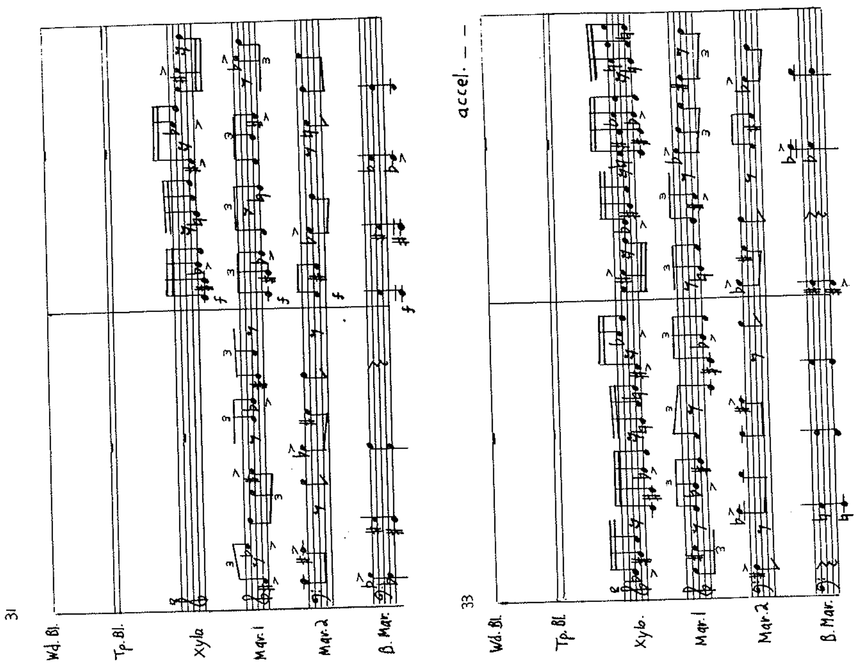


FIGURE 9 The Five Seasons (theme)

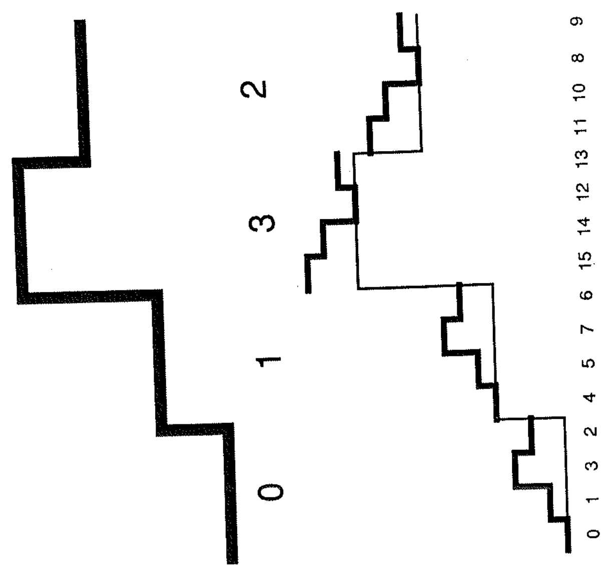
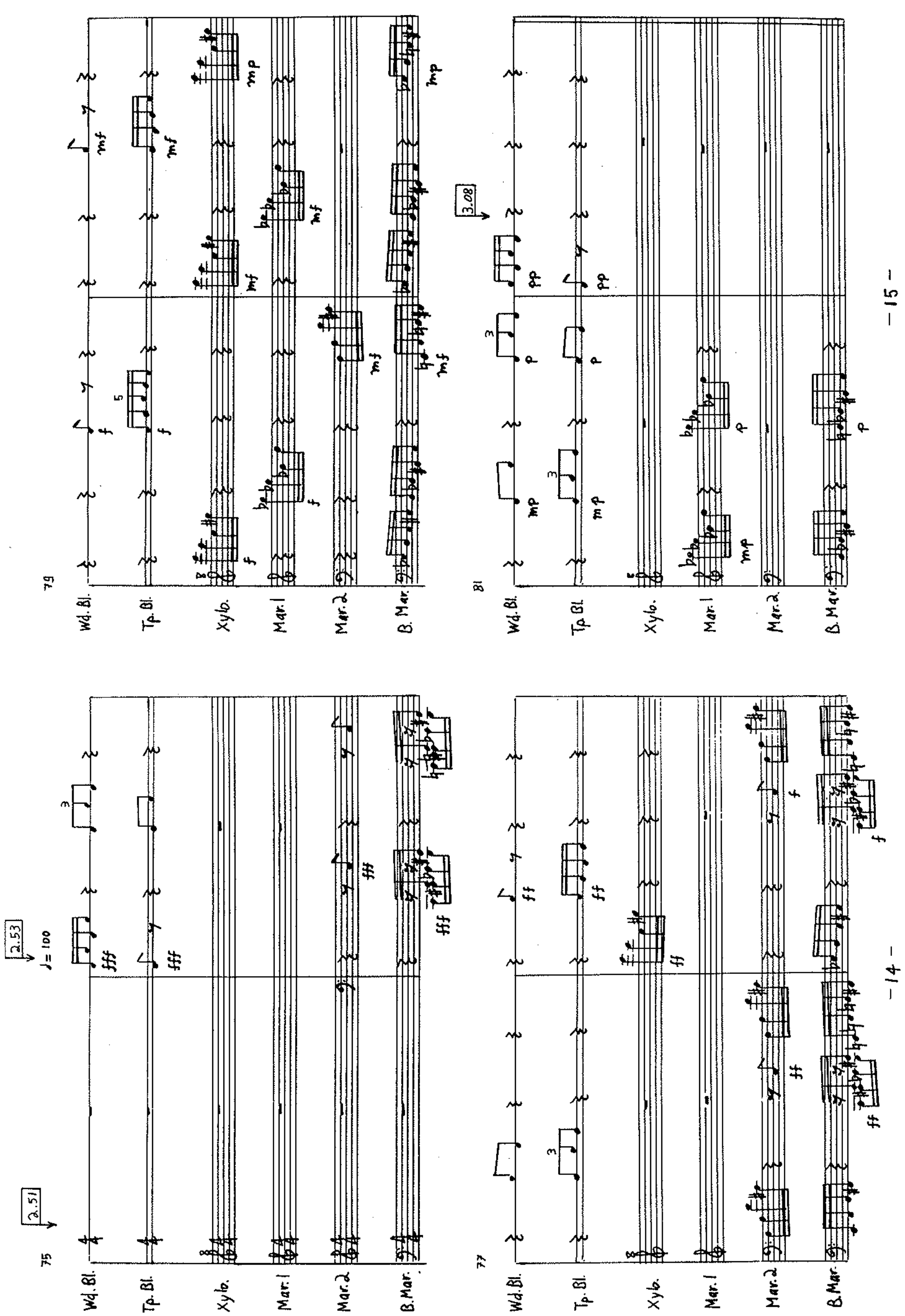


FIGURE 11 The Five Seasons p.14,15



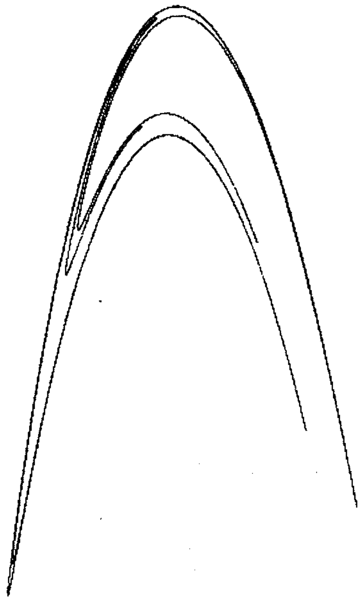


Figure 13 Henon Attractor

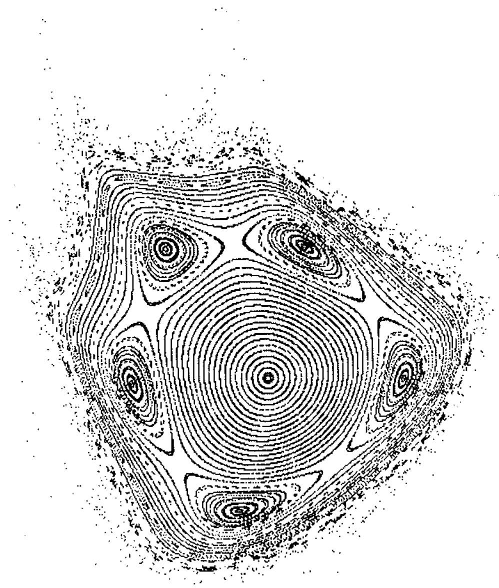


Figure 12 Poincaré Map

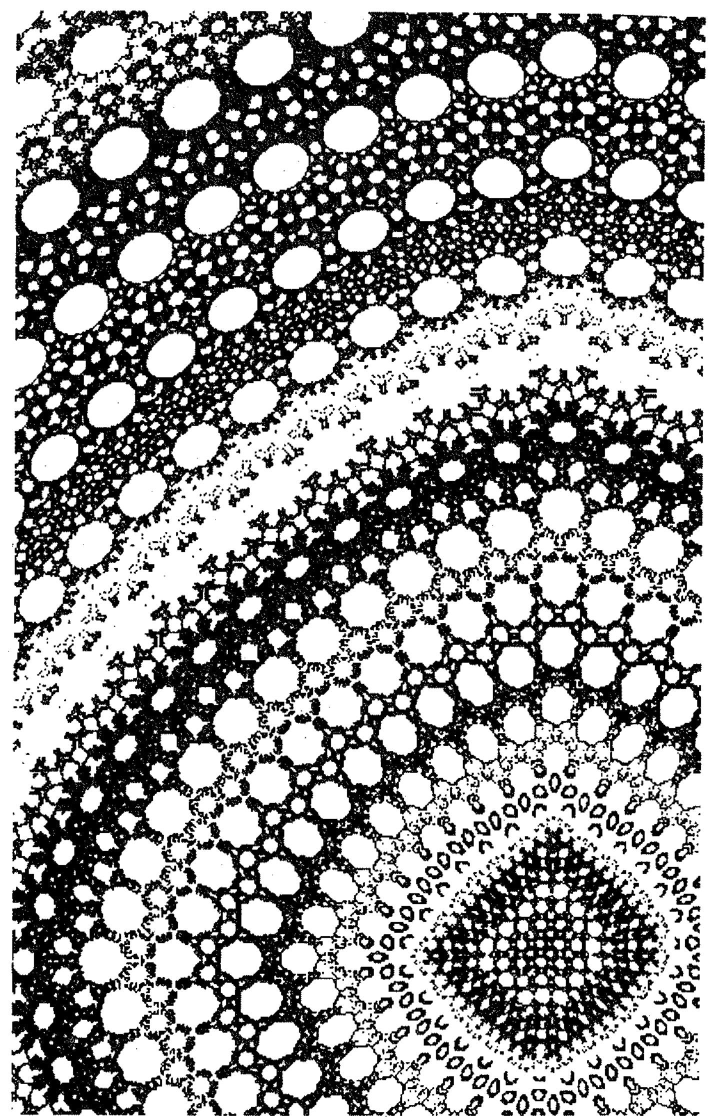


Figure 14 Barry Martin Formula

Images from the Aperiodic Time

J.J. GARCÍA ESCUDERO
Departamento de Física, Universidad de Oviedo, 33007 Oviedo, Spain

Abstract

A diffraction image in solid state physics is a measure of the order inside a distribution of points (masses). The image is related with the Fourier transform of the distribution. Some 1D aperiodic distributions have discrete Fourier components. We use this models in order to structurate the time in a non periodic way with discrete frecuency spectrum. The Fibonacci sequence ilustrates this aproach.

1.Introduction.

The scientific analogy has been very fertil in the evolution of musical thought. In particular Mathematics and Physics are among the disciplines more influent. A composer who wants to learn about a scientific model will go beyond it in order to show his own aesthetical view. But that knowledge before musical experience may open new expression possibilities that can not be anticipated by intuition.

A musical theory must be linked to a theory of the musical time. In this work we want to show how a new type of discrete structures introduced and extensively studied in the last decade in the domain of physical sciences can be used in order to structurate the time in a non periodic way. The order contained in the temporal structure (rithm) is reflected in the frecuency space (harmonic fields) through the Fourier transform.

2.The continuum pitch-durations-forms.

If we take a duration defined by two impulsions, we repeat it and then the duration is decreased until it lasts 1/16 of second, the impulsions can be heard separately. But if we continue decreasing the duration until 1/32 we begin to hear pitch (B_0 aproximately). With this experience (Stockhausen 1963) it can be seen how the same basic process is behind our perception of duration and pitch. On the other hand, durations higher than, say, 8 seconds are related with the form and its articulations.

A first attempt to extend in a large scale what happens inside the sound is to consider the very well known decomposition of a periodic function in a Fourier series. Let's take the following periodic function

$$M_a(t) = \sum_{n=-\infty}^{\infty} \delta(t - na)$$

where the Dirac delta-function $\delta(x)$ has the properties: $\delta(x) = 0$ unless $x = 0$, $\delta(0) = \infty$. The function $M_a(t)$ represents a periodic structure in one dimension with period of length a. If t denotes the time (in seconds) then the Fourier transform of $M_a(t)$ is proportional to $M_{\frac{2\pi}{a}}(\omega)$ where ω denotes the frecuency (in Hertz). For instance if we take $a = \frac{\pi}{33}$ we get the harmonic series:

$$C_2, C_3, G_3, C_4, E_4, G_4, Bb_4, C_5, D_5, E_5, \ldots$$

all of them with the same intensity which is measured by the proportionality factor of the function $M_{\frac{2\pi}{a}}(\omega)$.

Timbre is generated by summing up harmonics: fragmentation into microdurations of a macroduration defined by the fundamental. If we take timbre as a model for the musical form we can think in a musical piece generated by fragmentation of a global duration. This approach allows to structurate the musical time in a continuum pitch-durations-forms. But if a function is not periodic (representing a timbre of musical interest) it doesn't have in principle a Fourier series decomposition and it is necessary to use the Fourier integral which contains the continuum of frecuencies and not only integer multiples of a fundamental frecuency. This decomposition can not be incorporated into the musical writing which belongs in essence to a discrete domain.

In the next section we study an example of an aperiodic discrete time structure with discrete frecuency spectrum.

3.The Fibonacci time structure.

Aperiodic systems are in some place between order and disorder. A relevant 1D example is the Fibonacci chain with a distribution of points along the real line according to:

$$x_N = N + \alpha + \frac{1}{\phi}[[\frac{N}{\phi} + \beta]]$$

where $[[x]]$ is the greatest integer less than x, $\phi = \frac{1+\sqrt{5}}{2}$ is the golden number, α and β are arbitrary real numbers, and N is a non-negative integer. This equation describes a sequence of points such that the interval between two consecutive points can be only of two types: $L = \phi$ and $S = 1$ which appear in a non periodic sequence where the ratio of the number of L-intervals to the number of S-intervals is also equals to ϕ.

An equivalent way to define the Fibonacci sequence is through a formal grammar. Consider the alphabet $\{L, S\}$, the production rules $L \longmapsto LS, S \longmapsto L$ and the axiom L. The language consists in the words $L, LS, LSL, LSLLS, LSLLSLSL, \ldots$

The Fibonacci chain can be used to structurate the musical time in a non periodic way. Although the golden number is not rational we can take rational approximants for it .If we define the Fibonacci numbers F_n as $F_{n+1} = F_n + F_{n-1}$ with $F_0 = F_1 = 1$ it is very well known that $\frac{F_n}{F_{n-1}} \rightarrow \phi$ when $n \rightarrow \infty$. The sequence $\frac{2}{1}, \frac{3}{2}, \frac{5}{3}, \frac{8}{5} \ldots$ can be used to generate aperiodic rithms with two rithmic units L and S in a rational ratio.

The spectrum of a sequence of points in a line is a sum of discrete and continuous components. The discrete component indicates order, the continuous component disorder. The Fibonacci sequence has only a discrete part and can be computed with the help of the golden number ϕ and two integers p and q through the following expresion (Levine and Steinhardt 1986):

$$\omega_{pq} = \frac{2\pi}{1+\frac{1}{\phi^2}}[p + \frac{q}{\phi}]$$

and with a different intensity for each component. If we take $X = 2\pi q - \omega_{pq}/\phi$ then the intensity is proportional to

$$\frac{4sin^2\frac{X}{2}}{X^2}$$

A pitch defined by p and q is more intense if $\phi q - p$ is small or p/q close to ϕ , that is when (p, q) are successive Fibonacci integers (F_n, F_{n-1}). Outside this sequence the intensities decrease strongly. If we distribute the pitches more intense according with the intensity the following harmonic fields are obtained ($p, q = 1, 2, ...20$, and the frecuencies are scaled by a factor ten):

$$pp : \{D\flat_5\}; p : \{A_3, C\sharp_4, G_5, C\sharp_6\}; mp : \{F_4, B\flat_4, F_5\}; mf : \{G_2, D_3, B\flat_5\};$$
$$f : \{F_3, G_4, A\flat_4, E\flat_5; E_5, A_5, E\flat_6\}; ff : \{D_2, B_3, E\flat_4, B_4, D_5, F\sharp_5, A\flat_5, C_6, D_6\}$$

4.Conclusion.

The Fibonacci sequence is an example of the great variety of temporal structures we can get by means of aperiodic systems in 1D. We can think in the pairs intensity-pitch as the Fourier spectrum of aperiodic rithms generated automatically. These sequences have another interesting propierty: they are selfsimilar. There exists a transformation in which each interval is subdivided into pieces that can rejoin to form a new sequence with all intervals scaled down by a factor ϕ. This hierarchy can be used in order to articulate the global musical form.

Models based on aperiodic systems have been used by the author in works like *Moradas* for organ, *Nocturno* for sopran and ensemble, *Imágenes* for two pianos and others.

References.

Levine, D. and Steinhardt P. (1986) Quasicrystals.I.Definition and structure. *Physical Review B.* **34**,596.

Stockhausen K.(1963) ...wie die Zeit vergeht... *Texte zur elektronischen und instrumentalen Musik. Bd.1.* Dumont Buchverlag, Köln.

THE NECESSITY OF COMPOSING WITH LIVE-ELECTRONICS
A short account of the piece "Gegensaetze (gegenseitig)" and of the hardware (AUDIACSYSTEM) used to produce the real-time processes on it.

JAVIER ALEJANDRO GARAVAGLIA
Hendrik-Wittestr. 11
45128- Essen.
Deutschland. (Germany)
Tel: (0049-201-229831)
E-Mail Address: gara@folkwang.uni-essen.de
ICEM- FOLKWANG-HOCHSCHULE ESSEN
Klemesborn 39 / 45239- ESSEN. Germany

ABSTRACT

The aim of this paper is to speak about my piece -Gegensaetze (gegenseitig) [Contraries (reciprocally)] for alto flute, 4 Channel-tape and live electronics (1994)- making an account of how and why the work was concieved. The hardware-and-the-software environments which are responsable for the real-time processes (AUDIACSYSTEM, a project carried on by the ICEM (Institut for Computer music and electronic Media) at the Folkwang Hochschule-Essen and by the company Micro-Control GmbH & Co KG, both in Germany) will be described. Finally some examples and passages of the piece will be explained.

"CONTRARIES "

"Gegensaetze (gegenseitig)" was the result of an idea that I have had for a long time: to compose a piece in which contraries should be shown not only against each other (in a negative way), but also that they could be able to build some kind of unity by creating something completely new, constructive and positive.

My first problem was how to put this into music without using a text about the subject. At the beginning I simply wanted to make a contrast between a normal instrument and a prerecorded tape, but it didn't seem like being the solution to the problem because it could actually show only the contraries themselves but not the reciprocal action of both elements. The instrument should make with the electronic something really new and this should happen in real time and not with recorded material. That was the reason why I first began to work on the tape itself, making sounds with two Yamaha synthesizers (TX 802 - TG77) that shouldn't have any relation with normal instruments. I composed then a previous piece for stereo-tape alone, from which I took the materials for the definitive version of the work. When the tape materials were selected, I knew already that the instrument should have to be a very soft one, an the election was that of an alto flute. How should then the "reciprocal action " look like? I was now pretty sure that it should be performed with live-electronics. This decision conduced me to the next problem: what type of live-electronics did I really want and much further, which kind of system should I use?. There are basically two ways of working with live-electronics: on one side, those whose aim is to create a new conception of how the live instruments could be projected into a particular space or room, normally using only echoes and delay lines; on the other side, the more complicated ones, in which the sound will be actually processed in real-time (through FM, AM, filters, envelope generators, envelope-followers, tranpositions, etc) up to the point in which the instrument itself could be no longer recognizable. At the ICEM of the Folkwang Hochschule in Essen (Germany), there's no IRCAM board, but there's a completly different project, which has been carried on since eight years at the ICEM by a group of german composers and ingeneers. This project is the AUDIACSYSTEM, about which I shall speak later in this lecture.

Once I had already got the three Instrumental groups (alto flute, 4-channel tape and the 4-channel live-electronics), I wanted to prosecute composing each parameter (from the micro-up to the macro-structures) with the same concepts of THESIS-ANTITHESIS working together to create something new, so that at any point of the piece the main idea could be shown. For this purpose, I've chosen very empirically two principles which are opposite to each other: a "single-principle" and a "totality-principle". Both principles should have to be the main generators of every event throughout the work and are mainly represented everywhere in the piece by two objects: "glissando-object" representing the "totality-principle" and "a single-note-object", representing the "single-principle".

For the whole structure of the work, I have chosen a numerical-row, whose first four components were explicit selected by myself, but from the 5th component on, they should always be the addition of the last three numbers (that means that the next figure in the row, will be constituted with the reciprocal action of the former three). It comes as result a bigger new value that stands as a contrary to the first, for example, the row begins with (1 1 3 5), which are the numbers that I arbitrary selected; the next value will be 9 (1+3+5), the next 17 (9+5+3) and so on. Each single element contributes to make a partial new totality. This row plays an extremely important role in the composition of the pitches, rhythms, metronomic values, form, and the stage-production, as well.

The form of the piece consists of 5 parts, each one showing the principles already mentioned:

1- Solo alto flute ("single-principle")
2- Alto flute + Tape (as opposites)
3- Only Tape ("single-principle")
4- Alto flute + Tape + live-electronics ("totality-principle"- reciprocally action of all three Instrumentals)
5- Only live-electronics ("single-principle" as result of the reciprocally action of all three Instrumentals)

The rhythms have been composed with the numerical row too. There is a unit value which is the sixteenth, which will be multiplied or divided with the numbers 1, 3, 5, 9, in all possible combinations within these 4 numbers (for example, ratio 9:5 means that 9 equal durations should be instead of 5 sixteenths; ratio1:3 results in a dotted eight,etc).

The stage-production is also supposed to work with contraries. The stage should only be iluminated when the flautist has to play (parts 1,2,4 and 5). In part 3, where only the 4-channel-tape is present, the whole stage and the whole hall (if possible) should be dark.

The material for the pitches has been derived from a chromatic scale beginning with the pitch g3 (the deepest note for the alto flute in G), representing a whole or totality object, a metasymbol of the "glissando-object ". This object plays one of the most important roles throughout all parameters in the piece, not only for the flute-part, but also and mainly for the tape and the live-electronics. The process of genereting the whole pitches for the flute part are produced by an algorithm that eliminates some notes in such a way that at the end, there's only one pitch left. The result is a process going from the whole (all 12 tones) up to ONE SINGLE element, generating a tension between the two main principles mentioned above. The pitches which were eliminated, will be used later in part 4, in the form of 3 improvisations, in which only the rhythms are totally free. These improvisations make a counterpoint to the live-electronics and even modulate them, as it actually happens in the third one.

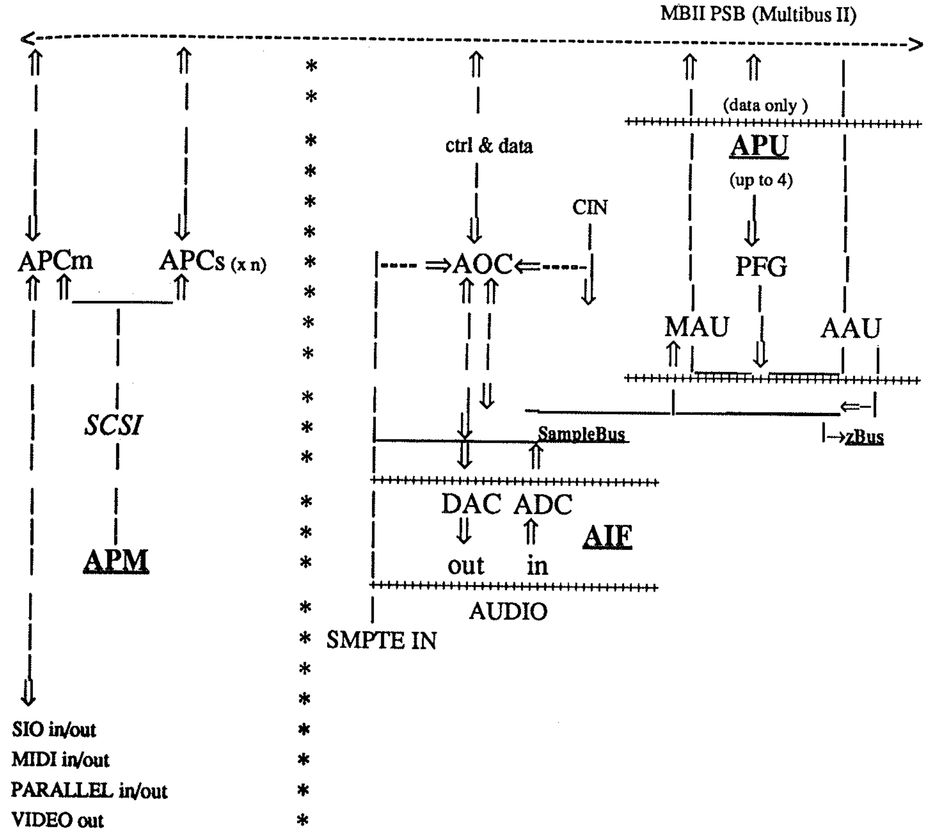
The whole 4-channel-tape part has been worked up with many different methods, for example CommonMusic, transpositions and filters (mostly with Sound Designer II), echoes and even with the AUDIACsystem itself. This twenty minutes long 4-channel tape makes at its begining a counterpoint to the alto flute, than develops itself alone and at last must fade out very slowly as the live-electronics start. I think that at this point, the time has come to make a short description of the AUDIACsystem and its application for the live-electronics in the piece.

THE AUDIACSYSTEM

The AUDIACsystem is a project developed at the Folkwang-Hochschule in Essen (Germany) by the ICEM (Institut for Computer music and Electronic Media) and the Micro-Control GmbH & Co KG. The people involved in its whole design are: Dr. Helmut Zander, Dipl. Ing. Gerhard Kümmel, Prof. Dirk Reith and the composers Markus Lepper and Thomas Neuhaus. The whole began in 1987 and attaches not only the hardware architecture, whose specially designed Audio Processor Unit has got still today the power of 2,5 Pentiums - naturally regarding only the audio processing capacities- but also the software itself, which was exclusively created for this particular environment. The hardware configuration employed in my piece should be contemplated today as an already finished stage of its own development, because almost the whole is going to be actualized, replacing the current design with a new one, which shall result in a chain of Pentiums or most probable P6s, adquiring a RISC- processor configuration and making the whole a bit smaller than today's one cubic meter , possibly making it also compatible with a Power-PC.

HARDWARE CONFIGURATION

The hardware configuration of the AUDIACSYSTEM is shown on the following schematic representation:



The hardware architecture of the AUDIAC has been concieved with the principle of the specialized subsystems. It has not only been made to generate organized forms for the musical production, but also incorporates the generation and working up of sounds in real-time. The whole implies a hughe measure of different demands in relation of its computing potential, which can only be solved with the above mentioned subsystems and their communication capacities.

The whole system could be described as the cooperation of a **"von-Neumann"** unit on the one side and a **Signal-processing unit** on the other. The former percieves configurations (devices), control and driving functions, which steer the processes of generating and working up of sounds from the latter. The communication is guaranteed with the help of the Multibus II. The "von-Neumann" part consists of a Manager (APM) and one or more control units, the APCs. Both do communicate via SCSI.

The APM (Audio Processor Manager) is a 486 Computer with 66 Mhz clock-rate, where the software specially designed for the AUDIAC is implemented. This software is the language APOS which means Audio Processing Operating System and which was specially created by the german composer Markus Lepper for this purpose. APOS pursues three goals which are:

1- a monolithic system architecture, in terms that every hard-and-software levels could be described with the same language, from an individual bit of the hardware up to very complicated abstract compositional models;

2- an enlargable antrophomorphic surface, in the sense that each composer can use not only algorithms that are already defined but also can implement his own language for a particular use as well;

3- an abstraction from the technical necessities, meaning that composing should be allowed on a symbolic level, without caring about technical details.

APOS is an object-orientated language that works with two levels of interpreter : an outer interpreter, which recieves the information in ASCII code, and an inner interpreter, which reads a row of object-references, which are references about objects that already exist and could be recognized as such. The software runs in protected-mode because of memory management reasons, and makes possible that some kind of tasks -which are necessary for the actual configuration of the system- can be percieved.

Regarding the APCs (Audio Processor Controller), the system can afford from only one up to four units. These are all 186 computers which, due to the ATOS kernel (a real-time operating system kernel specially developed for musical applications) has got many functions at their disposal, which are needed for the multitasking operations. The ATOS configurations are created on the APM in APOS and will be later called by the APC, generating or working up sounds. The APC and the Signal processor run asynchronously. The heart of the APC is the APU (Audio Processor Unit), the real Audio processor. Beside it, there are a number of auxiliary units, such as the AOC (a unit capable of transfering data and time code between the APUs, also from one to other two simultaneously, and which could be programmed separately; the CIN (a low control interface with a 16 times multiplex A-D converter, through which up to 16 control voltage units could be brought in); the AIF (the A-D and D-A converters). The APU consists of one Memory Unit (MAU=Memory Address Unit) and an arithmetic unit (AAU, a multiplier). It is possible to put up to 4 APU plus one AOC together, connected through a z-bus. The data could be read and written on the Multibus II. The two memories of the APU (XMY and YMY) can be addressed alone or parallel. The in-and-out sample ports work with the Fifo principle and connect the APU with the out world through the A-D and D-A converters. The interface has 2 inputs and 4 outputs, which could be enlarged up to 32 and 64 respectively. The computing processes run parallel, that means that it could make up to two additions (or substractions), one multiplication, twice read and write from and to the D-RAM (or four times from the S-RAM) at once. The flexible handling of the signal processing unit is guaranteed due to its totally free way of being programmed. The synthesis or working up of sounds result from micro-programms specially developed for this APU.

The Parameter-Functions-Generator (PFG), which is a computing unit in itself works within the APU. It is coupled on one side to the APU and can (due to its complexity) be seen as an independent unit. Its multiple possibilities of application could be resumed in the providing of control instruments for the manipulation of sound: envelopes, spectral control, sound intensity, etc. For each parameter to be controlled, there could be placed pro time-unit one "value-pair" plus a bit-control. Each sample of every four could take a new PFG value. There are altogether 128 PFG free for each APU. The PFG has basically two operating modes: one, in which a "value-pair" INC/FIN makes a linear interpolation, building an envelope which makes a continuous alteration of the y values through the time axis; the other, which interprets a "value-pair" y-*dt* , where y takes one value and *dt* represents the duration of it, building discrete values. The control-bits allow a flexible and interactive influence to the corresponding value rows, for example: back to the first value, mode switch, segment switch,interrupt and hold function (fermata). Interrupts are possible in the first operating mode over each FIN value; in the second mode, at the moment of any new y-value. Through the use or the interrupt features, new support values can be called, resulting in more support values for only one parameter function.

<u>MICROCODES</u>

The biggest time unit to take account of is that of the Sample-rate. The time between two samples will be called "MINICYCLE". There are multiple "cycle calculations" within such a "MINICYCLE" which are coordinated to different process channels (PROK). One cycle calculation can be divided into a given number of microcycles, which correspond to that of the machine rate, which normally is normally set at 10 MHz. All calculations necessary for the generation of a sample must occur within a single "MINICYCLE". The cycle will be finished with a reset signal, which guides to the next step, that is the D-A conversion. With a sample-rate of 48 kHz., the duration of a "MINICYCLE" comes up to around 20 micro-seconds.

WORKING WITH THE AUDIAC

The way in which the input data can be programmed, may be defined in two different forms: on one side it could be done algorithmically; on the other side however, a specially precomposed material could be later imported to the system. Both possibilities don't exclude each other, but could be mixed throughout a

composition, which is actually the case in my piece. The resulting Score can be defined anew in two different ways: <u>statically</u>, creating discrete values for the structure, or <u>dynamically</u>, in which the begin and end of each event is particularly significant, because any kind of process can be programmed between both extremes (for example, transpositions, dynamical filters, etc). This data will be then translated, resulting in a row of orders to be interpreted and fulfilled.

Coming back to my piece, the around 13 minutes long live-electronics part is divided into three different groups: "LA", "LB", "LC", "L" meaning in this case *"live"*. For the programming in APOS, I had got the unvaluable help of Markus Lepper, who I 've already mentioned as the creator of this language.

Regarding the first part,"LA"-with a tempo of quarter equal 50 and measure 3/4- the AUDIAC has to record three different types of single events played by the alto flute, namely: breath-out-noise, one multiphonic and a row of slap tones played separately. They will be given back with intervals of 9, 5 and 3 quarters (proportions taken from the numerical row, which I spoke about in the first part of this lecture), rotating from one channel to the other against the direction of the clock needles, in opposition to the Tape's channel distribution, which is the ordinary one (1. front-left, 2. front-right, 3. right-back, 4. left-back)

The recording of the different sounds is made through an object defined in APOS as "Recorder", which works in such a way as a normal recorder. It has a begin- and an end-buffer-time, an amplitude value, etc. The samples recorded will be played by another object, the "Player" which also has a begin- and an end-buffer-time, an amplitude, an input to vary its frequency ("FINC" transposing the sample) and an input to loop the sample from a given buffer-time-point. There are four players, each one corresponding to each one of the four channels. Each recorded sound becomes a different memory address, so that it could be called at any time. For the time allocation of these events, the computer was asked to find the best possible distribution through all four channels between space and time, in order to force the events to meet quite often at one same channel. When this actually happens, one event will multiply the other, modulating each other (Amplitude Modulation). When all events (once breath-out-noise, once a multiphonic, and five times different slap-tones) have been played and recorded, the computer begins to <u>transpose</u> the information of 3 of the 4 players with different ratios (which are taken from the numerical-row). This transposition, made through the input "FINC" of each "Player", takes place dynamically, that means that within its time limits given in the score, the frequency will be varied every fourth sample, making "glissando-structures".

For "LB", there are two moments to be recorded, both 12 seconds long. This part makes a formal "crossfade" with "LA", and is all about transpositons on all 4 channels of both recorded materials. These transpositions, however, are not dynamical, but discrete. The first of the two recorded materials of "LB", must be further stored, because it will be used in the next part "LC".

The score for "LB" is programmed half algorithmically and half precomposed, as the following APOS example shows:

```
new plstarts "pls2"  200
*  open pls-kanal 0
*;               ANZAHL       ABSTAND
*;               EINSAETZE
*  put pls2      1    *   17
*  put pls2      3    *    9
*  put pls2      5    *    5
*  put pls2      9    *    3
*  put pls2     17    *    1
*
*  apl pls2 110 to 150 [ if [[_1 mod 21] ?eq (110 mod 21)] ['@ .p0 _0 ok] ]
*  apl pls2 111 to 150 [ if [[_1 mod 19] ?eq (111 mod 19)] ['@ .p1 _0 ok] ]
*  apl pls2 112 to 150 [ if [[_1 mod 15] ?eq (112 mod 15)] ['@ .p2 _0 ok] ]
*  apl pls2 113 to 150 [ if [[_1 mod  9] ?eq (113 mod  9)] ['@ .p3 _0 ok] ]
```

All these lines describe every starting point of every of the four players. The last four lines use an explicit indication (precomposed) of how the structure should finish; on the other side, the "put" lines use an automatic way of creating the starting points with an special syntax implemented for this purpose. This syntax will be implemented with the following APOS source text:

```
new latch "pls-kanal"
*  new latch "pls-Position"
*
*; 4 new methods are going to be defined for this purpose (dm)
*
```

```
*   dm [ put (any plstarts) @ (any integer) .p (any integer)
*>                            (any integer) * (any integer) ]
*>     [ apl 0 to [pred _6]
*>               ['@ pls-Kanal    ['['_5 + __0] MOD 4] ;
*>                 @ pls-Position [' _3 + ['_8 * ['SUCC __0]]] ;
*>                 ~do _0_1 ]]
*;
*   dm [ ~do put (any plstarts) ]
*>     [ @ .p [pls-kanal] [@ _2 [pls-position]] OK ]
*;
*
*   dm [ open pls-kanal (any integer) ]
*>     [ @ _1_2 ; @ pls-position '(INTEGER 0) ]
*;
*   dm [ put (any plstarts) (any integer) * (any integer) ]
*>     [ _0_1 @ [pls-position] .p [[SUCC[pls-kanal]]MOD 4] _r2 ]
```

The evaluation of both texts results in a abstract-time-structure which could be edited either manually or automatically. In this latter case, it could be submitted to different processes of automatic transformation and interpretation, being the actual generation of sound only one of the multiple possible steps of such a chain.

The last part, "LC", begins with three eight-measure long statements of the alto-flute, which will be recorded and played by each speaker with an interval of 3, 5 and 1 quarters (the corresponding tempo is now quarter=90, the measure remains 3/4), making a canon that wanders all over the 4 speakers. Between each of these eight-measure statements, the alto flute plays three improvisations, whose durations are respectively of 9, 17 and 31 seconds. The third one modulates the frequency of the recorded signal of all three parts of the canon, and the result of this modulation will be anew recorded and again modulated from the alto flute. From this point on, the flute doesn't play any more, and the resulting modulation will be amplitude modulated -with the first element of "LB"- than it will be transposed dynamically. The transpositions will be gradually filtered with a *notch filter*, whose frequency is around the pitch f#5 and whose bandwidth will be dynamically narrowed up to this pitch. At the end there's only a filtered f#5 left, making an opposition to the first note of the piece, which was g3 (last and first note respectively of the chromatic row used as pitch-material).

"Gegensaetze (gegenseitig)" was the first piece of music using the AUDIACsystem in a real-time live performance. Up to its premiere, the system had only been used to steer another type of events (all within the electronic music production), but there were no pieces with live instruments composed specially for and with the system.

To bring this lecture to its end, I would like to clear up just one more point. The general conception of the work can be interpreted from several points of views, but my intention was to show the "Thesis-Antithesis" concept at the light of human, social and naturally also political relationships. I think that nowadays, a time in which Neonazi ideas and deeds wide out again (mostly in Europe and in the USA, but not only), the concept of a reciprocal action of opposite elements may be considered as the contrary to intolerance, racism and discrimination. That doesn't mean neither that my piece has got a secret program nor that it is a political work (which is mostly the case of Luigi Nono's music, to take only one example), but it may be able to recall these type of implications. The piece was first performed on June the 18th of 1994 in the city of Dortmund (Germany) by the german flautist Christianne Schulz.

Thank you very much.

June 1995
Javier Alejandro Garavaglia

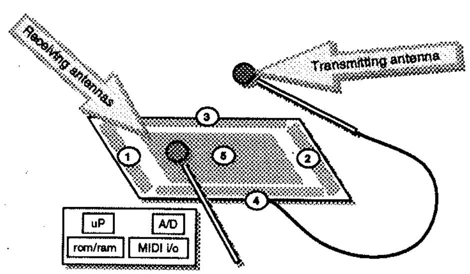
PadMaster: an improvisation environment for real time performance

Fernando Lopez-Lezcano
CCRMA (Center for Computer Research in Music and Acoustics), Stanford University
(nando@ccrma.stanford.edu)

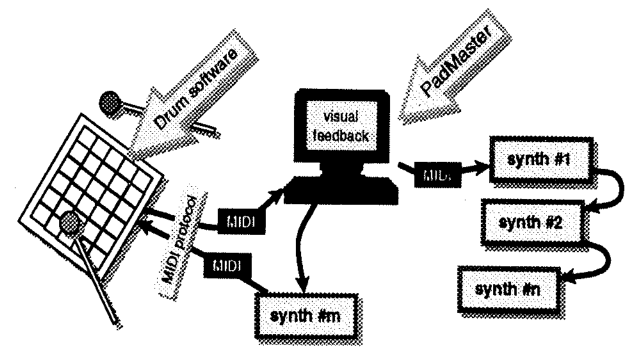
ABSTRACT: This paper will describe the design and implementation of PadMaster, a real-time improvisation environment running under the NextStep operating system. The system currently uses the Mathews/Boie Radio Drum as a three dimensional controller for interaction with the performer. PadMaster splits the surface of the drum into virtual programmable pads which can be grouped into scenes so that the behavior of the surface can be subtly or drastically altered during the performance.

1.0 The Radio Drum and the MIDI communication protocol

The current implementation of the Stanford Radio Drum was developed at CCRMA by Max Mathews as a simpler alternative to Boie's previous design. The two batons act as radio transmitting antennas. The signals are received by five antennas located underneath the surface of the drum. A multiplexed A/D converter translates the received signal strength from each antenna into numbers which are used by the on-board microprocessor to calculate the absolute position of each baton in space. The microprocessor uses this information to track the movement of the batons and to detect hits on the surface. Information about each hit includes both the x-y coordinates and the hit velocity. In addition to the two batons, the Radio Drum hardware includes two switches and four potentiometers. It has a MIDI interface that it can use to communicate with computers or synthesizers.



The behavior of the Drum is defined by the program stored in its EPROM. The drum software includes several functionally different programs that can be externally activated through MIDI System Exclusive messages. Through them, the Radio Drum can act as a stand alone conductor of a score or MIDI file, can improvise with several different options that map baton movement to MIDI commands or can act as a general purpose MIDI controller. This last program and the underlying protocol built on top of MIDI were originally designed by David Jaffe and Andy Schloss. This existing general purpose controller program was completely redesigned by the author. A more efficient and faster protocol was created that uses just one MIDI channel and is more bandwidth efficient in the use of MIDI resources. The protocol was also expanded to allow the controlling computer to upload / download calibration data from / to the Drum. Once the program is activated through a sysex message, the Radio Drum behaves as a three dimensional controller with six degrees of freedom.



Following is a short description of most of the control protocol:

- **System exclusive configuration messages**: can be used to turn

ON or OFF the communication program, set the MIDI channel used by the rest of the protocol, dump and load the internal calibration tables, set the trigger and release heights for both batons, request raw A/D measurements (useful for testing), etc.

- **Trigger / Release messages:** sent by the drum when a baton hits / leaves the surface. Each hit or release is represented by three MIDI controller messages, using continuous controllers 26 through 31. The messages are used to send the x and y positions and velocity of the hit or release.
- **Switches:** a switch message is sent by the drum when one of the two hardware switches changes state. The information is sent through controllers 5E to 5F.
- **Poll request:** sent by the computer to request the position in space of the batons at a given moment. The message uses a channel pressure MIDI message that encodes the required request as a pressure value. The controlling program can thus request the position of one or both batons and can also ask for the current value of the four potentiometers.
- **Poll answer:** sent by the drum in response to a poll request message. The requested information is sent through a string of channel pressure messages. As opposed to the Trigger / Release messages, the Poll Answer message contain no state information, which means that a state machine in the receiver program has to track the incoming messages. While this opens the possibility of garbled information due to lost MIDI bytes it was deemed more important to reduce the bandwidth used by the protocol as this is a frequently used message and the information gathered through it is refreshed periodically.

Planned enhancement to the protocol and underlying Radio Drum library routines include:

- **Detection of hits based on direction reversal.** The current implementation uses a height threshold based detection scheme which cannot reliably detect very fast rolls close to the surface of the drum.
- **Automatic position update.** To further decrease the MIDI bandwidth, the current polling scheme should be replaced with a timer based automatic transmission of the current position (that is, the Drum software should take care of sending periodic position messages). The new scheme will also include a sysex message to change the period of the transmission so as to enable the controlling software to throttle down the sampling rate of the position information when the MIDI stream becomes close to being saturated. This feature is currently implemented by changing the sampling rate of the position request polls.
- **Better internal linearization routines** for the three axes of control.

2.0 The PadMaster program

The PadMaster control code is written in Objective C, using the MusicKit as the foundation class hierarchy for MIDI event scheduling and control. The graphical interface was designed with NeXT's Interface Builder and the program runs on any workstation that supports the NextStep operating system (and has a MIDI driver available). PadMaster is connected through MIDI to the Radio Drum controller and to external synthesizers. The program uses the coordinates of the incoming Trigger and Release messages and an internal calibration map to split the surface of the drum into up to 30 virtual pads. Each pad is independently programmable to react in a specific way to the hit, and to the position information stream of one or more simultaneous axes of control. Pads can be grouped into Scenes, so that the behavior of the surface of the drum can be subtly or radically altered during the course of a performance. This is achieved by dynamically jumping to a different Scene, either through the use of a control pad programmed for that function or through another external controller. The screen of the computer continuously displays a representation of the virtual surface and gives visual feedback to the performer on the state of all the pads in the currently selected Scene.

The virtual pads can be split in two types depending on their function: **Performance** and **Control** pads.

2.1 Performance Pads

Performance Pads can be individually programmed to control the playback of MIDI sequences, note generating algorithms or soundfiles. The graphical representation of the pads on the screen gives instant visual feedback to the performer. Pads change color and status messages dynamically according to their state. A performance pad that is playing remains active even if the performer selects a different Scene, so that chains of events can be started from one Scene and will continue to run even though the performer latter switches to a

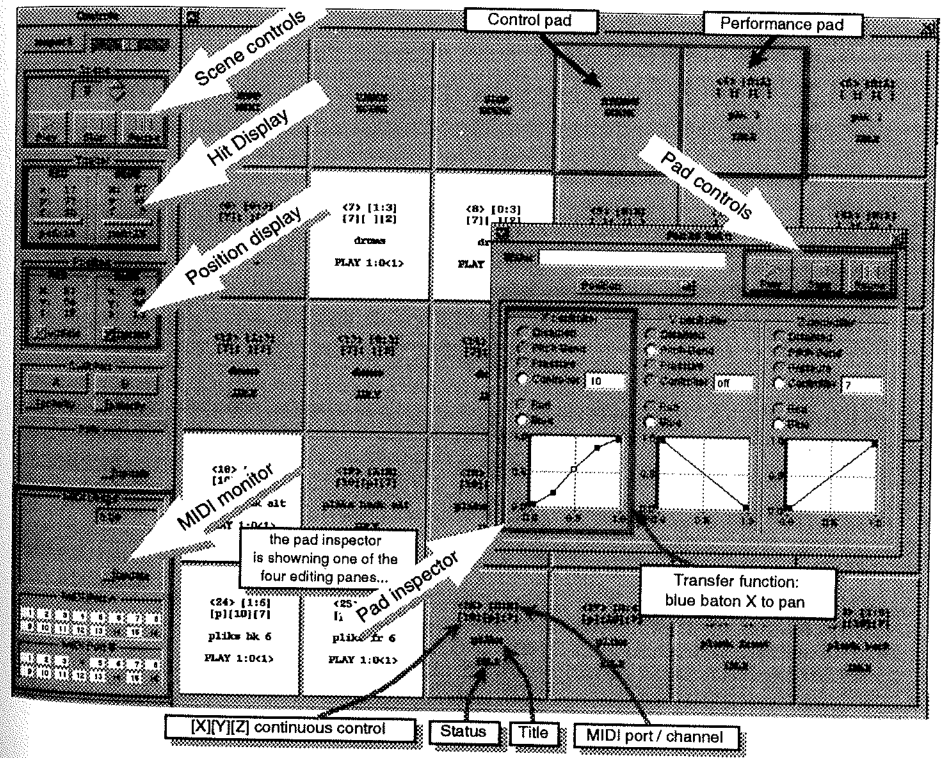
different Scene. The status is updated for all active pads but only those in the currently selected Scene show up on the graphical representation of the drum surface.

2.2 Control Pads

Control Pads are used to trigger actions that globally affect the performance of a Scene. A pad can be programmed to change the current Scene when hit, jumping to the next or previous Scene, thus redefining the behavior of the whole surface of the drum. Control pads can also be used to pause, resume or stop all playing pads in the currently selected Scene.

3. Inside a pad

Editable parameters inside each pad can be changed through a standard NextStep inspector window with several editing panes. The first pane can be used to select the type of pad and, in the case of performance pads, the triggering baton and the action that is executed when the pad is hit. The possible actions include starting / pausing / resuming a sequence, starting a new overlapping sequence, or playing the next note of a list of notes. It also selects the MIDI port, channel and program number that will be used for MIDI transmission and allows editing of a graphical mapping of hit velocity to note velocity for the selected sequence. The second pane edits the tempo options for the pad. Tempo can be global, per pad or per sequence inside a pad (as there can be more than one instance of a sequence playing at the same time). There is a tempo envelope and it is also possible to control tempo with the hit velocity or with any of the six available axes of continuous control. The third pane lets you associate up to three continuous MIDI message streams (pitch bend, channel pressure or any con-



troller) with the position of up to three of the six axes of control. All these function mappings are created through graphical function editors. The fourth pane edits the sequence of notes that are played when the pad is hit. The sequence is expressed as a normal MusicKit text scorefile. The scorefile format has been enhanced with additional tags that represent all programmable parameters in a pad. It is then possible to externally generate a textual representation of a pad and then load it into PadMaster (for example, a set of pads for a performance might be algorithmically generated and then loaded into the program).

4. PadMaster in performance

PadMaster has been used to compose and perform "Espresso Machine", a piece for PadMaster and Radio Drum, two TG77's and processed electronic cello (Chris Chafe playing his celletto). The piece is an environment for improvisation in which the PadMaster and celletto performers exchange ideas and play with predetermined materials. The piece is composed in three PadMaster Scenes, each with several groups of related materials that are triggered during the performance. One baton is reserved for triggering pads and the other for continuous three dimensional control of the currently performing pads.

The performance of this piece on several occasions has raised several issues. The simultaneous mapping for several pads of baton movement to MIDI continuous controllers is one of the most interesting performance capabilities of the program, but also raises the possibility of serious MIDI bandwidth clogging. The current version of PadMaster dynamically adapts to the changing conditions and adjusts the position sampling frequency to try to reduce the bandwidth used when several pads are playing simultaneously. More work needs to be done in measuring MIDI usage in a more precise way to avoid sending too much information, but at the same time to avoid control lag if the sampling frequency fall to a very low value. There is also a measurable gap in playback when scenes are changed, during which MIDI activity is not updated as the MIDI and graphical routines share the same execution thread.

5. Future developments

PadMaster is currently undergoing a complete rewrite to implement new and improved functionality.

Pads will be resizable so that each scene can have different number and size of pads if necessary. We have found that sometimes it would be desirable to concentrate important or critical performance functions in a few big pads. Resizable pads would also allow for linking performance behavior to the place where the pad is hit, something that is not possible in the current version. Another enhancement to pads will be inheritance, so that multiple pads with related behavior (something we have found common and very useful) could be grouped together, with the common functionality being editable as a group.

The action types of performance pads will be enhanced to allow for inclusion of algorithms and soundfile playback. The algorithms will be able to use a well defined API to enable linking of baton movement to algorithm parameters.

The NextStep operating system includes a remarkably easy to use system to communicate with remote objects (objects that live in other computer[s] but are directly linked to the execution of a local program). That opens the possibility of using remote computers connected through Ethernet as servers for MIDI or soundfile playback. There are also several scheduled refinements for performance such as a completely separate thread for all MIDI interaction so that graphics may lag behind the performance but there will be no delays when switching between scenes.

Another important enhancement will be defining a way to use different MIDI controllers in addition or instead of the Radio Drum (percussion controllers, normal keyboards, MIDI pedals, etc).

References:

[1] Max Mathews, *The Stanford Radio Drum, 1990*

[2] David Jaffe, Julius Smith and others, *The MusicKit* *"http://ccrma-www/CCRMA/Software/MusicKit/MusicKit.html"*

[3] Carlos Cerana (composer) / Adrian Rodriguez (programmer), *MiniMax, a piece for Radio Drum*

INTERACTIVE COMPOSITION USING MARKOV CHAIN AND BOUNDARY FUNCTIONS

JÔNATAS MANZOLLI * & ADOLFO MAIA JR. **

*Interdisciplinary Nucleus for Sound Studies (NICS)**

*Applied Mathematics Department (IMEEC)***

University of Campinas (UNICAMP)

13081-970 Campinas SP- BRAZIL

Jonatas@dsif.fee.unicamp.br

ABSTRACT

The research presented here reviews the use of stochastic processes on composition; Markov chains were studied as a tool for developing musical structures in the 70s. This paper presents a model based on Probability Vectors and Boundary Functions which are used on iterative processes to generate further sequences of vectors. The resultant numerical structures are mapped to different classes of musical parameters. The text presents this new approach - the musical ideas behind the method as well as the mathematical model. Finally, it describes a graphic-based compositional system for a MS-Windows Environment.

1. INTRODUCTION

One of the composer's main tasks is to make choice and it would be possible to find several moments within a compositional process in which his/her choices are based on chance. Adding to this the Boulez's concept (1986) that there is inside of each composer a *kernel of darkness* and the Cage's concept that *pure chance is ultimately irrational, by which I mean there can be no closed system of explanation that includes it, save for the Universe itself* (Cage, 1961), we have joined ideas which give us a good flavour and an insight for presenting an application of stochastic process to musical composition..

There are several historical references of composers using random methods in their compositions. Even musicians such as Mozart, Haydn and C.P.E. Bach worked with compositional techniques based on chance. As described by Loy (1988), this method was called Mozart's Dice Game (Würfelspiel); it was a music game used for composing minuets and other incidental works. If the strict musical aesthetic of the Classical Era provided a framework for such approach, it would not perhaps be so surprising to find Cage in the twentieth century using chance as a compositional tool. An interesting connection between the world of computer experimental music and the world of chances was established in the first Computer Music piece by Cage-Hiller called HPSCHD (1967-69) as described by Austin (1992). The genesis the compositional model used in HPSCHD was the work of Hiller & Isaacson (1959); their pioneering research was the seed for an model called *rules-driven-by-noise*. The development of this concept subsequently grew in many different directions and many different decision-techniques were related to stochastic process and Markov chains, as reported by Hiller & Isaacson (1959), Xenakis (1971) and Jones (1981).

The first attempt to integrate a graphic interface and Markov Chains in a computer compositional system was the Jam Factory program. It used four generative structures named *players* which were controlled by *Transition Tables*. This system was designed by Zicarelli (1987) and as he said: *an essential part of the Jam Factory algorithm is the probabilistic decision made on every note as to what transition table to use.* The aim of this program was to use a n-th order transition table to affect the degree of variations of an original material or seed material.

From this point, it is possible to present our research plan. We use a Markov process to generate sequences of *Probability Vectors* (or transition tables as described by Zicarelli, 1987) and these vectors, through numerical iterations, determine the trajectories of the musical material. We also add to this model a set of *Boundary Functions* in a way that the iterative process produces a sequence of probability vectors which converges autonomously to stable states. Finally, the computer graphic interface is a tool for controlling the initial parameters, to listen to the convergence process and to choose the best musical result.

The text below is the first report of our research and as such we still have a lot of questions and proposals. Our computer implementation is very simple and we intend to gradually move on the direction of complex compositional systems. We presented a brief review of the mathematical ideas related to stochastic processes. After we introduce a definition of the Boundary Functions and it follows a discussion of the computer implementation in a MS-Windows Environment.

2. MARKOV CHAIN AS A COMPOSITIONAL MACHINE

In this section we introduce the mathematical concepts underlying our approach. A definition of a Stochastic Process implies in two basic concepts: *State and Probability Transition Between States*. It is also a useful mathematical tool to describe these states with parameters so named as *State Variable* and the overall set of states is entitled *State Space*. In music, for example, we have an immense freedom of choosing this state space: *the 12 pitches of the chromatic scale, a set of sounds generated by a computer sound card, a set of sound from nature, a set of instruments and so on.* These state variables determine completely each particular state.

Now, suppose that we can go from a state to another one randomly. In this case we can ask this question: *which is the parameter to do such a transition between states?* This is named as *Probability Between States*. In this way, to run a stochastic process we must assign a transition probability between all elements of the state space. We define this as a *Distribution Probability* for all possible transitions. Any sequence of states can be constructed applying the distribution probability to control the transition process. These sequences are named *Trajectories*. For example, if a musician uses a Gregorian Mode as state space the resultant trajectory is a modal melody or if he/she defines the state space as a chromatic scale the trajectory will be a chromatic sequence.

Mathematically, we can write the following definition for the *State Space* with m elements:

$$\mathbf{S} = \{s_1, s_2, s_3 \ldots s_m\} \Leftrightarrow \mathbf{S} = \{s_i\}_{i=1\ldots m}$$

and a sequence such as

$$\{s_1^{\,1}, s_1^{\,2}, s_2^{\,3} \ldots s_i^{\,N}\} \Leftrightarrow \{s_i^{\,j}\} \text{ with } 1 \le i \le m,\ j=1\ldots N \text{ and } s_i^{\,j} \in \mathbf{S}$$

is a *Trajectory*; where repetition of a state is, of course, permitted and N is an arbitrary number which gives the length of the trajectory.

A very useful stochastic process is the so called *Markov Process*. This is a particular case in which the local probability transition depends only on the previous state. In mathematical terms:

$$P(S_{ij}) = (S_i \rightarrow S_j).\ (S_j)$$

Going down to the earth, we specialise our model to elucidate this idea with a simple musical example. Let us take a particular case whose space states are three pitches. We can denote them as $\{\mathbf{A_1, A_2, A_3}\}$. It is possible to define the probability transitions as

$$P_{ij} = P(A_i \rightarrow A_j) \text{ with } i,j = 1,2,3$$

and then we can write a set of all these probabilities as a *3x3 Matrix* $P = [p_{ij}]_{\text{with } i,j = 1\ldots3}$. The fundamental property of this transition probability matrix is that the sum each row must be equal to one and using this matrix we can write a multiple trajectory for all elements of the state space.

Joining the ideas above the *Markov Chain* with n-steps can be written as

$$v^{N+1} = P.v^{N} \quad i=0,1,2\ldots N$$

where V^0 is a given initial probability vector (distribution or transition table). In other words the Markov chain is nothing more than a sequence of probability vectors $\{\mathbf{v}^0, \mathbf{v}^1, \mathbf{v}^2 \ldots \mathbf{v}^N\}$ generated by an iterative process of multiplication of the initial vector $\mathbf{V}^0$ by the matrix $\mathbf{P}$.

To make contact with musical reality, we must construct a rule that transforms a numerical trajectory in a sequence of pitches. It is important to stress that we are using a set of three pitches as an example, but the sequence can be constructed using a set of dynamics, rhythmic figures, sounds, instruments, numerical samples taken from a waveform or any other set of musical material. Since the state space is finite (there are only 03 elements in this example) we can arrange it like a line vector which we name *Vector State*: $\mathbf{A} = [A_1\ A_2\ A_3]$.

In this way all trajectories of elements can be collected in the sequence of state vector:

$$\mathbf{A}^0 = [A_1{}^0\, A_2{}^0\, A_3{}^0] \Rightarrow \mathbf{A}^1 = [A_1{}^1\, A_2{}^1\, A_3{}^1] \Rightarrow \ldots.\ \mathbf{A}^N = [A_1{}^N\, A_2{}^N\, A_3{}^N]$$

This is achieved with some functions we have named *Boundary Functions* (BF). To construct them, let us define a fixed vector

$$\mathbf{K} = [k_1\ k_2\ k_3] \text{ satisfying } 0 \le k_i \le 1.$$

The boundary functions (BF) are defined as follows:

A. *Cyclical Boundary Function*

Take a probability vector $[v_1^N, v_2^N, v_3^N]$ of the N-th step of a Markov Process and compare its entries with the fixed vector $\mathbf{K}$. Now we impose the boundary condition:

$$\begin{aligned} \text{if } \ v_i^N \le k_i \ &\Rightarrow A_i^{N+1} = A_i^N \\ \textit{otherwise } &\Rightarrow A_i^{N+1} = A_{i+1}^N \end{aligned}$$

In this case, the resultant musical pattern is a single melodic line in which the pitches follow a cyclical order within the three-pitches state space. Of course this is nothing more than one of the infinite possibilities of BF for Markov chains applied to music. Each one of them will generate a different musical behaviour. Below we give two other BFs.

B.Rhythmic Boundary Function

We can repeat the above Markov Process independently, p times say. Putting these p lines in parallel we have a kind of *multi-phonic structure* and extending the three-pitches space state to include *silence*, *e.g.* $[A_1, A_2, A_3, \mathit{silence}]$. We can write the following definition:

$$\text{if } v_i^N \le k_i \;\Rightarrow\; A_i^{N+1} = A_i^N$$
$$\textit{otherwise} \;\Rightarrow\; \textit{silence}$$

This BF, by its own definition, generates a multi-layered sequence of pitches with an implicit rhythmic structure.

C. Parallel Boundary Functions

We can run an arbitrary number of independent Markov process simultaneously, in a way that each one controls a different musical parameter. For example we can have 03 stochastic processes to control pitches, dynamics and rhythmic figures. In this way the overall result is a complex sound pattern which can be understood as a kind of composition. Finally, we would like to stress, once more, that there is an infinite number of these B.F. Their definitions depends only of musical taste and the composer's imagination.

3. COMPOSING USING A GRAPHIC INTERFACE

Our intention is to provide an environment for sound exploration giving to a musician the essential mutability to control his/her creative decisions. As presented above, we use a graphic interface to control the probabilistic decision such as the Jam Factory program (Zicarelli 1987), but we change the probability vectors itself at each step of the process and we imposed Boundary Functions to drive the numerical sequences to stable states or attractors. Thus the compositional idea behind our model is to explore a system in which the musical material is self-organised through an iterative process. Instead of using probability vectors or n-th order tables as the decision criteria, the vector's' numerical behaviour is the generative process.

We designed a graphic interface to control the Markov process and the musical mapping. We decided to implement the *Parallel Boundary Functions Model* presented above with 03 stochastic processes to control pitches represented by MIDI Note Number table, dynamics represented by MIDI Velocity Number table and rhythmic figures input from a MIDI Keyboard and represented by floating points. The system was developed as a three part real time process. The first part we called *Mathematical Section* and it is the controller of the Markov process i.e. the stochastic Matrix and the probability vector. The stochastic matrix has *03 Diagonal Blocks* such that each block controls independently, or parallely, the three musical parameters above cited. Correspondingly, the probability vector is also partitioned in 03 parts on which the 03 blocks acts. This structure of the Mathematical Section is elucidated below in *figure 01*.

The *Musical Section* is a set of numbers with contains the musical State Space represented by MIDI parameters and rhythmic figures i.e. [1..127] for note, [1..127] for velocity, floating point for rhythm. There is also rhythmic pulse, input by the musician, to control the speed of the performance process. These musical parameters are stored in a numerical vector which is also input by the composer using a keyboard or any other MIDI Controller.

The third section is the *Visual Interface* which was developed for a MS-Windows environment. To design this interface, we used the idea of sonic process controlled by *parametric interactive cells*. A cell is defined as a set of controller parameters which is modified in real time. The program runs 12 Markov processes in parallel each of them is a parametric cell. A cell contains a stochastic Matrix, a probability vector and a representation of the musical state space to control the compositional process. This interactive process consists of two steps: a) *the computer initialises a cell using random numbers* and b) *these parameters are changed by the composer using a MIDI controller device*.

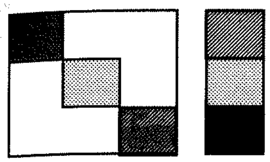


figure 1.0

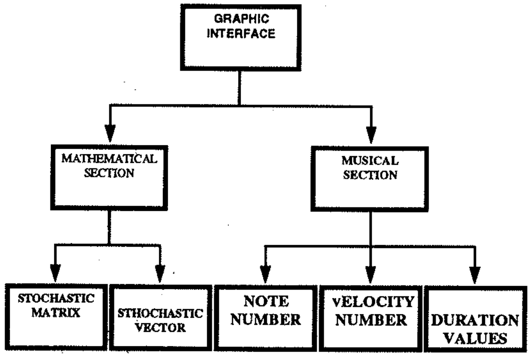
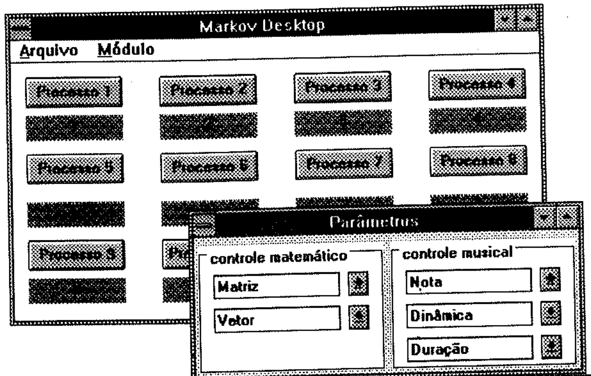


figure 2.0

Figure 1.0 the structure of the Probability Matrix and Probability Vector. Figure 2.0 is a diagram of the compositional system.

The computer controls the compositional process in real-time. The iteration of the Markov process runs at same time the composer listens to the musical material generated by the system. Thus each computer's iteration encapsulates multiplication of the stochastic matrices by the probability vectors and the mapping of the probabilities numbers to the musical State Space. The process is a tool for generating a convergence process and changing, controlling and listening to the musical result. In the *figure 02*, we present a diagram of the compositional system and the graphic interface is presented in *figure 03*.



This is the graphic interface of the for a MS-Windows environment. The top window controls the parametric cells and the right window is used to input control parameters.

4. CONCLUSION

In this work we have emphasised the dynamic aspect of the stochastic processes (in particular of the Markov processes) in contrast to the static concept of probability distribution as worked by several authors such as Xenakis (1970), Jones (1981), etc. We have investigated the applications of our method to a macro-structural approach based on the MIDI event representation. From this point in our research, we are going to construct a micro-structural model in which the Markov process will control waveforms or sound cells such as described by Manzolli (1994). Finally, the use of Stochastic Process in composition can provide many ways of helping the composer's choices. If under Boulez (1986) point of view there is a *kernel of darkness* inside each composer and for Cage (1961) *pure chance is ultimately irrational*, we are still searching for a good representation of this complex and immense world that is the compositional systems created by the composer's own imagination.

REFERENCES

Austin, L. 1992. "An Interview with John Cage and Lejaren Hiller". Computer Music Journal 16(3):15-29.

Boulez, P. 1986. Orientations. Translated by M. Cooper from "Points de Rèpere", 1981. London: Butler & Tanner Ltd, pg 60-67.

Cage, J. 1961. Silence, The MIT Press, Cambridge, Massachusetts.

Hiller, L. & L.Isaacson 1959. Experimental Music, New York:McGraw-Hill Book Company Inc.

Jones, K. 1981. "Composition Applications of Stochastic Processes." Computer Music Journal 5(2):45-61.

Loy, G. 1988. "Composing with Computers a Survey of Some Compositional Formalisms and Music Programming Languages." In Current Directions in Computer Music Research, ed. Mathews, M.V. and J.R. Pierce, Cambridge: The MIT Press, Massachussetts, 1989.

Manzolli, J. 1994. "Non-linear Dynamics as Timbral Constructs." Proceedings of the I Brazilian Symposium on Computer Music, Caxambú,MG.

Xenakis, I. 1971. Formalized Music, Indiana University Press, Bloomington.

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

ACKNOWLEDGEMENTS

We would like to thank the Brazilian Research Counsel (CNPq) for financial support, Microsoft Inc. for providing the basic software used for developing the graphic interface, to the head of the NICS, Prof. Raul do Valle, for allowing us to develop this new approach under the NICS umbrella and our student Marciel Rocha who has been pleased into debugging the compositional program.

THE DEVELOPMENT OF A GESTURE INTERFACES' LABORATORY

JÔNATAS MANZOLLI
Interdisciplinary Nucleus for Sound Studies (NICS)
University of Campinas (UNICAMP)
13081-970 Campinas SP- BRAZIL
Jonatas@dsif.fee.unicamp.br

ABSTRACT

This describes the goals of the Laboratório de Interfaces Gestuais (LIGA) at the Interdisciplinary Nucleus for Sound Studies (NICS-UNICAMP). The aim of LIGA is to build MIDI controllers using digital circuits and various transducers, and to create graphic interfaces for interactive compositional purposes. Three prototypes are presented: two new graphic interfaces developed for MS-Windows Environment and a glove interface which uses the movement of the hands and wrists to produce MIDI events.

1. INTRODUCTION

The desire to use gestures to produce music with electronic devices dates from 1919 when an electronic instrument *Theremin* was invented by the Russian scientist Lev Termen. Music was produced when a performer moved his hands in the neighborhood of antennae to control the amplitude and the pitch of two quasi-sinusoidal oscillators (Glinsky, 1992; Boston, 1989).

More recently, the use of computers as musical instruments was referred to by Boulez (1977) in the following terms: *Oscillators, amplifiers, and computers were not invented in order to create music; however, and particularly in the case of the computer, their functions are so easily generalized, so eminently transformable, that there has been a wish to devise different objectives from the direct one: accidental conjunction will create a mutation.*

The development of the MIDI protocol allowed to use the computer as a controller of musical parameters enabling interactions with musician(s) in real time. However, the MIDI digital data stream differs fundamentally from the analogue approach. For example, using a Theremin a musician acts directly in the sound continuum. In opposition, MIDI devices control only a discrete sequence of sound events called MIDI events or messages. Therefore the development of MIDI-based interfaces to operate closer to the sound level is an important Computer Music research issue. The Steim Foundation in Amsterdam, has conducted significant research in this direction and has developed a series of interfaces (Krefeld, 1990). Composers have designed new instruments (Teiltelbaum, 1984; Beck, 1991). A novel approach to musical organization was presented by Orton (1992): a MIDI-based interactive instrument called *MIDIGRID*. Rowe (1993) discusses and illustrates very well the concepts of interaction between musicians and computer, contains a survey on interactive systems and a description of his own *Cypher*.

The seed for the *Laboratório de Interfaces Gestuais* (LIGA) was planted at the Institute of Sonology (1991-92) when I worked in a join project with the Steim Foundation (Manzolli, 1993) developing interactive gloves with Hall effect sensors and mercury switches for the Sensorlab (Cost 1992). I came back to Brazil with those gloves but without the Sensorlab. This new situation left me to create LIGA. A search for solutions to replace the SensorLab and to use other transducers begun. Another goal was to create graphic environment interfaces.

The next sections present a brief description of the research's going on LIGA at this moment. They describe the LIGA's activities elucidating theoretical issues and presenting a brief description of three new interfaces: *LIM, Quadrilátero and Luvas de Pelica.*

2. RESEARCH GOALS

Interaction between musicians is a tool for development of musical ideas in many musical and cultural contexts. In the harmonic cadences of a Jazz section, in the rhythmic patterns of the Indian Tabla, or in the massive rhythmic structure produced by a Brazilian school of samba, the individual ideas are transformed in collective ideas, the sonic context and the musical skills are adjusted to the communal sense of the musical realization and music becomes a synergetic interaction of the participants.

LIGA's main goal is to investigate the possibilities of creating mechanisms to control musical transformations in performance situations. The basic proposals are: a) *to use a performer not only as an interpreter but also as a source for musical development and* b) *to use the computer as musical instrument.* LIGA also intends to humanize the relation between machine and musician, creating means for performers to influence the musical results. These concepts imply seeing the compositional act as an exploration in which the composer, the computer and the performer are entities within an integrated and unified real-time system.

In the first period (1994-95) LIGA efforts were concentrated on software design. Some hardware was developed that produces interesting sonic results using few electronic components. LIGA also forms human resources, as Walter Ohtsuki, working on software design and Renato Meirelles, taking care of the hardware development.

3. GRAPHIC INTERFACES

This section discusses the musical and algorithmic concepts incorporated in two graphic systems called *Laboratório Interativo MIDI and Quadrilátero.* They can be defined as Graphic Instrument Environments wherein the mouse location on the computer screen is used to control musical events during performance. They use these gestures as a real-time generator of MIDI events.

A. *Laboratório Interativo* MIDI *(LIM)*

LIM explores the MIDI protocol in an innovative way by not sequencing MIDI events linearly thus giving freedom for combining MIDI data in many ways. The mouse movements upon iconic operators or objects (faders and a graphic pad) are stored in an array or track. This information is further treated in various ways to form MIDI messages. Added to LIM are two other devices: *a time function and a modulation procedure.* The time function changes speed of individual tracks making possible to stretch the duration of each recorded event. The modulation uses MIDI input and MIDI feedback to produces complex transformations in the tracks as well. Two modulation coefficients interfere in a track by combining its events with the incoming MIDI data as follows:

$$\text{new-event} = \alpha.\text{MIDI} + (1 - \alpha).\text{old-event} \quad \textit{with } 0 \leq \alpha \leq 1$$

It is surprisingly how large is the number of combinations that LIM is able to generate. The sonic results being complex, this interface produces interesting soundscapes. LIM explores quite all possible MIDI messages using simple graphic devices: a set of six scroll buttons for Program Change, Channel, Note, Velocity and Pitch Bend, a graphic pad, a graphic mixing desk to control the volume of 16 MIDI channels, a small keyboard and two array records each one with three tracks .

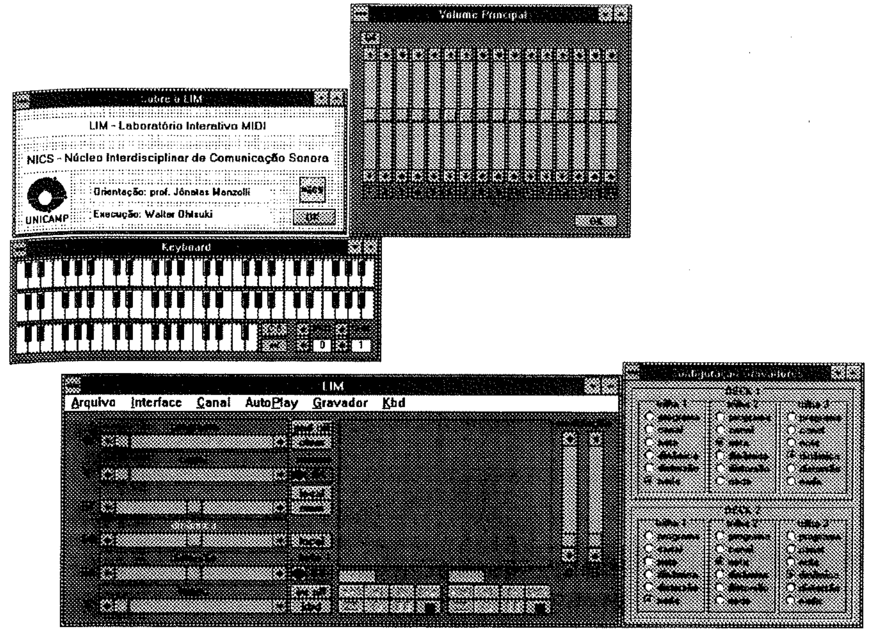


Figure 1.0 - LIM's windows: the presentation form, the mixing desk, the controller keyboard, the control panel and the array record configuration table.

B. Quadrilátero

The genesis of *Quadrilátero* was inspired by the art of painting. This interface differs from LIM by the way the system interprets the mouse's gestures. While LIM uses the mouse to generate MIDI events in several ways, *Quadrilátero* uses the mouse as a graphic brushes transforming the computer screen in a sound canvas.

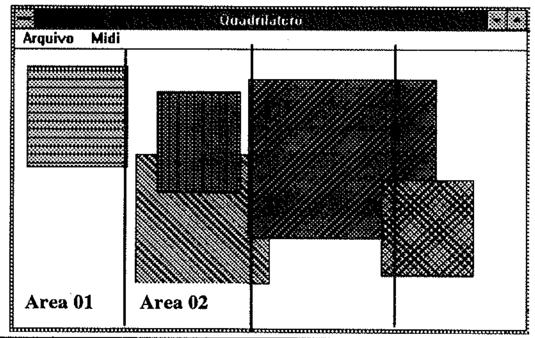


Figure 2.0 - Quadilátero's graphic interface showing the sonic frames divided by vertical lines in four areas of interaction.

The algorithmic approach was to simulate a situation wherein the computer controls a sound environment and the performer produces changes in the soundscape by means of manipulating rhythmic structures, clusters of notes and pitch sequences. The performance and composition processes are integrated in *Quadrilátero*. The musician draws rectangle areas with the mouse and these became sonic frames. Then fills these frames with MIDI data (this can be done by the mouse or by a MIDI controller). After this the sonic canvas is explored with the mouse - its paths upon frames make their contents to be sent to the MIDI port.

A graphic function looks after frame's overlaps. When the performer moves the mouse along the screen it takes care of overlapping data. Take *Figure 2.0* as an example: if the mouse is in *Area 01* the computer plays only one set of events, but if the mouse is in an overlapping sonic frame in *Area 02*, the computer plays two sets of events simultaneously. Using Quadrilátero a musician constructs a space for sonic exploration and can use various rectangle combinations to create different sonic frames overlaps.

4. GLOVE INTERFACE

The expressive potential of the human body, was the starting point for this interface. The project was to draw an explicit line between music and body expression. Music resulting from movement, and dancing to music is seen as the same thing in many different musical cultures; that is, the regular rhythmic movements involved in producing music are regarded as a form of dance. The first device needed here is a sensor/transducer to measure the density of performance actions related to some scale. The information from the sensor produces a control for the gesture output; the system's triggers are a set of these sensors fixed in a pair of gloves. The second device used on *Luvas de Pelica* is a set of mercury switches which create rhythmic patterns produced by the vibrations and movements of the performer's wrists.

The aim of this research was to search for a simple glove interface which could be constructed with few hardware resources. The goal was to develop a very simple controller black box to receive information from non-expensive transducers. In this way, the ideas used on the hardware project were: a) the controller black box is a combination of the Alesis D4 drum module and a very single digital multiplex circuit and b) the transducers used are piezo electric devices and mercury switches.

Luvas de Pelica was developed to produce music by contrasts such as turbulence/calm, loud/soft sounds, high/low densities and silence/sound. The performance is a hand-dance in which gestures interplay sound events and the music happens as a consequence of the action of these movements. These produce pressure changes on the piezo electric transducers and vibrations in the mercury switches. The diagram of the interface's hardware is presented on the next page.

CONCLUSION

Performance with new interfaces, is a musical situation which should be further explored and studied. The point here is not so much the presentation of a new technology in the stage, but rather the understanding of relations between algorithmic real-time generated music and live musicians. The effective integration of computers and performers to provide an appropriate control of real-time sonic events is a goal still to be fully achieved.

The handling of interactive compositional paradigms implies the combining of two opposite concepts in one system: determinacy and indeterminacy. Further the objectivity of a formal approach enables the expression of subjectivity in the transforming of musical materials. The use of improvisation as a way to produce good music, emphasizes an aspect of musical practice which may not have a priority in the West. It would certainly be interesting to see the use of computers developing new ways of human interaction through music.

AbCMus: Uma abordagem para construção musical interativa

CARLOS GUSTAVO M. GUERRA
Programa de Pós-Graduação em Educação
Universidade Federal de Santa Catarina
Caixa Postal 5044 — CEP 88040-970
Florianópolis — SC — Brasil
e-mail: cgguerra@ced.ufsc.br (ou cec1cgg@ibm.ufsc.br)

Abstract

Traditional music teaching gives more importance to theoretical aspects and instrument performance than to creation or research. In the way of allowing also to **non-specialist** the experience of ***doing music***, we propose and present **AbCMus** — acronym from Portuguese to **Musical Construction Approach** — a trans-disciplinary approach involving (also but not just) philosophy, psychopedagogy and computer music, where musical knowledge comes from the **interaction** between the non-specialist and his/her musical experimentations — and not just as mental rules —, allowing creativity, curiosity, dance, feeling, self-consciousness, game, research. AbCMus also propose the interaction of alternative musical notations, as support for another musical conceptions than the European Erudite. Computational aspects are also presented, as a set of operations to interactive musical knowledge construction and a drag-and-drop user interface proposal.

No ensino tradicional de música há uma ênfase muito maior em aspectos teóricos e técnicos, pouco se oferecendo para o **não-especialista** a experiência do *fazer música*. Como veremos a seguir, este tipo de abordagem envolve concepções de arte, cultura, educação e sociedade fortemente interligadas e inter-relacionadas. O objetivo deste artigo é apresentar a ***Abordagem de Construção Musical***, uma abordagem transdisciplinar alternativa para a pesquisa e o aprendizado musical. Na medida que esta abordagem é uma tentativa de resposta a um problema, comecemos apresentando o problema e, a seguir, o processo de construção desta alternativa

O problema: o ensino tradicional de música

Entende-se aqui como *ensino tradicional de música* aquele majoritariamente realizado nas escolas de 1° e 2° graus, nas faculdades (institutos, universidades) de música e nos conservatórios musicais; e, em especial, nos países de colonização européia. Claro que este ensino não é realizado exatamente da mesma forma em todos estes ambientes e em todos os lugares; mas isto não impede uma análise geral das características mais comuns, talvez mais adequadas a uns ambientes que a outros.

O conhecimento humano envolve as mais diferentes áreas, as mais diferentes culturas, as mais diferentes abordadagens, filosofias, crenças, ideologias. Os currículos escolares são resultado de um processo de **seleção** destes conteúdos. Na escola, o conhecimento científico é priorizado sobre o artístico, o filosófico, o místico. E a cultura erudita européia é tomada como referencial de anállise das demais culturas. Assim, dentre as várias concepções de música e dentre os diversos estilos musicais, é a música erudita européia (principalmente dos séculos XVI ao XIX) que é usada como referência do que deve ser ensinado. A própria separação entre *música popular* e *música erudita* reflete o preconceito para com outros povos, outras classes sociais, outras épocas, outros ambientes culturais.

O ensino de música se faz de forma independente do ensino de História e Filosofia (da Arte, da Cultura), de Física (acústica), de Eletrônica (construção de equipamentos), de Matemática (séries de Fourier, proporções), de Técnicas Industriais (confecção de instrumentos), de Comunicação e Expressão (notações musicais, criação), de Educação Física (dança, ritmo corporal), de Ciências Sociais (antropologia da música, dos rituais) — relações que são fundamentais para uma compreensão mais global da música.

A relação entre fragmentação do conhecimento, especialização do trabalho, música erudita européia e arte como produto é explorada por Guerra (1995): o palco separa o especialista (que produz) do leigo (que consome). Da *platéia*, espera-se que fique passivamente sentada, sem dançar, sem se exprimir — sua interferência fica limitada a comprar o ingresso, sentar e bater palmas no final das obras. Uma pessoa compõe (um *gênio*), outro rege (o chefe), outros tocam; se há dança, é planejada por um especialista (o coreógrafo) e dançada por outros especialistas (os bailarinos), dos quais não se espera criatividade, mas eficiência; os *artistas* (técnicos) são amparados por um grande número de outros trabalhadores especializados (produção, segurança, iluminação, bilheteria, motoristas, contra-regras, faxineiros, etc.).

Na medida em que há esta separação entre o processo de criação e a execução musicais, também passa a haver uma diferenciação na formação dos dois tipos de especialistas. Mas como a aceitação de um autor e sua obra como *erudito* é um processo lento e um tanto desvinculado dos interesses atuais, temos como resultado um ensino de música muito mais voltado à formação de *executores* musicais do que de compositores.

Assim, no que chamamos *ensino tradicional*, os conceitos musicais são transmitidos através do aprendizado da técnica de um determinado instrumento. A técnica de manipulação do instrumento passa a ser mais importante que a produção e a composição musicais. O próprio espaço para se aprender música muitas vezes se chama ***conservatório*** — sugerindo um espaço para o aprendizado de **reprodução** musical — em oposição a um espaço de criação, que poderia ser chamado ***criatório***.

Este vínculo entre música erudita européia, aprendizado musical baseado na reprodução e músico como técnico na execução de instrumentos musicais fica ainda mais claro quando lembramos que uma das características básicas do ensino tradicional de música é a utilização da partitura (ou pentagrama) como sistema notacional básico:

> O ensino de música passa a ser baseado no aprendizado da partitura, um sistema de notação musical voltado para a música erudita européia dos séculos XVI em diante. Além de tomar como música uma visão particular de música, o ensino baseado em partitura assume que o estudante é principalmente um técnico, um executor, uma vez que esta é voltada mais para a **representação** musical do que para a **pesquisa** e o **aprendizado**. E ainda mais, passa-se a privilegiar o piano, o violino — instrumentos bastante associados a este tipo particular de música; e a técnica de execução passa a ser critério de decisão do *saber* e do *não-saber* música.
>
> (Guerra, 1995, p.59)

O desenvolvimento de outra concepção de ensino musical passa, necessariamente, pela exploração de notações musicais alternativas, conforme apresentado por Guerra (1994b, p.271): "As notações, como as linguagens, não são apenas formas de representação do conhecimento, de registro final do processo de pensar. As linguagens são o próprio suporte para o pensamento, tendo íntima relação com as estruturas mentais. (...) Desenvolver novas notações musicais representa possibilitar novas formas de se perceber e entender a música, assim como de pesquisar e criar." Como veremos a seguir, não se trata "de substituir a partitura por uma nova notação, *mais atual* ou *mais completa*; mas sim, de se conceber a música como passível de representação sob **diversas notações simultaneamente**" (idem, p.272), refletindo as várias concepções, abordagens, usos, relações, etc., possíveis para a música.

Outra característica da visão de mundo erudita européia, transferida para a educação, a música, etc. é priorizar as atividades mentais — essencialmente linguagem, raciocínio, memória e inteligência — sobre as demais. Assim, no ensino de Artes, por exemplo, decorar nomes de autores e obras toma o tempo de se fazer poesia ou pintar; o desenho geométrico substitui o abstrato; ao invés de escultura, artesanato; o teatro enquanto relação e trabalho de corpo, é substituído pelo jogral; aulas de música são ocupadas muito mais com *a vida dos grandes mestres*, com a memorização de hinos nacionais, do que com vivência de ritmos e execução de instrumentos.

Como vemos, não podemos situar o problema do ensino tradicional de música apenas no contexto do ensino musical, nem mesmo no da Educação ou da Música: este é, isto sim, manifestação de toda uma visão de mundo, de conhecimento, de sociedade, de ser humano. Assim, ao esboçar uma alternativa a este ensino musical, não podemos nos furtar de questionar também estes aspectos, de uma maneira **transdisciplinar**. É o que pretendemos a seguir.

A cultura erudita européia e o privilégio das atividades mentais

Como vimos o ensino tradicional se baseia principalmente da visão de homem e de conhecimento da *cultura erudita européia*, relacionada com o que muitos autores chamam *ocidental* ou, outros, *hegemônica* (ou, ainda, *burguesa*). Esta foi difundida por quase todo o planeta através da colonização na Modernidade.

Podemos situar como origem desta cultura o Império Romano, baseado na cultura grega, na religião cristã, na organização social e militar do imperialismo helênico, entre outras inflluências. Mas foi organizada, sistematizada e transformada em **doutrina** durante a Idade Média, acabando por assumir um caráter de controle, resignação e subordinação. Para negar este controle, justificado pela autoridade espiritual eclesiástica, o homem moderno negou os aspectos transcendentais da existência, colocando a razão e a matéria como única referência para o existente e o verdadeiro.

Muitos autores colocam que este ideal iluminista de privilégio da ***razão*** sobre as demais atividades humanas, apesar de ser um critério anteposto à autoridade eclesiástica, foi **resultado** do próprio trabalho da Igreja na Idade Média, como veremos. Em primeiro lugar, separou-se o corpo (material) dos aspectos imateriais, dando-lhe um caráter *inferior* (os gregos, por exemplo, privilegiavam o corpo na escultura, nos esportes), associado às *tentações do demônio* (nem a pesquisa fisiológica podia ser feita na Idade Média). Além disso, ao elaborar o Cristianismo como uma *doutrina*, ou seja, um conjunto de princípios, normas, **regras**, para serem discutidos, explicados, ensinados, servindo como base até para julgamentos legais, o que se fez foi tirar seu caráter místico, colocando-o na esfera do mental; restou à Modernidade descartar a importância do místico, do metafísico (para os gregos a *psiché*, ou alma, era também a sede do pensamento). Mais ainda: a Verdade passa a ser uma coisa única, objetiva, impessoal e capaz de ser conhecida através da razão, pressupostos fundamentais para o surgimento da *ciência moderna*.

Conhecer é saber e fazer, nos fala Ubiratan D'Ambrosio (1993, p.87). No entanto, este privilégio das atividades mentais fez com que se subestimasse a *vivência*, a descoberta coletiva dos problemas, em troca do aprendizado de **regras** prontas de resolução de problemas também prontos, do acúmulo de **informações** desvinculadas de um contexto que lhes dê significação. A própria Arte — e a Música em especial — são entendidas como um **conhecimento sistematizado**, que envolve mais o *conhecimento (mental) de regras* do que a experiência corporal, do que criatividade, emoção, sensibilidade, transcendência.

Não por acaso, a escola passa a ser o lugar da ciência moderna, do saber sem fazer, das atividades mentais. Curiosamente, mesmo o ensino inicial de ciências procura a vivência do *fazer ciência* através da experimentação, da modelagem, da resolução de problemas práticos. No caso da Arte, ainda mais importante seria a vivência do *fazer*, pois "supõe a criação de sensações ou de estados de espírito (...) carregados de vivência íntima e profunda" (Ferreira, 1975). Convém salientar que *poesia* para o grego (*poiesis*) é a ação de fazer algo, relacionado à criação (*poese*), ou seja, o *fazer criativo;* e a palavra *arte* tem origem na mesma raiz grega que a palavra *técnica — techné* (voltamos aos gregos apenas para efeito de comparação, mostrando a transformação no uso das palavras com a transformação cultural ocorrida na Idade Média e na Modernidade).

Percebemos, então, que para podermos compreender as atividades humanas precisamos incluir também os aspectos imateriais e não-mentais, justamente os aspectos relegados a segundo plano na Modernidade: emoções, intuições, estados de espírito. Justamente pela importância destes aspectos, principalmente na compreensão da Arte, e por serem nebulosos e polêmicos, cabe aqui um aprofundamento (mesmo saindo do âmbito da Computação e Música), que faremos a seguir.

Emoção, estados de espírito, transcendência, criatividade, inspiração artística e respiração

Para discutir os aspectos não-mentais, chegamos a um paradoxo: estes aspectos não podem ser compreendidos apenas mentalmente.

O artista bem sabe disso: as emoções não se descrevem, mas se sentem. O poeta ou o cineasta não se dedicam a descrever nem os estados mentais nem as sensações corporais associadas aos sentimentos, mas procuram descrever as situações que provocam estas emoções ou sensações, esperando que o leitor ou o espectador compartilhem a relação, possibilitando, sempre, múltiplas impressões, interpretações (é comum, por exemplo, descrevermos uma dor como uma *pontada*, ou *como um caminhão que passou sobre mim*). Assim, a linguagem para se tratar de emoções ou sensações é uma linguagem simbólica, metafórica.

Este mesmo paradoxo é bem conhecido pelos místicos: a experiência mística é pessoal, interior, não podendo ser relatada, transmitida. Nem transformada numa doutrina, como tentado na cultura erudita européia. Já dizia Lao Tsé, no século VI A.C., no primeiro verso do Tao Te King, livro taoísta chinês clássico (traduções variadas):

> O Tao que pode ser expresso não é o Tao absoluto
> O nome que pode ser dito não é o nome eterno
> O sem nome é a origem de Céu e Terra
> O com nome é a mãe de todas as coisas
> ...
> Este é o mistério
> Mistério ainda não desvelado
> E porta para a compreensão da maravilha do Universo

Allan Watts (1988, p.15-23) nos traz a visão do zen-budismo, onde procura-se deixar claro que os ensinamentos não são a sabedoria, a compreensão em si, mas descrições desta compreensão; esta não pode ser atingida com palavras ou idéias, mas apenas pela experiência individual; os ensinamentos são como um dedo que aponta uma direção — não devemos nos enganar indo em direção ao dedo ao invés de ir na direção em que ele aponta.

Assim, não seria objetivo aqui definir o indefinível, mas apenas esclarecer como diversos autores e culturas entendem alguns dos aspectos imateriais e não-mentais, como as emoções, os estados de espírito, a transcendência, a criatividade, a inspiração artística. E, como veremos, todos estes aspectos se inter-relacionam.

Num modelo antigo, o arquétipo dos quatro elementos — terra, água, fogo e ar —, usado por diferentes tradições, da Física aristotélica à Astrologia, é o **ar** o elemento das **atividades criativas** e da **transcendência**, da abertura para o novo e para a **ligação com o cosmos**. A idéia do **sopro** como origem da vida está presente no judaísmo, no surgimento de Adão (boneco de argila, ou seja, terra e água), símbolo do primeiro homem; aparece também nos filósofos gregos pré-socráticos, como em Anaxímenes e, de certa forma, em Heráclito (que supõe o ar ou o fogo, idéia semelhante ao *sopro quente divino*). E o fogo simboliza as emoções.

Segundo o dicionário Aurélio (Ferreira, 1975), a palavra ***alma*** se relaciona ao princípio de vida; engloba as faculdades psíquicas, intelectuais e morais de um indivíduo; também é a sede dos **afetos**, dos **sentimentos** e das **paixões**; assim como expressão, animação, criação. E *animar*, segundo ele, é **dar vida**: *Narra a Bíblia que Deus, soprando o barro, animou o homem*. (Aliás, por não haver referência do mesmo ter acontecido na criação dos outros seres, abriu-se espaço para a suposição cristão de que a alme é um privilégio humano, com suas conseqüências...) O psicólogo Carl Gustav Jung (Ratis, 1986) "utilizou a palavra latina *anima* (alma) para designar o arquétipo que no homem vinha trazer a renovação da consciência através de inspirações para atitudes criativas."

Se formos à origem da palavra ***espírito***, encontramos a palavra latina *spiritu*, representante da **parte imaterial** do ser humano, ou seja, tanto a alma como o intelecto; é usada ainda como nome (em latim) para o sinal da língua grega que indica aspiração. Esta palavra latina está associada com duas palavras gregas: ***psyché*** (alma, espírito, intelecto) e ***pneuma*** (sopro ou espírito aéreo, considerada a origem da vida). Muitas tradições preferem diferenciar *alma* (associada à *psyché*) de *espírito* (associado a *pneuma*), sendo a primeira usada como algo pessoal, individual, e o segundo algo presente em todas as coisas (ou seres humanos), relacionado à ligação com o Universo ou com Deus.

Esta ligação com o Cosmos, com a Origem (origem Ontológica, do Ser, não necessariamente origem no tempo) ou com o Criador (Deus) é o sentido da palavra latina *religare*, que originou a palavra **religião** (pelo latim, *religione*), assumindo que esta ligação, de certa forma, se rompeu. Segundo o físico Albert Einstein (1981, p.19-22), a visão de religião como ligação com um Deus antropomórfico, que preside ao destino, socorre, recompensa e castiga, tende a ser superada por uma religiosidade cósmica, difícil de falar a respeito, onde o ser deseja provar a totalidade do Ente como um todo perfeitamente inteligível. "Ela não tem dogmas nem Deus concebido à imagem do homem, portanto nenhuma Igreja ensina a religião cósmica." E, associando esta ligação com o Cosmos à criação científica, afirma "com todo o vigor que a religião cósmica é o móvel mais poderoso e mais generoso da pesquisa científica".

Nas tradições orientais, principalmente o hinduísmo, o taoísmo e o budismo, a ligação como o Cosmos, a compreensão da **Unidade do Universo**, é atingida principalmente com a **meditação**. Apesar da grande variação de práticas, costumes e crenças, nestas tradições em geral e no Zen-budismo em particular acredita-se que atingir esta Unidade depende do **ajuste da respiração**, além do ajuste da postura e da concentração.

Wilhelm Reich (1982, p.254-282) mostra a inter-relação entre a estrutura psíquica e a estrutura biofísica: prender a respiração é uma reação natural (inclusive das crianças) para lutar contra os estados de cólera, angústia ou medo, inclusive medo do prazer. A respiração introduz **oxigênio** no organismo, usado na combustão dos alimentos, fonte de sua energia (gerando calor, movimento e eletricidade). Relaciona, assim, o controle da respiração ao controle das **emoções** e da **energia vital**.

Segundo Pierre Weil (1993, p.54) "É antiga a idéia de que tudo no Universo é constituído ou é a expressão da mesma força ou energia. Essa energia era conhecida por diferentes nomes segundo as tradições espirituais: *prana*, em sânscrito, *rlung*, em tibetano, *ruach*, em hebraico, *pneuma*, em grego, *espiritus*, em latim. Autores contemporâneos também a designam com nomes diversos: *libido* de Freud e Jung, *élan vital* de Bergson, *orgone* de Wilhelm Reich." Poderíamos incluir nesta lista o termo chinês ***ch'i***, que desde a antigüidade representa "uma energia ou sopro vital que anima o cosmos" (Page, 1991, p.8) e, mais tarde, também "éter, sopro, vapor, energia ou energia-matéria" (idem, p.13); segundo Sohn (1992, p.19,28-29), também pode ser entendido como *desejo de ação, vontade de realizar*, sendo possível relacionar os termos chineses *Ki* (Energia/Matéria), *li* (Idéia) e *ch'i* à trilogia hindu *Sat* (Existência), *chit* (Consciência) e *ananda* (Êxtase), ou, em outras palavras, corpo, mente e emoção.

O objetivo deste texto não é e nem poderia ser fazer uma síntese conclusiva de todas estas visões. Mas espero que tenha sido possível mostrar que, nas mais diferentes culturas e tradições e nos mais diferentes tempos, pode-se encontrar uma relação entre respiração, emoção, estados de espírito, transcendência, criatividade e inspiração artística. E que, além de inter-relacionados, estes aspectos exigem, principalmente para a Arte, uma abordagem que vá além da dicotomia **mente-corpo**, caracterizada na Modernidade pela exposição de Descartes.

Procuramos nesta pesquisa uma palavra que pudesse comportar todos estes aspectos, o que, claro, é difícil. Alguns autores se referem a eles como ***espírito***, ***emoção*** ou ***vida***. Outra possibilidade seria a palavra ***coração***, por ser o responsável pela distribuição de energia no organismo (ar e alimento), por ser considerado a sede das emoções e por ter fundamental importância na experiência mística. Outra ainda seria a palavra ***fluxo***, representando algo **dinâmico** (como o fluxo do rio) — que só existe como movimento, troca, ação — e **cíclico** (como o fluxo das marés): circulação sangüínea, inspiração/expiração, fluxo de energia, comunicação.

Pelo menos neste trabalho, optamos pela palavra ***respiração***, por estar associada ao ar (símbolo da criatividade), à inspiração artística, ao sopro que origina a vida, a *pneuma*, à energia vital, ao controle das emoções, sendo uma metáfora para a própria vida — que só pode ser entendida como **troca** (de ar, de alimento, de informações genéticas, de idéias, de emoções) e como **ciclo** (cadeia alimentar, nascimento/reprodução/morte); é, assim, usada de uma maneira simbólica e metafórica, associada às outras possibilidades aqui apresentadas.

Música: mente, corpo, respiração

Definir o que é música é uma tarefa difícil. Mas todos nós temos uma certa compreensão (mais ou menos imediata ou consciente): que envolve a produção e organização de sons, em primeiro lugar; que é uma arte, ou seja, que envolve sensações, emoções, intuições, estados de espírito; que é individual, íntima, profunda, ao mesmo tempo que é intersubjetiva, social, coletiva; que é um fator de relação de pessoas ou grupos consigo mesmos, com outras pessoas ou grupos, com a Natureza e com o Cosmos.

Ao buscar significado, função, proposta para a música nos deparamos com muitas possibilidades. E, de acordo com as condições culturais, técnicas e econômicas, entre outras, associadas a estes objetivos, intenções, desejos, funções coletivas, surgem os mais diversos tipos de música.

Mas, como vimos, a música erudita européia se tornou referência central para definir o que é música, e separou a música da dança, dentro de uma perspectiva de associar o corpo como algo inferior, *popular* (numa conotação pejorativa).

Buscamos, então, resgatar a **função corporal** da música, através da dança, principalmente a dança criativa, expressiva, prazerosa. Também apresentamos a importância para o Ser humano e para a Arte da emoção, dos estados de espírito, da transcendência, da criatividade e da inspiração artística, representados pela **respiração**. Mas é importante pensar estes aspectos numa **unidade dinâmica**, diferenciados por questões histórico-culturais, mas não separados, isolados. Assim, criamos o símbolo apresentado na figura 1 (baseado no T'ai Chi T'u, ☯, antigo símbolo chinês) para representar a música como uma interação dinâmica da mente (representada em branco), do corpo (representada em negro) e do fluxo (que poderia ser representado por diferentes cores, mostrando sua multiplicidade interpretativa, aqui aparecendo em cinza).

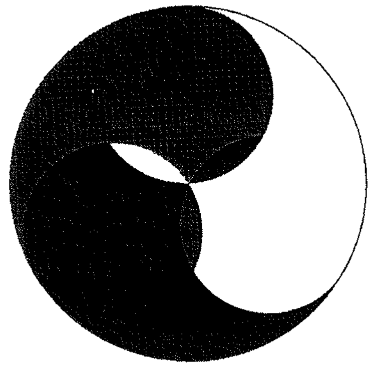


Figura 1: Música e Ser humano como a interação dinâmica mente-corpo-respiração.

Notações musicais alternativas

Como vimos o uso de diversas notações musicais integradas possibilita novas formas de se **compreender**, de se **aprender** e de se **fazer** música, assunto explorado por Guerra (1994b): da mesma forma que diversas linguagens de programação foram desenvolvidas para satisfazer as mais diversas necessidades em Computação, diversas formas de representação musical vêm sendo criadas para cobrir os mais diversos aspectos da Música e os mais diversos tipos de objetos — desde características dos sons, mais próximas do nível físico, até intenções expressivas implícitas ou explícitas, num nível mais conceitual, passando pela organização da peça musical, pela formação de escalas e acordes, pela escolha das notas, ritmos, andamentos, etc.

> Qualquer sistema de notação é deficiente para representar inteiramente todas as formas de expressão musical. Isso porque quanto mais elementos forem utilizados, maior número de símbolos serão necessários para representar o que se deseja tornando-se, muitas vezes, impraticável e confusa sua aplicação de forma integral.
> Cada sistema de notação tem, portanto, melhor aplicabilidade para determinados casos.
>
> (Cabral, 1994, p.79)

Podemos aqui apresentar alguns exemplos de linguagens e notações musicais com diferentes funções:
- para representação de solos (tablatura) e de acordes (cifras), encontradas nas revistas especializadas em violão e guitarra;
- a *notação posicional*, desenvolvida para a construção de ritmos (Moraes, 1994);
- a *representação angular*, para compreensão da formação de acordes (Cabral, 1994);
- para a manipulação de trechos musicais e para organização da peça musical (introdução, partes, refrão, solos), como a usada na *Biblioteca Maestro* (Guerra, 1992, p.47-52) e que orignou as operações musicais aqui descritas;
- várias possibilidades para uma descrição mais detalhada do som, a fim de possibilitar sua síntese artificial;
- linguagens de inteligência artificial para a organização de bases de conhecimento musical;
- para interface entre equipamentos, como a MIDI, extendida também para manipulação de arquivos.

Possibilitar o uso destes diversos sistemas notacionais num ambiente integrado é objetivo da interface AbCMus, que será apresentada adiante.

A informática aproximando o não-especialista da experimentação

O aprendizado de conceitos musicais através do *fazer música* muitas vezes esbarra na dificuldade técnica inicial na execução dos instrumentos, na grande quantidade de informações envolvidas na composição musical,

no entrosamento necessário entre diversos instrumentistas para a execução de uma peça mais elaborada. Mas os ambientes computacionais atuais oferecem recursos que possibilitam a experimentação musical pelo não-especialista, priorizando-se a produção e composição musicais. Alguns destes recursos são exploradas por Guerra (1994a) e, dentre elas, podemos salientar:
- obtenção de trechos musicais sem a necessidade do domínio da técnica de um instrumento, através, por exemplo, da **reutilização** de trechos musicais gravados por outras pessoas, da **adaptação** de gravações musicais, da **geração** automática de trechos, além das **transformações** musicais tratadas neste artigo;
- possibilidade de diversas formas de comunicação: digitação, execução de instrumentos, manipulação direta de ícones, etc;
- utilização de notações musicais alternativas às pautas musicais;
- facilidade na orquestração e na audição simultânea de vários instrumentos;
- recursos novos, como transposição de tom, troca de timbre e alteração de andamento das músicas;
- possibilidade de identificação de padrões de composição não percebidos pelo usuário;
- facilidade para ouvir e comparar o resultado de experiências musicais, melhorando sua compreensão e motivando o aprendizado e a pesquisa;

AbCMus: uma abordagem transdisciplinar

Iniciei minha investigação em Computação e Música preocupado com o desenvolvimento de um ambiente para pesquisa e aprendizado musical por parte da *pessoa comum*, do *músico diletante*, do curioso; um ambiente que não exigisse um conhecimento maior nem de música nem de informática. Percebi, no entanto, que o desenvolvimento deste ambiente não era um problema puramente tecnológico, mas estava diretamente relacionado a uma nova maneira de se conceber a Ciência, a Arte, a Educação e o Conhecimento em si.

Busco um ambiente onde o aprendizado seja baseado na **criatividade** e motivado pelo **prazer**. Onde conhecimento musical passa a ser algo **dinâmico**, **vivo** — o aprendizado das teorias musicais clássicas pode vir a acontecer como necessidade colocada pelo próprio usuário para ampliar seu domínio musical, mas não como condição ou barreira. Onde o usuário tenha à disposição **diversos sistemas notacionais** e possa construir sua música utilizando-se dos que mais lhe convierem, de acordo com seu interesse e/ou possibilidade. Busco também a **função corporal** da música, através do ritmo, da dança, do movimento prazeroso. E busco o espaço para a pesquisa do **inconsciente cognitivo**.

Assim surgiu esta abordagem transdisciplinar, que chamo ***Abordagem de Construção Musical*** ou, simplificadamente, **AbCMus**.

Nesta abordagem não há nível fundamental para a construção de uma música: não é o som, nem o acorde, nem o compasso, nem o trecho musical, nem a música como um todo. Mas os elementos de todos estes níveis se inter-relacionam para formar a música. Cada elemento, seja do tipo que for, é visto como um **trecho**, e passível de ser representado através de um **ícone**. Estes trechos podem ser divididos em três tipos básicos, como veremos mais detalhadamente a seguir: os **geradores**, que não dependem de outros; os **transformadores**, que produzem alterações em outros trechos; e os **analisadores**, que extraem informações de trechos musicais. Manipulações diretas sobre estes ícones refletem operações musicais. Podemos chamar isto de ***interface*** **AbCMus**, que será apresentada em detalhe mais adiante.

Através deste ambiente de experimentação, associado a uma **base de dados musical**, pode-se pensar na construção tanto de músicas como de conhecimento musical. Os próprios padrões de composição surgem, não como regras estéticas ou culturais prontas, imposta pela sociedade ao indivíduo, mas como resultado da **interação** entre o autor e sua obra (Guerra, 1994a).

Para efetivar esta experimentação é importante que a comunicação entre aplicativo e usuário seja a mais agradável, confiável e simples de operar, exigindo-se o mínimo de conhecimentos técnicos tanto de Computação quanto de Música; que se amplie ao máximo os recursos de experimentação; que torne acessível ao usuário vários tipos de informações musicais; e que o sistema computacional não dependa apenas dos comandos explícitos dos usuários, mas que seja capaz também de aprender automaticamente.

Sob o aspecto da **pesquisa** em Computação e Música, a AbCMus permite a integração de diversas pesquisas realizadas isoladamente: cada gerador, transformador ou analisador musical pode ser visto como um módulo, com operações próprias e tipos de dados compartilháveis. E as implementações podem se dar em diversas linguagens, procedurais ou declarativas.

Operações musicais

As operações musicais aqui apresentadas foram inicialmente desenvolvidas como parte das ferramentas para aplicativos musicais implementadas pela **Biblioteca Musical *Maestro*** (Guerra, 1992). A fim de testar e aperfeiçoar estas operações, foi desenvolvido um aplicativo chamado ***Construtor Musical Interativo*** (Guerra, 1994a). Neste aplicativo dispõe-se de 16 canais independentes, onde trechos musicais podem ser gravados ou executados via MIDI, lidos ou escritos através de arquivos e/ou transformados através de várias das operações aqui descritas. Este foi um passo importante na definição de um ambiente integrado e interativo para pesquisa e aprendizado musical, voltado também para o **não-especialista**

Como vimos, na AbCMus não há nível fundamental para a construção de uma música: cada elemento é visto como um trecho, construído através de um módulo, representado através de um ícone próprio. Assim, cada operação aqui apresentada — gerador, transformador ou analisador — pode ser vista como um módulo.

Geradores

São independentes de outros trechos. Exemplos de geradores são arquivos MIDI, amostras digitais de áudio (arquivos .WAV, por exemplo), descrições de som, representações textuais, sorteadores, geradores de música fractal, autômatos celulares, compositores automáticos de melodias, etc.

Transformadores

Produzem alterações em outros trechos. Exemplos de transformadores são:

- **mistura**: execução simultânea de uma lista de trechos;
- **concatenação**: execução em seqüência de uma lista de trechos, um após o outro;
- **repetição**: execução repetida de um mesmo trecho (pode ser vista como concatenação contínua de um mesmo trecho);
- **cópia**: dobramento do conteúdo de um trecho (geralmente antecede uma outra transformação, como transposição, podendo ser usado para execução simultânea ao trecho original ou não);
- **retrogradação**: reordenação das notas musicais do trecho, do fim para o começo;
- **inversão**: alteração dos intervalos do trecho, tornando ascendentes os descendentes e vice-versa;
- **transposição**: todos os tons do trecho são modificados um mesmo intervalo cromático (um mesmo intervalo de tom);
- **troca do timbre**: alteração do timbre (som característico de um instrumento musical) das notas do trecho;
- **andamento**: definição de uma velocidade relativa de execução do trecho;
- **base de tempo**: muda o número de batidas por compasso por alteração no tempo das notas;
- **ajuste do *delay* inicial**: ajusta a temporização do primeiro evento do trecho para posicionar o início de sua execução;
- **ajuste do *delay* final**: ajusta a temporização do último evento do trecho para, por exemplo, tornar a temporização total do trecho um múltiplo (ou sub-múltiplo) exato da temporização do compasso;

Outros transformadores definidos pelo usuário podem ser definidos e utilizados.

Analisadores

Extraem informações de trechos musicais, como contadores de compassos e batidas, contadores de tempo, contadores de instrumentos, analisadores de tom de trechos e acordes, entre outros.

Proposta de interface com usuário

Como vimos, associada a AbCMus há uma proposta de interface com usuário interativa que exija o mínimo de conhecimento técnico tanto de Computação quanto de Música. Esta deve possibilitar ao usuário se expressar das mais diversas formas — escrevendo, tocando, indicando figuras na tela, etc. —, através de diversas notações e concepções musicais, tornando o processo de criação e pesquisa o mais simples, agradável e eficiente possível.

Dentro desta abordagem, a **metáfora espacial**, usada nas interfaces de manipulação direta (*drag-and-drop*), se mostra a mais adequada: os objetos são representados por ícones (símbolos gráficos) e operações dire-

tas sobre estes ícones se refletem como operações sobre os próprios objetos. Estas operações, geralmente usando *mouse* ou algum outro dispositivo que permita apontamento e movimentação, são mais **ações físicas** (apontar, selecionar, arrastar) do que lingüísticas, mais simples (não há tantas nuances de sintaxe) e mais intuitivas.

Algumas características desta interface são apresentadas por Guerra (1994b), donde tiramos o exemplo apresentado na figura 2 a seguir: um trecho musical definido como a mistura de (1) um ritmo executado por instrumentos percussivos e desenvolvido numa notação rítmica própria, (2) uma base de piano produzida por um seqüenciador MIDI comum, (3) efeitos gravados digitalmente por amostragem e (4) um solo de um instrumento sintético semelhante a um saxofone processado digitalmente numa estação de trabalho e transferida como arquivo de som; a música, no seu estado atual, seria a execução deste trecho a 80 batidas por minuto e depois a 120 batidas por minuto.

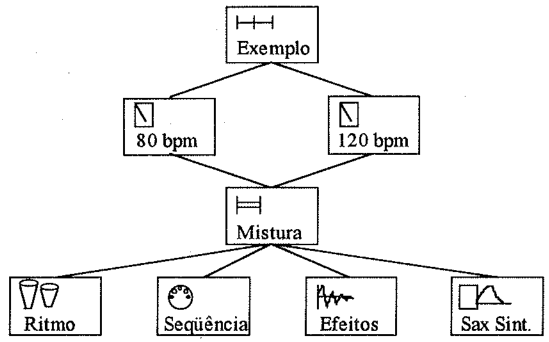


Figura 2: Exemplo simplificado de representação musical usando a interface AbCMus

No nosso caso, os ícones representam os **trechos** musicais. Cada trecho é construído usando um **módulo**. Os módulos podem ser operações básicas AbCMus — apresentadas neste artigo — ou operações da **biblioteca** AbCMus, desenvolvidas por diferentes autores. Assim, se possibilita a integração de diversas notações, ferramentas, concepções musicais: cada módulo tem um ícone genérico e um ambiente próprio de construção musical. Exemplos de ações seriam:

- apontar com um *click* do botão esquerdo: abertura do ambiente associado;
- apontar com dois *clicks* do botão esquerdo: execução musical do trecho;
- apontar com três *clicks* do botão esquerdo ou um *click* do botão central: definido pelo módulo;
- apontar com um *click* do botão direito: abertura da janela de opções do módulo;
- arrastar: cópia do módulo e de seus componentes;
- arrastar com tecla de controle (CTRL) pressionada: mover o módulo e seus componentes;
- arrastar com tecla de alteração (ALT) pressionada: ligação do módulo arrastado como componente do módulo original e também do módulo-destino;

O exemplo da figura 2 procura esclarecer que a representação musical não necessariamente corresponde a uma árvore, mas é melhor representada como um grafo com um nó inicial, por onde inicia a execução da música (possibilidades de ciclo ainda não foram analisadas), uma vez que um mesmo trecho pode ser repetido em diferentes situações ao longo da música.

Para que os módulos autônomos possam ser integrados no sistema é preciso definir-se claramente o modo de comunicação entre eles, a fim de que se possam executar as operações comuns, como execução (via equipamento MIDI, por síntese, etc.), aplicação das transformações musicais (como alteração do andamento, ajustes de temporização) e leitura e gravação usando arquivos.

Cabe salientar que a receptividade dos ambientes multimídia tem simplificado a implementação de programas musicais — como a definição de arquivos MIDI ou a interface para controle de mídia MCI do *Windows* para IBM-PC-compatíveis —, simplificando, também, a compatibilização de módulos.

Conclusão

Não podemos situar o problema do ensino tradicional de música apenas no contexto do ensino musical, nem mesmo no da Educação ou da Música: este é, isto sim, manifestação de toda uma visão de Mundo, de Conhecimento, de Sociedade, de Ser humano. Assim, ao esboçar uma alternativa a este ensino musical, não podemos nos furtar de questionar também estes aspectos, de uma maneira **transdisciplinar**.

Neste sentido foi desenvolvida a AbCMus, propondo a construção do conhecimento musical através da **vivência direta** de cada um, procurando a integração dinâmica de *mente*, *corpo* e *respiração* — esta, simbólica e metaforicamente, representando a emoção, os estados de espírito, a transcendência, a criatividade e a inspiração artística.

Em termos computacionais, a AbCMus propõe um ambiente interativo, acessível ao não-especialista nem em Música nem em Computação, onde diversos sistemas notacionais estejam integrados através de uma interface de manipulação direta, que implemente um conjunto de operações musicais aqui apresentados e que possibilite a integração de novos módulos, com linguagens, interfaces e abordagens musicais próprias.

Alguns passos para o desenvolvimento deste ambiente foram aqui propostos, esperando por críticas e sugestões. A efetivação deste ambiente, no entanto, depende do interesse de pesquisadores em Computação e Música em relacionar suas pesquisas à abordagem aqui apresentada, uma abordagem também em construção.

Bibliografia citada

Cabral, Edilson E. (1994). Representação angular para notação musical. In: Simpósio Brasileiro de Computação e Música (1. : 1994 : Caxambu). *Anais*... Caxambu: UFMG. p.79-81.

D'Ambrosio, Ubiratan. (1993). A transdisciplinaridade como acesso a uma história holística. In: Weil, Pierre; D'Ambrósio, Ubiratan; Crema, Roberto. *Rumo à nova transdisciplinaridade: sistemas abertos de conhecimento*. São Paulo: Summus.

Einstein, Albert. (1981). *Como vejo o mundo*. 6. ed. Rio de Janeiro: Nova Fronteira. Trad. H. P. de Andrade.

Ferreira, Aurélio. (1975). *Novo dicionário da língua portuguesa*. 1. ed. Rio de Janeiro: Nova Fronteira.

Guerra, Carlos Gustavo M. (1992). *Maestro: uma biblioteca básica para música computacional*. Porto Alegre: UFRGS. Projeto de Graduação.

_____. (1994a). Aprendizado e pesquisa musical auxiliados por computador. In: Simpósio Brasileiro de Informática Educativa (5. : 1994 : Porto Alegre). *Anais*... Porto Alegre: Sociedade Brasileira de Computação — Instituto de Informática PUC/RS. p.38-50.

_____. (1994b). Desenvolvimento de interfaces gráficas para notações musicais alternativas. In: Simpósio Brasileiro de Informática Educativa (5. : 1994 : Porto Alegre). *Anais*... Porto Alegre: Sociedade Brasileira de Computação — Instituto de Informática PUC/RS. p.271-275.

_____. (1995). Uma concepção integrada, criativa e prazerosa para a ciência, a arte o trabalho e a vida. In: *Coleção Laboratório*. Educação: tramas e temas. Florianópolis: UFSC/CED/NUP, n. 2. p.57-66.

Moraes, Marcos R. (1994). Positional rhythmic notation: an implication for a positional theory of rhythm. In: Simpósio Brasileiro de Computação e Música (1. : 1994 : Caxambu). *Anais*... Caxambu: UFMG. p.83-88.

Page, Michael. (1991). *Ch'i — energia vital*. São Paulo: Pensamento. Trad. Beatriz Sidou.

Ratis, Pedro. (1986). *Jung, a gente se vê em Olinda*. São Paulo: Brasiliense.

Reich, Wilhelm. (1982). *A Função do orgasmo* — problemas econômico-sexuais da energia biológica. 8. ed. São Paulo: Brasiliense. Trad. Maria da Glória Novak.

Sohn, Robert. (1992). *O Tao e o T'ai Chi Kung*. São Paulo: Pensamento. Trad. Carlos A. L. Salum e Ana Lúcia Franco.

Weil, Pierre. (1993). Axiomática transdisciplinar para um novo paradigma holístico. In: Weil, Pierre; D'Ambrósio, Ubiratan; Crema, Roberto. *Rumo à nova transdisciplinaridade: sistemas abertos de conhecimento*. São Paulo: Summus.

Lexikon-Sonate. An Interactive Realtime Composition for Computer-Controlled Piano

KARLHEINZ ESSL

SAMT - Studio for Advanced Music Technology (Hagenberg/Linz – Austria – Europe)

essl@ping.at – http://www.ping.at/users/essl/

Abstract

Lexikon-Sonate is a work in progress which was started in 1992. Instead of being a composition in which the structure is fixed by notation, it manifests itself as a com-puter program that composes the piece – or, more precisely: an excerpt of a virtually endless piano piece – in real time. *Lexikon-Sonate* lacks two characteristics of a tra-ditional piano piece: 1) there is no pre-composed text to be interpreted, and 2) there is no need for an interpreter. Instead, the instructions for playing the piano – the indication "which key should be pressed how quickly and held down for how long" – are directly generated by a computer program and transmitted immediately to a player piano which executes them. In this paper I will describe from where I started and how I arrived at the concept of an infinite interactive realtime composition.

Origins

In the late sixties the Austrian/Slovakian poet Andreas Okopenko started to write the novel *„Lexikon-Roman"* (Okopenko, 1970) – the first literary HyperText, several years before this term was introduced by Ted Nelson (Nelson, 1970). This novel – „a sentimental journey to a meeting of exporters in Druden" (subtitle) – consists of several hundred small chapters which were brought into alphabetical order. By reference arrows as in a lexicon the reader could make her own investigations through the multiple nested web structure of the text. Instead of presenting a sequential text with a predefined direction of reading, Okopenko provides a structure of possibilities, which challenges the reader to become a creator of her own version of this novel.

Twenty-five years later an interdisciplinary group of artists and computer freaks called *„Libraries of the Mind"* started to create an electronic version of this book using HyperCard as a programming environment. Now the navigation through the text was easily achieved by clicking onto the reference arrows, the „links". The electronic implementation (which is about to manifest itself as a CD-ROM) also provides new features that were impossible with a printed book: an electronic logfile which keeps track of the ways and deviations of the reader, search for keywords, the possibility of making annotations etc.

Andreas Okopenko, who himself belongs to the „Libraries", suggested adding other media like pictures, photos, spoken language, music and sound. And so other artists joined the group: a graphic artist, a photographer, and at last myself, a composer.

After reading the book three demands for the music became obvious:

(1) Music for the „Lexikon-Roman" cannot consist of „jingles" which are played whenever a certain text particle has been selected. With music the problem of time emerges: music – unlike a static pictorial object or even a text – is always related to time: it takes place „in time", whereas beholding a picture or reading a text happens „out of time". One can meditate over a poem for a long time, or just read over it. But music is always linked to a certain time span, reflecting time. So it became clear that the music cannot consist of pre-recorded pieces that are simply recalled. It should reflect the reading behaviour of the reader: if she spends a long time on a chapter, the music should stay in the same „mood" or character, and if she starts zapping nervously between the textural links, this should also be reflected by the music, resulting in quick changes of character.

(2) The complex structure of the novel challenged me to achieve something related in musical composition: a complex network of musical meanings, an infinite maze of sounds.

(3) The lexical principle of references – starting at a certain point and arriving somewhere else by reference arrows – gave me an idea of the formal aspect of the composition. If the music changes, this change should not be abrupt, but taking some aspects of its former state and perpetuate it, while something new is added. Consider you are making a transition from A to B to C – for instance, when you are reading an encyclopaedia starting with the keyword „A" which leads you to „B" by a link, and from there to „C". There is a semantical relationship between A and B, but to a lesser extent between A and C. When you are in the B state, you will still remember A which provided the reference; and when you approach C, A will still be present, but only to a lesser extent. If you dare move towards D, you will probably forget about A. Indeed, this lexical concept of links is the underlying formal principle of *Lexikon-Sonate*.

Piano Music

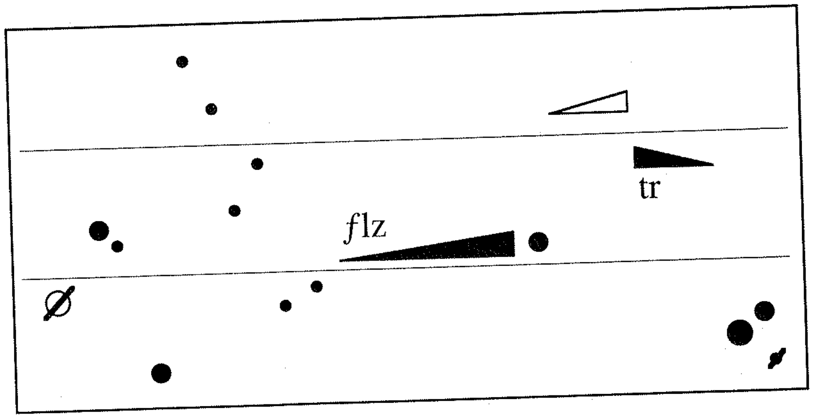
I confess that I have serious problems with the piano. As a composer and a double bass player I am mostly interested in sound processes, whereas the piano does not offer much flexibity in sound production: once the key is pressed, nothing can be done to shape the sound afterwards, as opposed to a bowed instrument, for example.

On the other hand, the piano has a big advantage: as a polyphonic instrument it allows different voices to be played at the same time. Due to its equal timbral characteristic it is predestinated to represent the structure of traditional music. In this respect it was utilized during the last 250 years. Before the development of radio and records, piano transcriptions were used to obtain an acoustical impression of a symphony or opera.

The decision to write a piano piece can also be seen in another light: taking revenge for the piano lessons I had to take since my early childhood. Using a piano also implies awareness of its incorporated history: its role in the bourgeois salon, as an inspirational tool for a composer, and as a handy instrument to unload emotional energies. At last: writing for the piano means to reflect on the whole history of this instrument, its repertory, its highly developed virtuoso techniques and its typical compositional subjects. Writing a solo piece for this beloved and hated instrument must result in a „hyper"-piano piece which increases its historical, social and compositional implications; a music beyond the scope, virtually never-ending, which exceeds the facilities of a human player. A composition, that can only be executed by a computer-controlled piano.

Real Time Composition

For a long time I have had a vision of an infinite music which is „composing itself" without lacking a personal style and interesting flavor. I had some theoretical ideas about how to achieve this goal, and I even developed a set of playing rules for the performance project *„Partikel-Bewegungen"* (1991-94) for flute, bass clarinet and saxophone. In this piece each musician plays an independent, graphically notated part which was generated and printed by a computer program written in my own xLOGO-based „Computer Aided Composition Environment" which I have been developing since 1988. Each class of graphical signs can be interpreted according to a certain set of rules, which only gives a rough outline. The „fine tuning", however, is achieved by the musicians themselves during the performance – in real time – by listening to each other and coordinating or even juxtaposing their playing with the sounds of the others.



Partikel-Bewegungen (1991-93) for flute, bass clarinet & saxophone
page from one of the instrumental parts (duration: 20 ")

When I started to work on a commission at IRCAM in 1992 (*Entsagung* for flute, bass clarinet, prepared piano, percussion and the IRCAM Musical Workstation), I came across MAX, a „graphical development environment for multimedia and music" (IRCAM/ Opcode). Immediately I realised that this was the very programing language I was looking for since a long time – a powerful tool which allows to experiment with compositional strategies and to listen to the result immediately.

Real Time Composition Library

First I started to re-implement some objects which already existed within my own xLOGO-based „Computer Aided Composition Environment", like specialized random functions and rhythm generators. The realtime facilities of MAX offered me the fantastic possibility of rapid prototyping and refining after listening, whereas in my xLOGO-environment a transcription into musical notation had to be done.

The result was so compelling and encouraging that I began to implement some models for algorithmic composition, which later became the starting point for *Lexikon-Sonate*. As a side-effect a whole library of compositional tools – the „Real Time Composition Library" (RTC-lib) for MAX – evolved, whose version 2.0 is available from several ftp-sites (see appendix).

This library offers the possibility to experiment with a number of compositional techniques, such as serial procedures, permutations and controlled randomness. Most of these objects are geared towards straightforward processing of data. By using these specialized objects together in a patch, programming becomes much clearer and easier. Many functions that are often useful in algorithmic composition are provided with this library – allowing the composer to concentrate on the composition rather than the programming aspects.

Regardless of the fact that this library was conceived for a particular project it became more and more obvious that its functionalities are open and generic enough to be used by other composers in different compositional contexts.

Although the theoretical point of view of the library is based on paradigms which have been extracted from serialism and its further developments until today, it does not force a single aesthetic, but provides a programming environment for testing and developing musical strategies. „Serialism" here refers to a certain method of musical thinking rather than orthodox dodecaphonic techniques which has been abandoned by serial theory itself (cf. Stockhausen, 1957 and Koenig, 1965).

The library is composed of two main categories of objects: basic programming tools (like toolbox functions, chance and list operations) and specific musical functions (harmony, rhythm, envelopes) – see Fig. 1.

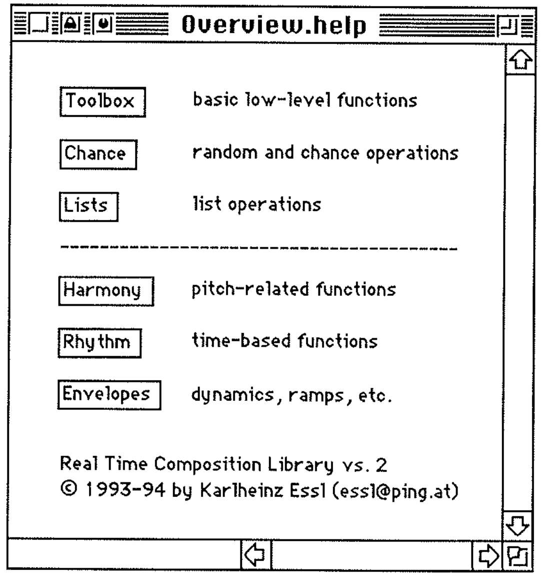


Fig. 1: Content of RTC-lib 2.0

As an example I will discuss the „group-rhythm"-object which generates a rhythmic structure according to Stockhausen's „Gruppen"-theory (Stockhausen, 1957) and takes into account the concept of „periodicity" as it was formulated by Gottfried Michael Koenig (Koenig, 1965). These concepts indicated the end of the orthodox „punctual" serialism and finally led to the abolition of row permutation techniques. Instead of a permutation program which was derived from a given basic row, Koenig introduced the method of random selection as it manifestates itself first in his *Streichquartett 1959* and later in his composition program *Project 1* (Koenig, 1970 and Essl, 1989).

As its basic material this rhythm generator uses a supply of entry delays (ED) which form a geometrical row between a minimum and a maximum ED in a certain number of values. In our example the min.ED is 100 ms, the max.ED is 1000 ms – between these boundaries a geometrical row is constructed:

```
row index:     0    1    2    3    4    5    6
entry delay: 100  464  215  316  464  681  1000
```

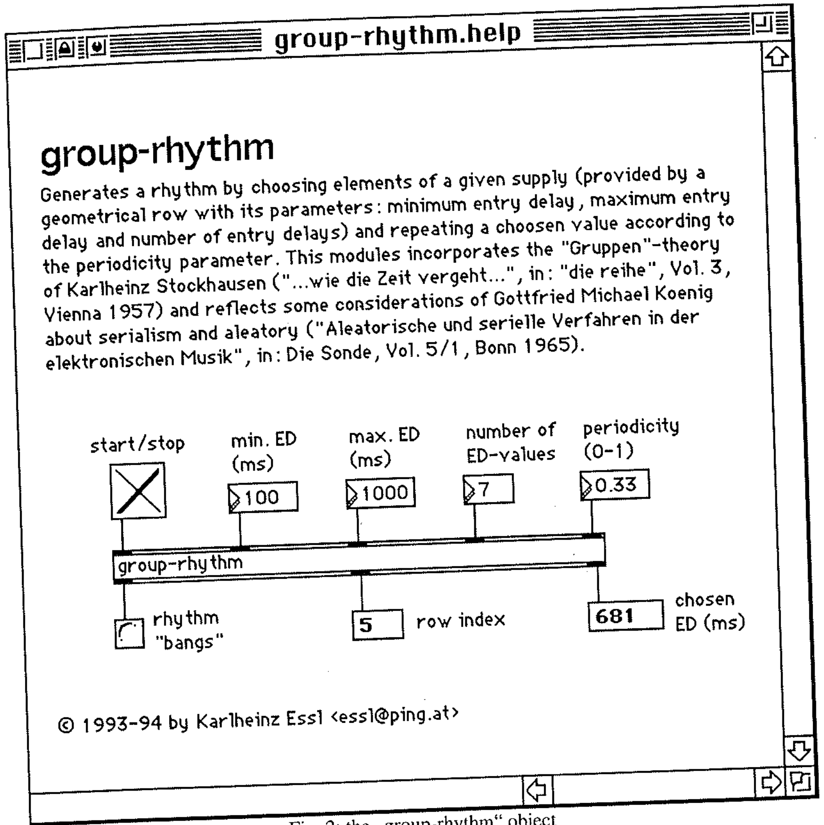


Fig. 2: the „group-rhythm" object

When „group-rhythm" is switched on, it chooses an ED-value by chance. Now the „periodicity factor" determines how often this value will be repeated, before another one is chosen. When the factor is 1, the resulting rhythm will be completely periodic – an even pulsation. If the periodicity factor becomes 0, a completely aperiodic rhythm with no repetitions of a chosen ED is generated. In between these boundaries of pure „periodic" or „aperiodic" rhythms a broad field of interesting intermediate steps is situated. By gradually changing the periodictiy parameter over the time, transitions between different grades of (a)periodicity can be achieved easily.

These specialised generators of the RTC-lib are functional implementations of a certain algorithmic model whose „behaviour" can be changed by the model's parameters. In this way an infinite variety of variants can be produced, which are always linked to the central idea of the model, even when the results are very different. Combining different RTC-generators in one patch, is a convenient way to implement specific algorithmic compositional models, as will be shown in the following chapter.

Modules

Up to now the *Lexikon-Sonate* consists of 24 music-generating modules which are related in a very complex way. Each module generates a specific and perceptual characteristic musical output (a „language") due to a certain compositional strategy applied. A module represents an abstract model of a certain musical behaviour. It does not contain any pre-organised musical material, but a formal description of it and the methods how it is being processed. The idea of autopoiësis – material organizing itself due to specific constraints – plays an important rule.

By using different random generators which are controlling each other (which – serially thought – form a scale between a completely deterministic and a completely chaotic behaviour) new variants of the same model are generated. Variants may differ dramatically from each other, although they are always perceptable as „instances" of the given structural model.

One of the simplest (but nevertheless most compelling) modules of *Lexikon-Sonate* can be found in ESPRIT:

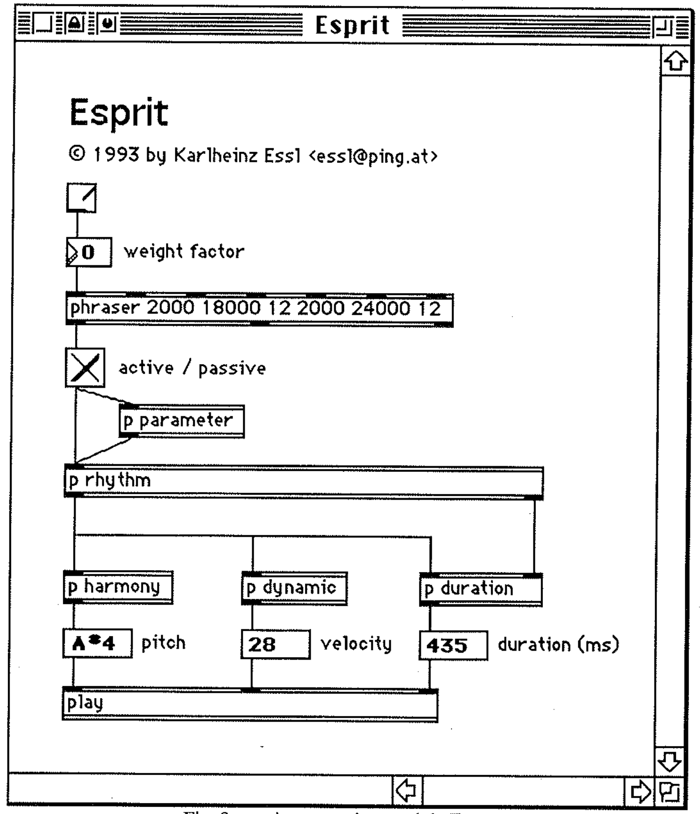


Fig. 3: music-generating module ESPRIT

This module generates melodies with a pronounced „espressivo" character. Before investigating what „espressivo" means and how it is achieved let us first look at the flow diagram of this patch; a basic structure, that appears in nearly all modules of *Lexikon-Sonate.*

From top to bottom we notice the following object boxes which are connected with lines:

phraser – alternatively generates phrases (AD = „Aktionsdauer") and pauses (PD = „Pausendauer") of a certain length. The concept of „Aktionsdauern" (time filled with sound) and „Pausendauer" (empty time, without sound) was developed by Karlheinz Stockhausen (Henck, 1980) in order to control structural „density". In *Lexikon-Sonate* the statistical time proportions between AD and PD are controlled by a so-called „weight factor", an integer between 1 and 3.

weight factor	AD	PD	perception level
1	short	long	background
2	medium	medium	middleground
3	long	short	foreground

In other words: the weight factor determines the perceptual level of a module – whether it is dominantly playing in the foreground, being modest in the middleground, or hiding itself in the background.
parameter – before generating a new phrase, all parameters of the module are randomly changed within pre-defined boundaries. With this new set of parameters rhythm, harmony, dynamic and duration are calculated.

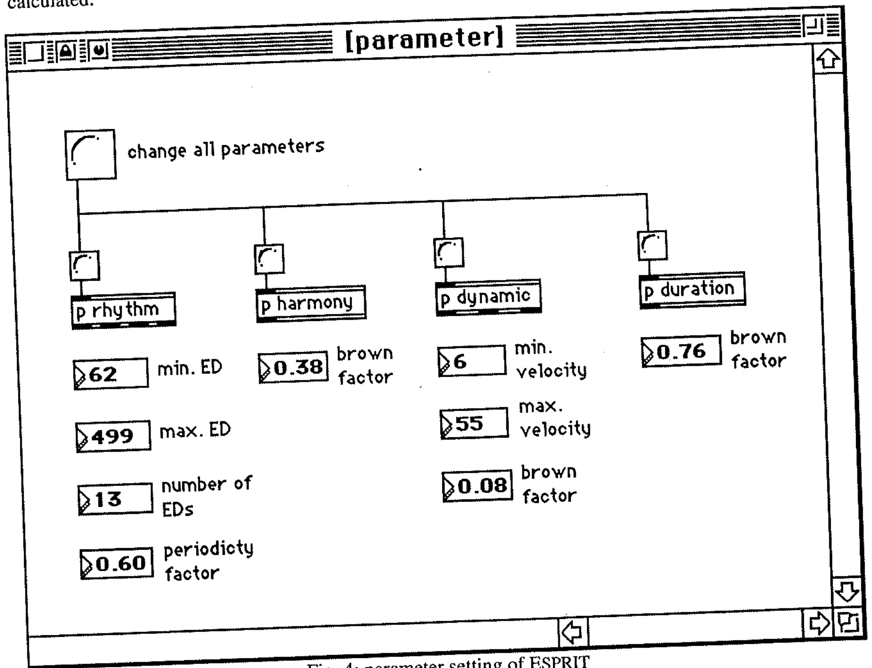


Fig. 4: parameter setting of ESPRIT

rhythm – generates a sequence of rhythm pulses („bangs", in the terminology of MAX) during the length of an AD. Each of these rhythmic bangs marks an entry point of a note whose parameters of pitch, velocity and length are calculated by the harmony-, dynamic-, and duration-objects. In ESPRIT, the `group-rhythm` object, as discussed above, is used (see Fig. 3).
harmony – a rhythm „bang" sent to the harmony object causes it to generate a pitch. In ESPRIT, the harmony algorithm uses the random generator `brownian` which selects a number within defined boundaries (min, max) according to a brown factor. With this factor (a real number between 0 and 1) the statistical distance between consecutive values is determined – the „Freiheitsgrad".

brown factor	effect
0	always repeats the same value
1	each value between min. and max. can be chosen

In order to filter out tone repetitions, octaves, and oscillating pitches, the resulting stream of pitches is evaluated by two objects, `anti-octave&prime` and `anti-bis&osc`. If such an undesireable event were about to take place the pitch is surpressed and `brownian` is asked to supply another one that fits into the constraints. This method avoids disturbing musical effects of a not-so-smart harmonical algorithm.
dynamic – uses `brownian` to generate velocities between boundaries that are defined by the `parameter` object (see above). Due to the fact that the velocity value depends on the value chosen before (according to the „brownian factor") envelope shapes like crescendo, decrescendo can occur.
duration – uses `brownian` to determine the length of the note. By this the articulation („phrasing") of the melody is controlled: whether a phrase is comprised of legato, portato, staccato, pedal effects etc. or any combinations of them.

play - combines pitch, velocity and length into a MIDI note message which is sent to the MIDI-controlled piano.

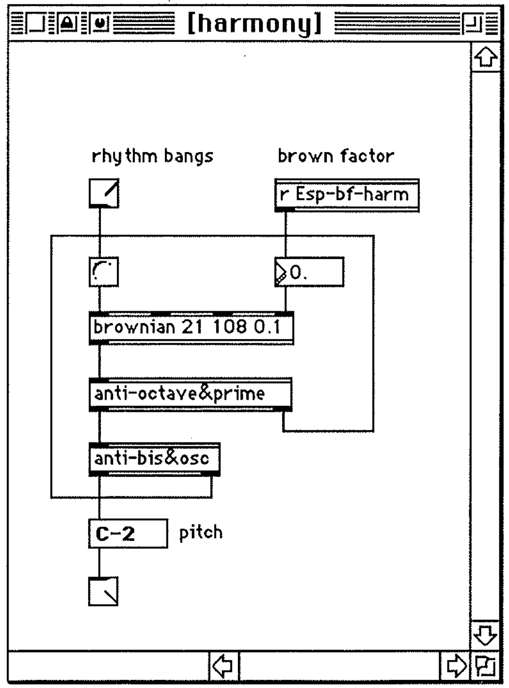


Fig. 5: harmony generator of ESPRIT

Although the music generation is achieved by random operations, it will never sound like this. In rhythm the repetitions of values create „islands of periodicity" within a complex (chaotic) situation; whereas in harmony, dynamic and duration the randomness is controlled by the „brown factor" leading to the emergences of melodic cells, envelope shapes, articulated phrases etc. By this the desired „espressivo" character – with its association to Viennese music since Mozart until Schoenberg, Webern and Berg – emerges; not at will, but as a consequence of a particular compositional strategy.

The 24 Modules

The 24 different music-generating modules of the *Lexikon-Sonate* can be assigned to 5 different types of musical structures. Superimpositions may occur. These structural types are:

- melody
- chord
- texture
- repetition
- pauses

In the following all modules are listed, together with a short description; furthermore their relationship to the 5 types of musical structures and references („cf.") to other modules with similiar properties are given.

ARPEGGIO: chord / texture (cf. GLISSANDO)
Arpeggios of 4 to 11 notes, ascending or descending, which are built up of 2 – 4 different intervals.
BROWNCHORDS: chords (cf. PULSCHORDS)
1–6part chords on a brownian rhythm. The harmonic structure is built of intervalic rows.
CLOUDS: texture / melody; (cf. TRILLER)

„Clouds" of fast moving notes (*„rubato rhythm"*) within a certain pitch range. Some notes are highlighted by dynamic accents.
ESPRIT: melody
Espressivo-melody of some complexity, refering to the so-called „Viennese Espressivo".
FERMATA: pause
Inserts global pauses into the whole musical stream and sustains the notes at these „cutting points" as resonances with the sustain pedal.
FIGUREN: melody / texture
Grace-note figures with *crescendo* towards the main note.
GLISSANDI: texture; (cf. ARPEGGIO)
Glissandi, composed of minor and major seconds which are sustained with the pedal.
GENERALPAUSE: pause
Entirely stops the stream of music. By this the whole infinite process of music generation will be organized in sections.
GRUPPEN: melody, repetition
Parameters organized in „Gruppen" according to serial theory (see above). Each of the structural parameters rhythm, harmony, dynamic range and duration factor obtain their own, individual periodicity factor which determines whether the respective parameter value is being kept for a longer period („periodicity") or is changing immediately („aperiodicity").
HACKER: pause; (cf. FERMATA)
Interrupts the global stream of music by inserting short breaks (statistically shorter than FERMATA).
HOQUETUS: melody / texture
Periodic rubato-rhythm with constantly changing registers and generally loud dynamics. The harmony is built up from a supply of two intervals.
JOYCE: melody; (cf. MOTIV)
Music obtained from a radical choice approach: for each section values from predefined parameter supplies are choosen (by the selection principles ALEA, SERIES or SEQUENCE) to build up a musical structure.
MELOCHORD: melody / chord
Structural transition between melody and chord. Depending on the duration of the entry delay (ED) the repetition rate of the chosen ED and the chord size are determined, according to the following relation:

- shorter ED: high repetition rate, small chord size
- longer ED: low repetition rate, large chord size

MOTIV: melody; (cf. JOYCE)
For each phrase different sequences of parameter values (for rhythm, harmony, dynamic and duration) are calculated which are „looped" for the duration of this very phrase. By this method the concept of „motives" (as it traditionally appears in rhythmical-harmonical contexts) is extended to other structural parameters.
ORGELPUNKT: repetition; (cf. REPLAY)
A repeated note which is dynamically increasing and decreasing.
PAUSE: pause
Simply does nothing at all. Like GENERALPAUSE, HACKER and FERMATE it allows that only two different modules are combined, instead of three.
POINTILIST: melody
Parameters changing each note („Punktuelle Musik"). Parameter ranges and row sizes may change.
PULSCHORDS: chord (cf. BROWNCOCHORD)
Up to 6-part chords on a constant rhythmical pulsation of different speed. The harmony is built of intervalic rows of different sizes.
REPLAY: repetition / texture; (cf. ORGELPUNKT)
Layers of repeated notes of different speeds which are dynamically increasing and decreasing. The harmonic structure is composed of two or three different intervals.
RÊVERIE: melody
Melodic line of complex *rubati* with moving harmonic constellations.
RICOCHET: repetition; (cf. ORGELPUNKT)
Repetitions of a single note with increasing or decreasing speed with crescendo or decrescendo.
SLEEP: melody / pause
Most of the time it does nothing (like PAUSE), but sometimes it plays a short melodic phrase.
SUSPCHORDS: chord / repetition
1 – 4 part legato-chords on a constant pulse which is structured by rests. The harmonic structure consists of intervallic rows where several note may occur in the next chord (harmonic „suspension").
TRILLER: texture; (cf. CLOUDS)

Thrills of 2 – 6 notes, dynamically increasing and decreasing, mostly together with *accelerando* or *ritardando*. NB: A six-note thrill consists of rapid permutations of six notes within a single octave register.

As stated before these 24 modules are forming a sort of musical HyperText. At the „boundaries" of its algorithmic model a module can obtain characteristics of another one, giving a reference („link") to it. Some examples:

- A variant of ESPRIT which is only composed of fast rhythmical values would give us the same impression as a structure generated by CLOUDS.
- A phrase of BROWNCHORD which only contains chords of chord size=1 could be similar to a melodic phrase generated by RÊVERIE.
- A thrill of six notes (a permutation of a set of six notes within an octave) could be similar to a structure generated by CLOUDS.

This shows that those modules are not closed entities with an exclusive behaviour – they are linked to each other in a very complex way by references. When modules are being combined during the piece, they are acting completely independent of each other. They don't „know" what the others are doing. Hence, by the process of perception the listener will relate some structural aspects of different modules to each other, composing her own „version" of the piece. This approach, as it is viewed by *„Radical Constructivismus"* (Essl, 1992), entitles the listener to become a „composer" – constructing the piece in her mind by finding an individual way through a polyvalent maze. A way, that is determined rather by personal criteria of the observer than by objective structures.

Combinations of Modules

During the piece up to three different modules are combined whereas each of them occupies a different „weight". This weight factor will determine the statistical proportions between „Aktionsdauern" (time filled with music) and „Pausendauern" (empty time) of a module, as it was shown before in the discussion of the „`phraser`"-object of the module ESPRIT. In other words: the weight factor determines the perceptual importance of a module.

When combining modules in *Lexikon-Sonate*, there will always be one in the „foreground" (weight = 3), one in the „middleground" (weight = 2), and one in the „background" (weight = 1). The weight for each module will change whenever a new one is brought into the game: e.g. a „foreground" structure could turn into a „background" or the „middleground" into the „foreground" etc.

The combination of modules takes place in a chain of three boxes, which is filled by the chosen modules, from top to bottom. On the right side of each box the number refers to the weight of the position. In the example shown below, the first box always occupies weight 3 (= foreground), the middle 2 (= middleground), and the last 1 (background).

For example, imagine ESPRIT has been chosen. It is being put into the first box to which the weight=3 is associated. Hence, ESPRIT will be played as a foreground struture: long melodic phrases, interrupted by short pauses (see above).

chosen module	„weight"
Esprit	3
	2
	1

After a certain time the next module FIGUREN is selected. Now ESPRIT will be transfered to the second position, which holds the weight 2, whereas FIGUREN will be played as a foreground structure with weight 3. The „influence" of ESPRIT becomes weaker, being displaced by the recently entered FIGUREN.

Figuren	3
Esprit	2
	1

Finally, BROWNCHORDS enters the scene, taking over the foreground. ESPRIT is turned into the background, and the weight of FIGUREN is decreased.

BrownChord	3
Figuren	2
Esprit	1

When ARPEGGIO is put into the chain, ESPRIT is abandoned. We have seen that this module – which started as foreground – gradually lost its power, becoming weaker and weaker until it was dropped completely. We also notice a formal transition: in the beginning ESPRIT alone, then in counterpoint with FIGUREN, and at last a polyphony of three different modules. Since the occurence of ARPEGGIO, however, a situation is established, where two modules are kept (BROWNCHORDS and FIGUREN) as a „memory" of the previous situation.

Arpeggio	3
BrownChord	2
Figuren	1

With the entrance of JOYCE, FIGUREN is cancelled. Again we notice a formal shift, where two compounds of the recent musical situation are maintained, while a new aspect is brought into the game.

Joyce	3
Arpeggio	2
BrownChord	1

The different modules are chosen by random – an already picked module is being blocked until all others are selected. Whenever GENERALPAUSE occurs, all active modules will be switched off, resulting in silence. By this the infinite process of *Lexikon-Sonate* will be articulated in „movements". Afterwards the sequence of weights will be mixed again (now – instead of 3–2–1 perhaps 2–3–1 in) which will serve as a formal principle for the next movement.

User Interface

Now, after knowing enough about the elements of *Lexikon-Sonate*, let us finally have a view to its user interface (see Fig. 6). On the top one can find a box named `control` with some buttons attached to it:

- `auto`: a toggle which switches on the automatic playing mode;
- `add module`: whenever this button is clicked, a new module will be chosen by random and put into the chain;
- `change weight`: chooses another sequence of weight factors;
- `stop`: turns off the whole machine.

The „`control`" box serves as a sort of conductor which gives cues to the music modules, that are placed below. Clicking on the auto-toggle starts the simplest performance mode: the built-in conductor will entirely take control, selecting the modules in time and switching them on and off. If one wants to influence the behaviour of the machine, one can click on the „add module" button. Whenever this is done, a new module will be chosen by random and combined with the two others that are still active. Clicking on the „change weight" button will change the weight factors as they are applied to the module, determining whether a chosen module serves as a foreground, middleground or background structure. This can be seen at the bottom of the display – the three boxes forming the „chain" that was discussed in the previous chapter.

However, there are also more advanced levels of interaction. Instead of merely requesting a change to take place (by clicking on the „add module" button), one can decide at will, which modules shall be combined. This is achieved by choosing a module from the „select a module" pop-up menu, which will be sent into the

combination chain. Furthermore, the sequence of weight factors can be determined, by selecting it from the other pop-up menu „select a sequence of weights".

But one can also by-pass the combination chain which allows only three modules at a time to be played, and with different „weights". By opening a module itself (by double-clicking on its box), one can directly access its parameters, choosing various settings and all the possibilities of combinations.

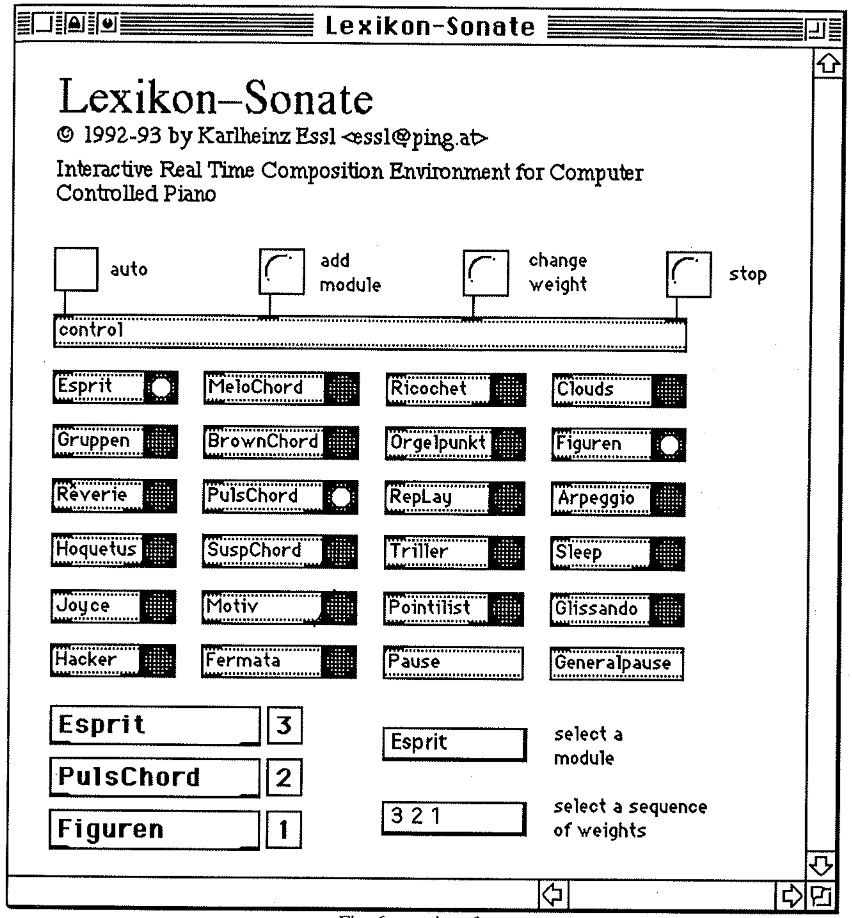


Fig. 6: user interface

Performance Aspects

The fact that *Lexikon-Sonate* never repeats itself creates a challenge to invent a particular performance situation that utilizes its interactive facilities. The premiere took place on February 2, 1994 in the concert hall of the Austrian Radio as a live broadcast during the radio program „Kunstradio – Radiokunst". On stage there was the only the fantastic „Bösendorfer SE Grand Piano", but no player at all. The radio listeners (who were not sitting in the concert space) nevertheless had the possibility to interact with the computer program by dialing a certain telephone number. Whenever a call came through, *Lexikon-Sonate* would change its compositional behaviour by adding a new and randomly selected module into its combination chain. In this way the totality of radio listeners would „govern" the form of the music, even though nobody could know the exact effect of their contribution.

At a lecture I once asked two persons from the audience to come on stage and sit there, back to back, so that they could not see each other. By giving signs with their hands, they indicated when they desired a change in music. Although these persons could not see each other, they could hear when the other had required a change – this led to a situation where the both started to „play" with each other, resulting in a wonderful and energetic performance.

References

Essl, K. (**1989**). Zufall und Notwendigkeit. Anmerkungen zu Gottfried Michael Koenigs Streichquartett 1959 vor dem Hintergrund seiner kompositionstheoretischen Überlegungen. *Musik-Konzepte*, 66 „Gottfried Michael Koenig". Munich: edition text + kritik.

Essl, K. (**1992**). Kompositorische Konsequenzen des Radikalen Konstruktivismus. *Positionen. Beiträge zur neuen Musik*, 11.

Henck, H. (**1980**). *Karlheinz Stockhausens Klavierstück X. Ein Beitrag zum Verständnis der seriellen Kompositionstechnik.* Cologne: Neuland Musikverlag, 19–23.

Koenig, G.M. (**1965**). Serielle und aleatorische Verfahren in der elektronischen Musik. *Die Sonde*, 5 (1).

Nelson, T. (**1970**). Literary Machines. Swarthmore/PA: self-published

Okopenko, A. (**1970**). *Lexikon-Roman einer sentimentalen Reise zum Exporteurtreffen in Druden.* Salzburg: Residenz Verlag

Koenig, G.M. (**1970**). Project 1. *Electronic Music Reports*, 2. Utrecht: Instituut voor Sonology.

Stockhausen, K. (**1957**). ...wie die Zeit vergeht.... *die Reihe. Informationen über serielle Musik*, 3. Vienna: Universal Edition

Acknowledges

I want to thank Bob Willey (University of California San Diego), Dennis Patrick (University of Toronto, Faculty of Music), and Gerhard Eckel (IRCAM, Paris) for their discussions which helped me to formulate this article.

Appendix

The software described in the paper runs on an Apple Macintosh computer and requires Max 2.5 (© by IRCAM /Opcode) or later. It is in the public domain and available via ftp or WWW.

Real Time Composition Library for Max 2.5
Currently a version 2.0 of the „*Real Time Composition Library*" for MAX 2.5 is available from the following ftp-sites:
(a) ftp.ircam.fr, /pub/IRCAM/programs/max/patches/composition/RTClib2.0.sea.hqx
(b) kahless.isca.uiowa.edu, /ftp/pub/max/RTC-lib_2.0.sea.hqx
(c) ftp.mars.let.uva.nl, /pub/software/RTC-lib_2.0.sea.hqx

Lexikon-Sonate is available as:
(1) MAX program: A diminished version specially designed for the „Yamaha DisKlavier" can be obtained from the „Disklavier Archive" which is maintained by Bob Willey (http://crca-www.ucsd.edu/bobw/disklavier.html). It can be retrieved via anonymous ftp from: wendy.ucsd.edu, /pub/midi/disklavier/essl/LexiconSonate.sit.Hqx
(2) MIDI file: 5 different MIDI-files generated by *Lexikon-Sonate* can be obtained via anonymous ftp from: kahless.isca.uiowa.edu, /pub/max/lexicon/
(3) Disklavier disk: A recording of *Lexikon-Sonate* as a Disklavier disk can be found at: http://crca-www.ucsd.edu/bobw/disk3.html
(4) Audio on CD: An excerpt of the premiere of *Lexikon-Sonate* (featuring the „Bösendorfer SE Grand Piano") was released on the CD „Karlheinz Essl: Rudiments" (1995). It can be ordered from my publisher: TONOS Musikverlags GmbH, Ahastr. 9, D-64285 Darmstadt / Germany / Europe, Tel: +49–6151–31 23 47,Fax: +49–6151–31 32 78.

Reconhecimento de timbres musicais através da rede neural auto-organizável de Kohonen

MARCELO JARA PÉREZ[1], JOSE EDUARDO FORNARI[1], FURIO DAMIANI[1,2]

[(1)]*Departamento de Semicondutores, Instrumentos e Fotônica, Faculdade de Engenharia Elétrica-UNICAMP*
[(2)]*Núcleo Interdisciplinar de Comunicação Sonora-UNICAMP*

DSIF/FEE/Unicamp, 13081-970, Campinas, SP, Brasil.
e-mail: Jara@dsif.fee.unicamp.br

Resumo

Foi realizada a simulação de uma rede neural para a discriminação das diferenças timbrísticas de tons musicais. O método consiste em treinar uma rede neural auto-organizável de Kohonen com uma sequência de 17 amostras de instrumentos orquestrais. No final da fase de treinamento formam-se mapas auto-organizados onde ocorrem agrupamentos das amostras por família instrumental. Verificou-se a capacidade de reconhecimento utilizando-se todas as amostras. O sucesso do reconhecimento e a classificação do timbre dos instrumentos está diretamente relacionada à geração de mapas cuja qualidade é fortemente dependente das propriedades de convergência e da estabilidade do modelo. Esta rede neural é adequada para o reconhecimento de padrões timbrísticos com pequena taxa de erro.

Introdução

Diversos trabalhos nas áreas de psico-linguística, acústica fisiológica e psico-acústica tem abordado a discriminação de timbres (Plomp,1976; Grey & Moorer, 1977; Singh,1987). A sua percepção pelo sistema auditivo humano é um fenómeno complexo, que envolve grande capacidade de processamento para ser analisado e classificado no cérebro, de acordo com regras não sempre bem compreendidas. O reconhecimento do timbre musical depende de uma série de condições, tais como o contexto em que o sinal é percebido (Grey, 1978), sua complexidade, a amplitude e a forma como os harmônicos estão distribuídos no espectro de frequência. A forma do ataque do sinal e a variação do espectro de energia nos instantes iniciais são fundamentais na percepção (Gordon, 1987). Neste trabalho avaliamos a capacidade da rede neural de Kohonen (1982) de reconhecer e classificar timbres sonoros de instrumentos musicais, tocados isoladamente. O modelo de Kohonen foi utilizado com sucesso no reconhecimento de fonemas na língua finlandesa e na geração automática destes fonemas num computador, em tempo real (Kohonen, 1987).

A rede neural recorrente simples (SRNN) foi já utilizada para o reconhecimento de tons dos fonemas da língua Mandarin (Wang & Chen, 1994), reconhecendo variantes de tons de uma mesma estrutura fonética. Existem poucas pesquisas disponíveis na literatura sobre o reconhecimento de características timbrais usando redes neurais. A características de auto-organização e classificação de sinais sensoriais dos Mapas de Kohonen (1990), determinaram a escolha deste modelo, pois possibilitam o treinamento da rede sem supervisão. A seqüência de padrões de treinamento é aleatoriamente apresentada à rede. As respostas aos padrões são automaticamente mapeadas pelos neurônios. A fase de reconhecimento é feita após a elaboração auto-organizada dos mapas.

Na próxima secção descrevem-se suscintamente estudos sobre a discriminação e percepção do timbre. Na secção 2 especifica-se a arquitetura da rede neural e a forma como foi empregada no reconhecimento das amostras. Na secção 3 descreve-se a metodologia empregada nas simulações, o pré-processamento dos sinais e a forma como eles foram utilizados no treinamento da rede. Na secção 4 apresentam-se os resultados. Descreve-se também o processo utilizado no reconhecimento das amostras. As conclusões e sugestões para futura pesquisa na área são apresentadas na ultima secção.

1. Noções sobre o timbre sonoro e o seu reconhecimento natural

Nos estudos sobre a percepção e o reconhecimento do timbre sonoro tem sido abordados principalmente os aspectos da **altura** (*pitch*) e **sonoridade** (*loudness*). Plomp (1978) publicou o resultado dos seus estudos, baseados em diversas experiências e simulações computacionais que lhe permitiram gerar tons complexos com espectro de fase variável e estudar o seu efeito sobre a percepção do timbre. Propôs uma técnica chamada de *escalamento multidimensional*, que permite descrever de forma mais eficiente a natureza do timbre. Um sinal sonoro é descrito por :

$$p(t) = \Sigma^{m}_{n=1} \{\alpha_n \sin(2\pi nft + \phi_n)\} \qquad \text{(e.1.1)}$$

O timbre é, então, determinado pelo espectro das amplitudes $\alpha_1\alpha_2,...\alpha_m$ e o espectro de fase dado pelos componentes ϕ_1, ϕ_2, ..., ϕ_n dos harmônicos sucessivos. O propósito da técnica de escalamento multidimensional é obter informação de alguma estrutura ou padrão de dados e poder representa-la de forma que possa ser facilmente visualizada. No caso do timbre, esta informação é obtida a partir da derivação de uma matriz de dados, cujas entradas representam a dissemelhança em timbre de um par de estímulos sonoros. Utiliza-se um espaço, em que as distancias entre pontos representam índices de dissemelhança. Plomp chegou à conclusão que, embora os componentes da fase de um tom influenciam de algum modo no reconhecimento do timbre, a fase não exerce quantitativamente tanta influência quanto à distribuição dos harmônicos no espectro e as suas amplitudes. Nas experiências, ouvintes comparavam as dissemelhanças entre uma série de estímulos sonoros de mesma sonoridade e altura. Não se constatou uma correlação significativa entre as diversas formas de onda de fase aleatória e a sua dissemelhança no timbre, concluindo que o efeito da fase na percepção natural de diferenças timbrísticas é relativamente pequeno.

Plomp procurou uma relação entre o timbre e o espectro de frequência. Foram utilizados 17 instrumentos diferentes que tocaram a mesma nota, com frequência de 349 Hz, para um auditorio treinado musicalmente. O espectro foi analizado por um conjunto de filtros de larguras de faixa de 1/3 de oitava, obtendo-se assim uma representação 15-dimensional. A dissemelhança entre dois timbres foi relacionada à diferença de magnitude entre seus harmônicos.

Outros aspectos, como a relação entre o timbre e a altura, efeitos espúrios como a reverberação e a interação entre harmônicos vizinhos influem no reconhecimento do timbre. Plomp concluiu que, na percepção de dissemelhanças de timbres sonoros, a fase tem alguma influência nas baixas frequências. O efeito da fase no timbre depende da relação entre as componentes seno e coseno, diminuindo à medida em que aumenta a frequência fundamental; julga-se que as diferenças na envoltória das formas de onda sejam a principal causa da sensibilidade à fase. O timbre depende fortemente da envoltória do espectro, mais do que da frequência fundamental.

Outros trabalhos têm pesquisado a influência do contexto musical na discriminação do timbre. Grey (Grey, 1978) demonstrou que esses efeitos variam dependendo dos instrumentos utilizados. Fez estudos comparativos utilizando ouvintes que julgavam se o timbre variava ou permanecia inalterado durante uma sessão. Comparou-se a discriminação do timbre de tons isolados em relação a diversos contextos musicais, utilizando até 12 contextos diferentes: 4 tons isolados, 4 padrões musicais monofônicos e 4 padrões musicais polifônicos. Grey fez experiências com três instrumentos distintos: clarinete, trompete e fagote. Os seus resultados mostraram que os padrões musicais parecem destacar as diferenças espectrais existentes entre as versões de um timbre, enquanto os tons isolados parecem permitir uma melhor comparação dos detalhes temporais. A discriminação é prejudicada quando são utilizados padrões polifônicos, Grey especula que a causa disto pode ser devida às complicadas formas de interação na distribuição do espectro da energia acústica e/ou a efeitos de mascaramento em algumas vozes por outras mais fortes. Os resultados do seu estudo variam, de acordo com o instrumentos musical utilizado.

Um aspecto relevante na percepção do timbre é a forma como a energia acústica varia no tempo e a variação associada do espectro. Grey e Gordon(1987) mostraram que modificações espectrais afetam a percepção do timbre. Em gravações digitais de 16 instrumentos musicais, fizeram diversas transformações, alterando as características da envoltória do espectro original. Geraram novos timbres usando síntese aditiva digital, adicionando à gravação original um conjunto de harmônicos modulados no tempo amplitude e frequência. Das16 amostras, 8 foram modificados em 4 pares, de forma que a envoltória do espectro de uma, amostra modificada fosse correspondente

à de seu par associado. Essas modificações da envoltória demonstraram ter forte influência na percepção do timbre sonoro. Outro trabalho (Gordon, 1987) mediu e modelou o ataque de instrumentos musicais, definindo o PAT (*Perceptual Attack Time*) para medir as características e a precisão rítmicas de uma execução musical.

2. O modelo de rede neural auto-organizável de Kohonen

O modelo e o algoritmo propostos originalmente por Kohonen (1982), conhecido como Mapa Auto-Organizado (*Self-Organizing Feature Map*), tem a propriedade de gerar representações internas das características dos sinais que são apresentados à sua entrada durante a fase de aprendizado. Os mapas têm sido utilizados como filtros adaptativos no reconhecimento de voz e imagens, no: controle de processos, robótica e representação de estruturas complexas de dados. Uma de suas características é preservar internamente a ordem topológica dos sinais de entrada. O aprendizado da rede é não-supervisionado, mas Kohonen (1990) apresentou variantes de sua proposta

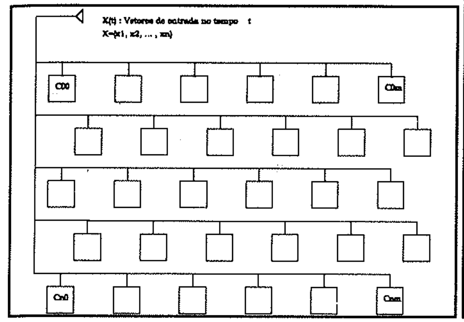


fig.2.1. Arranjo com 30 células. $X_i(t)$ é o vetor de entrada.

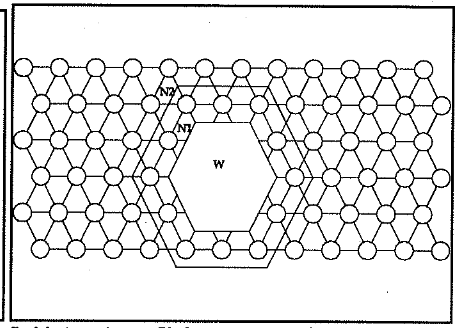


fig.2.2 Arranjo com 72 elementos.; W - célula vencedora.

original, inserindo um processo supervisionado de ajuste fino do mapa, o algoritmo LVQ (*Learning Vector Quantization*).

Na fase de aprendizado, as células da rede são automatica e gradualmente *sintonizadas* aos sinais de entrada apresentados seqüencialmente, formando internamente aglomerados de padrões de classes semelhantes. Esta sintonia é quantificada pela mudança de valor dos pesos de cada célula. Na fig.2.1, vemos que cada célula C_{ij}, possui um peso, representado por um vetor n-dimensional $\mathbf{W}=\{w_1,w_2,...,w_n)$, cujos componentes variam à medida que uma sequência de sinais, representada por vetores $\mathbf{X}(t)=\{x_1,x_2,...,x_n\}$, é apresentada na sua entrada. Cada vetor de entrada é apresentado simultaneamente à toda a rede. A cada iteração (apresentação de um vetor), são modificados os pesos da célula vencedora e os das células pertencentes a sua vizinhança, de acordo com o algoritmo descrito a seguir. Na fig. 2.2, vemos a forma como as células componentes da rede são interconectadas, formando uma vizinhança de tipo hexagonal, onde cada uma das células (exceto as das bordas) possui seis vizinhos. As seis células em torno da vencedora formam a vizinhança de raio 1, N_1, as outras 12 pertencem a vizinhança de raio 2, as 18 células formam a vizinhanca total $\mathbf{N}_2$; W, a vencedora, é a célula central da vizinhança hexagonal. Uma vizinhança pode ter forma retangular.

2.1 Algoritmo de operação.

Um vetor de entrada $\mathbf{X}=\{x_1, x_2, ..., x_n\}^T$ Î $\mathbf{R}^n$, é fornecido no instante *t* a todas as células da rede. O vetor de pesos de uma célula é $\mathbf{W}=\{w_1,w_2,..., w^n\}^T$ Î $\mathbf{R}^n$. O algoritmo de aprendizado muda os pesos de conexão de cada uma das células, em função dos vetores de entrada. Inicialmente os valores dos pesos w_i de cada célula são inicializados com valores aleatórios. A geração do mapa auto-organizado é feita utilizando-se um número suficiente de vetores de entrada distribuídos estatisticamente no espaço de entrada, pelo algoritmo seguinte:

Passo 1: Achar a unidade (célula) **c**, cujo vetor de pesos **W**(t) tenha a mínima distância da entrada **X**(t):

$$d_c(\mathbf{X}) = \| \mathbf{X}(t) - \mathbf{W}_c(t) \| = \min_i \{\| \mathbf{X}(t) - \mathbf{W}_i(t)\|\} \qquad \text{(e.2.1)}$$

Neste caso, diz-se que a unidade **c** responde a **X**(t). No caso mais simples, a distância d_c(**X**), entre **X** e **W**, pode ser a *Distância Euclidiana*.

Passo 2: Modificar os vetores de peso da unidade ganhadora **c** e dos seus vizinhos topológicos. A vizinhança topológica N_c, tal como a ilustrada na fig.2.2, pode assumir formas variadas (por exemplo: hexagonal ou retangular). No processo de aprendizado, N_c diminui continuamente ao longo do tempo.

O raio de N_c, é inicializado com um valor amplo (p/ ex., 75% do raio total da rede). O raio vai diminuindo, de acordo com uma função, que pode ser linear ou não-linear. Os pesos são atualizados de acordo com as expressões seguintes:

$$m_i(t+1) = m_i(t) + \alpha(t)[x(t) - m_i(t)] \; \forall \; i \in N_c \qquad \text{(e.2.2)}$$

$$m_i(t) = m_i(t) \; \forall \; i \notin N_c \qquad \text{(e.2.3)}$$

Na expressão (e.2.2), a função a(t), é real e positiva, variando entre os valores ($0 < \alpha(t) < 1$). A função $\alpha(t)$ é conhecida como o *ganho* e pode ir decrescendo de forma linear ou não-linear até chegar a valores próximos a zero.

2.2. Mapas auto-organizados de características topológicas.

O algoritmo descrito acima é a base da operação da rede para utilizá-la como um sistema de reconhecimento de padrões. O componente x_j do vetor de entrada pode ser considerado como a atividade de um neurônio *j*, numa camada sensorial de entrada, considerando-se o modelo de Kohonen com 2 camadas: uma sensorial (as entradas) e uma de mapeamento bi-dimensional (as células de processamento) . Uma célula **c** possui uma resposta forte a um sinal **X**, quando d_c(**X**) é pequena. Em consequência, o vetor $\mathbf{W}_c$ aponta diretamente para àquela posição no espaço de entrada *n*-dimensional, na qual a célula **c** está melhor *sintonizada*. Alguns autores chamam o $\mathbf{W}_c$ a *posição virtual* da célula **c** no espaço de características de dimensão *n*. O processo de aprendizado dos vetores $\mathbf{W}_c$, representa a evolução do mapa em intervalos de tempo discreto $t = 0, 1, ..., t_{max}$. Como resultado deste processo, os pesos evoluem até abrangerem um espaço característico virtual, de dimensão *n*, constituindo um **mapa auto-organizado**.

3. Metodologia e procedimento experimental

Neste trabalho utilizou-se a arquitetura descrita nas fig.2.1 e 2.2 num arranjo de 12 x 8 células, onde se aplicou o algoritmo apresentado na secção 2.1. A rede foi estimulada por uma sequência de vetores de dimensão 15 e 16, representando o sinal sonoro. Os estímulos foram gerados a partir de amostras de instrumentos acústicos gerados por um sintetizador ROLAND D-20. Os 17 arquivos que representam os timbres escolhidos abrangeram uma ampla gama de instrumentos musicais, pertencentes às famílias ou classes das cordas, madeiras, metais e percussão. Os timbres foram rotulados de acordo com a seguinte convenção: **A**=Apito, **B**=Bumbo, C=Contrabaixo, **D**=Corne Inglês, **E**=Violoncello, **F**=Clarinete, **G**=Prato Orquestral, **H**=Caixa, **I**=Flauta, J=Trompa, **K**=Oboe, **L**=Pizzicato de violino, **M**=Queixada, N=Trompete, **O**=Trombone, **P**=Tuba, **Q**=Violino.

3.1 Estímulos

Os sons foram amostrados numa *workstation*, no padrão U-LAW (8 bits; fs=8 Khz), com duração de cerca de 500 mseg. Utilizando o *software* MATLAB 4.2 para UNIX, foi desenvolvido um programa que recebe a amostra com a duração indicada e a divide em 3 trechos, correspondentes ao ataque (**A**), sustentação (**S**) e decaimento (**D**) do sinal. Para cada trecho, calculou-se o espectro discreto, através da FFT. A função *spectrum* do MATLAB implementa o método Welch de estimativa de Espectro de Potência (Oppenheim & Schafer, 1975). Utilizando esta função foram gerados vetores de 16 componentes, representando o espectro de potência de cada um dos trechos A, S e D. Nas figs.3.1-fig.3.4, a imagem superior representa a forma de onda do sinal no tempo. **A**, **S** e **D** indicam o trecho em que foi calculado o espectro, para cada amostra instrumental. A figura 3-d inferior representa a evolução do espectro de energia do sinal, ao longo do tempo, durante o intervalo de 500 ms.

3.2. Formato dos padrões de entrada

Os vetores correspondentes aos 3 trechos amostrados foram concatenados num só vetor de 48 componentes (3 x 16), deslocando cada sub-vetor à esquerda (ver fig. 3.5). Um método similar, aplicado a redes do tipo TDNN

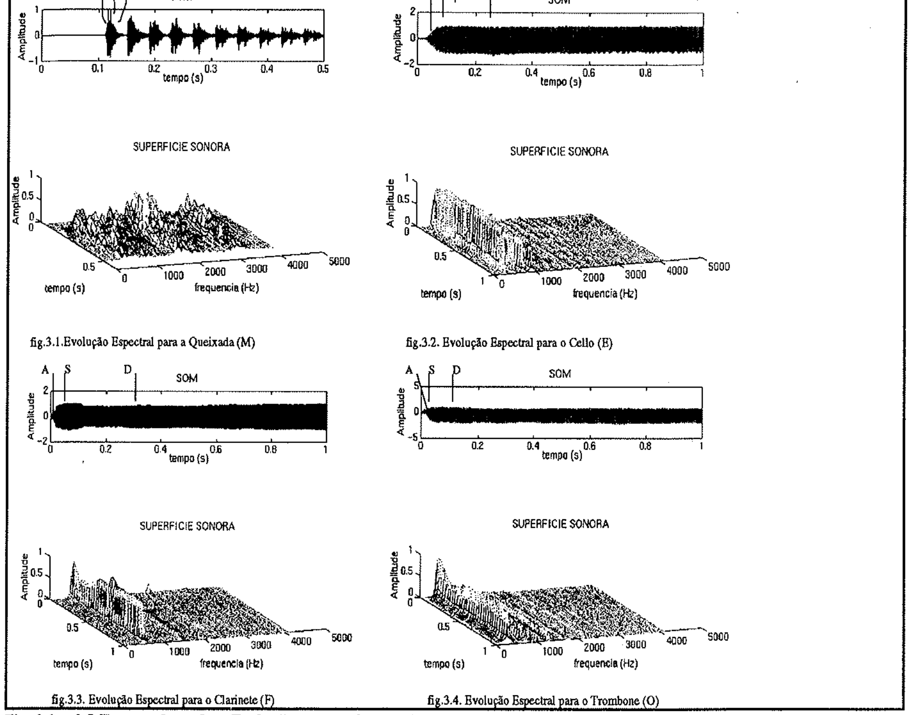


fig.3.1.Evolução Espectral para a Queixada (M)
fig.3.2. Evolução Espectral para o Cello (E)
fig.3.3. Evolução Espectral para o Clarinete (F)
fig.3.4. Evolução Espectral para o Trombone (O)

Fig. 3.1 a 3.5 Formas de onda e Evolução espectral para alguns instrumentos.

(*Time-Delay Neural Network*) foi utilizado por J. Kangas (Kangas, 1994) para acrescentar informação de contexto em problemas de reconhecimento de fonemas. A estrutura dos vetores de entrada, **X**(t), que serão os padrões definitivos de ensino da rede, é apresentada na fig.3.5.

A rede é treinada por uma sequência de vetores de padrões de timbre com a estrutura mostrada na fig.3.5.

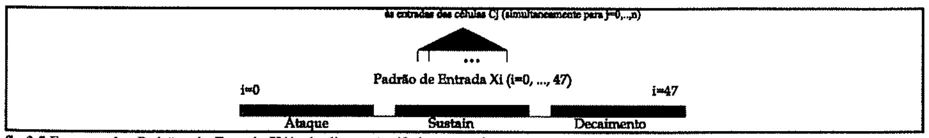


fig.3.5. Formato dos Padrões de Entrada **X**(t), de dimensão 48, à uma rede composta por *n* neurônios

O número destes padrões deve ser grande, para ter uma boa representação do espaço amostral. Nem sempre é possível alcançar esse número de vetores de entrada, quando os padrões são obtidos a partir de dados experimentais. Neste caso, o conjunto desses vetores deve ser reapresentado obter mapas com as características desejadas.

4. Mapas auto-organizados resultantes

A classificação das famílias de instrumentos, utilizada neste trabalho obedece à seguinte convenção :

Classe A - Cordas : violino, violoncello, contrabaixo, pizzicato de violino.
Classe B - Madeiras : oboe, flauta, clarinete, corne inglês
Classe C - Metais : tuba, trombone, trompete, trompa.
Classe D - Percussão : queixada, bumbo, caixa, prato orquestral, apito.

O mapa da fig.4.1 foi obtido após 81600 passos de treinamento. Vê-se a formação de regiões, correspondentes às diferentes classes ou famílias instrumentais. Cada unidade do mapa representa uma célula de uma rede de 12 x 8 neurônios. Cada célula, ou neurônio, ficou sintonizado num determinado padrão.

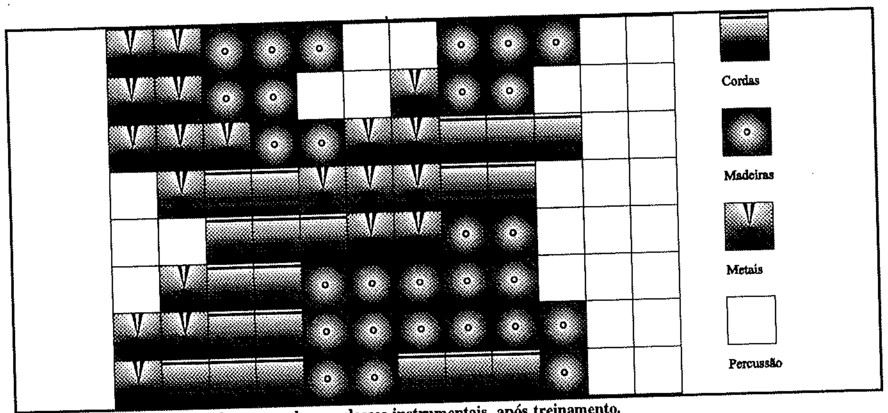


fig.4.1. Mapa Auto-Organizado, agrupado por classes instrumentais, após treinamento.

Numa segunda fase, após a formação do mapa, apresentou-se à rede uma série de 68 vetores de entrada, representando uma sequência de vetores *desconhecidos*, obtidos de forma similar ao procedimento descrito no item 3.1. O modelo conseguiu reconhecer esses padrões *desconhecidos*, com uma taxa de erro de aproximadamente 7%, utilizando-se os mesmos parâmetros do mapa da fig.4.1. Os padrões de entrada desconhecidos representam pequenas variantes dos vetores de treinamento que foram inicialmente apresentados à rede. Uma série de mapas similares ao apresentado na fig. 4.1, foram obtidos em cada simulação. A evolução no tempo e a forma final do mapa (após o treinamento) é fortemente dependente dos parâmetros utilizados no algoritmo descrito na secção 2.1. A forma como a função $\alpha(t)$ evolui no tempo exerce una enorme influência na formação final do mapa. A rapidez com que a vinzinhança N_c diminui, a medida que o tempo evolui na fase de treinamento, também demonstrou ser fundamental. Os mapas com características topológicas pobres possuem muitas regiões inconexas, ao contrário daquele mostrado na fig.4.1.

5.Conclusões

Nas redes similares às estudadas neste trabalho, o sucesso do reconhecimento de um padrão, previamente ensinado, está associado à geração de mapas auto-organizados com características topológicas bem definidas. Nas nossas simulações obtiveram-se resultados muito variáveis; desde taxas de erro razoavelmente boas (da ordem do 4%) até resultados bastante pobres (taxas de erro superiores a 70%). Estes resultados são dependentes de muitos fatores. Primeiramente, do número de passos de treinamento (ou aprendizado). O processo é melhor comportado enquanto se tiver um número de passos de treinamento superiores a 1000. Kohonen utilizou, em seus trabalhos, um número de passos da ordem de 10000, para conseguir bons resultados. Outro fator são os parâmetros que controlam o comportamento da função $\alpha(t)$ e a sua evolução no tempo. Ao contrario do sugerido em algumas publicações, a escolha de uma função (linear ou não-linear) exerce grande influencia nos resultados, como foi comprovado nas nossas simulações. Do mesmo modo, a forma como a vizinhança da célula vencedora evolui no

tempo de treinamento também é muito importante. Para o efeito do reconhecimento de timbres sonoros, é fundamental um adequado pre-processamento dos sinais. Apesar de se tentar introduzir informação relativa às características transientes do sinal acústico, utilizando vetores concatenados, o uso só de informação do espectro de energia parece ser insuficiente. A informação da fase como informação adicional seria um fator interessante a estudar, embora, como foi mencionado, para o caso da percepção humana de diferenças timbrísticas não seja tão importante. No relativo ao estudo de novas técnicas de reconhecimento de sinais acústicos, utilizando sistemas neurais artificiais, existem muitas possibilidades. Um aspecto particularmente interessante é o estudo da influência do contexto musical e da utilização de tons polifônicos, em lugar de tons isolados.

No final deste trabalho, verificou-se a capacidade de reconhecimento de padrões timbrísticos pelo modelo de Kohonen, e identificaram-se alguns importantes aspectos que permitem diminuir a taxa de erro no reconhecimento e melhorar a formação de mapas auto-organizados de características topológicas. Outra característica da arquitetura do modelo é ser especialmente adequada para implementação em sistemas VLSI (*Very Large Scale Integration*), o que torna o modelo particularmente interessante para aplicações em tempo real.

Bibliografia

Erwin ,E., Obermayer ,K. & Schulten, K. (1992). Self-Organizing Maps: stationary states, metastability and convergence rate. *Biol. Cybernetics*, 67, 35-45.

Gordon, J.(1987).The perceptual attack time of musical tones. *Journal of the Acoustical Society of America* (82)1, 88-105.

Grey, J.(1978). Timbre discrimination in musical patterns. *Journal of the Acoustical Society of America*, 64(2), 467-472.

Grey, J. & Gordon, J.(1978). Perceptual effects of spectral modifications on musical timbres. *Journal of the Acoustical Society of America*, 63(5), 1493-1500.

Grey & Moorer (1977). Perceptual evaluations of synthesized musical instrument tones. *Journal of the Acoustical Society of America*, 62(2), 454-462.

Kangas,J. (1994). *On the Analysis of Pattern Sequences by Self-Organizing Maps*, Phd dissertation, Helsinki. Univ.

Kohonen, T. (1982).Self-organizing formation of topologically correct feature maps.*Biol. Cybernetics* 43,59-69.

Kohonen (1988). The Neural Phonetic Typewriter. *IEEE Computer magazine*, 21(3), 11-22.

Ritter , H. & Kohonen, T.(1989). Self-Organizing Maps, *Biological Cybernetics*, 61, 241-254.

Kohonen, T.(1990).The Self-Organizing Map, *Proceedings of the IEEE*, 78(9),1464-1480.

Lo, Z. & Bavarian, B. (1991). On the rate of convergence in topology preserving neural networks. *Biological Cybernetics*, 65, 55-63.

Plomp, R. (1976). *Aspects of Tone Sensation - A Psychophysical Study.* Academic Press, London.

Ritter & Schulten, 1986.

Singh, P. (1987). Perceptual organization of complex-tone sequences: A tradeoff between pitch and timbre? *Journal of the Acoustical Society of America*, 82(3), 886-899.

Oppenhein, A. & Schafer, R.(1975). *Digital Signal Processing*, Prentice-Hall.

Wang,Y. & Chen,S. (1994).Tone recognition of continuous Mandarin speech assisted with prosodic information. *Journal of the Acoustical Society of America*, 96(5), 2637-2645.

A neural model to segment musical pieces

Otávio Augusto Salgado Carpinteiro
School of Cognitive and Computing Sciences
University of Sussex
Falmer, East Sussex, BN1 9QH, England
otavioc@cogs.sussex.ac.uk

Abstract

This paper proposes a knowledge representation of rhythmic patterns, and a neural model to segment musical pieces in accordance with three cases of rhythmic segmentation. The neural model has a topology which is identical to that of NETtalk (Sejnowski & Rosenberg, 1987). It is trained on sets of contrived patterns, and evaluated on two two-part inventions, two three-part inventions, and two fugues of Bach (Bach, 1970, 1989).

1 Introduction

Due to memory constraints, it is believed that listeners do not grasp a musical piece in its entirety, but on the contrary, they segment it into parts which can be analysed, and then later related to each other (Drake & Palmer, 1993). Studies of segmentation of nonmusical sound sequences (Gabrielsson, 1973; Garner & Gottwald, 1968) as well as of musical sequences (Drake & Palmer, 1993; Lerdahl & Jackendoff, 1983; Kirkpatrick, 1984) suggest that the Gestalt principles of proximity and similarity may be the basis on which listeners segment music. Based on such principles, several researchers have proposed three cases of rhythmic segmentation: longer durations (Drake & Palmer, 1993; Lerdahl & Jackendoff, 1983), pauses (Drake & Palmer, 1993; Lerdahl & Jackendoff, 1983), and breaks of similarity (Lerdahl & Jackendoff, 1983; Kirkpatrick, 1984).

A neural model is proposed here to segment musical pieces in accordance with the three cases of segmentation. The topology of the model is identical to that of NETtalk (Sejnowski & Rosenberg, 1987). In the next sections, details are given of the three cases of segmentation. A novel knowledge representation for rhythmic sequences is introduced, along with details of the model. Finally, results of three experiments are presented — one for each case of segmentation.

2 Three cases of rhythmic segmentation

We have considered as an example of *longer durations* the case when given four notes $n_1n_2n_3n_4$, the duration of n_2 is greater than the durations of n_1 and n_3. Again, given four notes $n_1n_2n_3n_4$, we have considered as an example of segmentation given by a *pause* when there is at least one pause between n_2 and n_3. Finally, we have considered as an example of *breaks of similarity* the case when given eight notes $n_1n_2n_3n_4n_5n_6n_7n_8$, the durations of n_1, n_2, n_3, n_4 are identical, the durations of n_5, n_6, n_7, n_8 are also identical, and the durations of the first four notes are different from the last four. Figure 1 illustrates the three cases.

3 Knowledge representation for rhythmic sequences

If we set the small figure in a musical sequence (or in a whole piece) to be the *time interval (TI)*, all other figures become multiples of TI. For example, if TI is an eighth note, a quarter note lasts two TIs, a

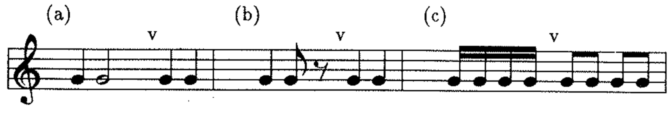


Figure 1: (a) longer durations; (b) pauses; (c) breaks of similarity;

half note lasts four TIs, and so on. We may also define a counter, *time interval counter (TIC)*. One TIC lasts one TI. The TIC is the unit in which the musical sequence is measured. Therefore, at each TIC, either there is a pause, or a note is played, or a note is sounded. Figure 2 illustrates the representation. It shows a sequence which lasts nine TICs, and whose TI is an eighth note. In the experiments, each of the three events was represented by a pair of neural input units. A pause was represented by (00). Note sounded was represented by (10), and note played by (11).

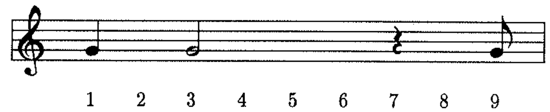


Figure 2: A musical sequence lasting nine TICs

4 The model

The model presented here in brief was developed by Sejnowski and Rosenberg (1987). It is shown in figure 3. The input layer holds a number of pairs of units which make up a window. Each pair represents one of the three events mentioned above (note sounded, note played, pause). The activations of the pairs of units in the window represent a rhythmic pattern. For instance, let TI be an eighth note, and the size of the window be nine pairs of units. Figure 4 would then represent the example in figure 2.

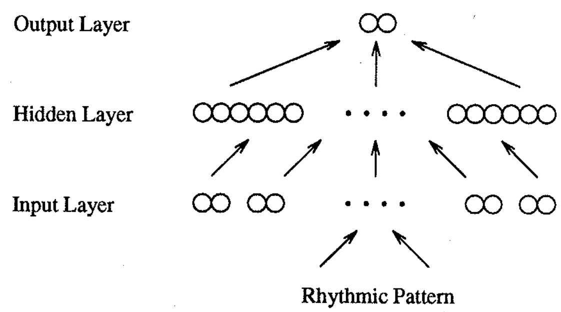


Figure 3: The model

The activation a_i of each hidden unit i is given by the sigmoid function

$$a_i = \frac{1}{1+e^{-net_i}} \tag{1}$$

net_i is given by

11 10 11 10 10 10 00 00 11

Figure 4: Representation for the sequence in figure 2

$$net_i = \sum_j w_{ij} a_j + bias_i \qquad (2)$$

where w_{ij} is the weight from input unit j to hidden unit i, a_j is the activation of input unit j, and $bias_i$ is a special weight which adjusts the values of net_i to make an efficient use of the treshold of the sigmoid.

The activation a_i of each output unit i is given by

$$a_i \doteq net_i = \sum_j w_{ij} a_j + bias_i \qquad (3)$$

where w_{ij} is the weight from hidden unit j to output unit i, a_j is the activation of hidden unit j, and $bias_i$ is again a special weight[1].

The weights are updated according to the generalized delta rule (Rumelhart, Hinton, & McClelland, 1986),

$$\Delta w_{ij}(p) = \alpha \delta_i a_j + \beta \Delta w_{ij}(p-1) \qquad (4)$$

where $\alpha, \beta \in (0,1)$ are the learning rate and momentum respectively. The subscript p indexes the pattern number, and the learning takes place on a pattern-by-pattern basis. Both the learning rate and momentum are modified at the end of each epoch . The learning rate is reduced when the total error increases, and increased when the error decreases. The momentum is disabled until the end of training if the total error increases. The error signal δ_i, for an output unit i is given by

$$\delta_i = t_i - a_i \qquad (5)$$

where t_i is the desired activation value and a_i is the activation obtained. For a hidden unit i, δ_i is given by

$$\delta_i = a_i(1 - a_i) \sum_k \delta_k w_{ki} \qquad (6)$$

where w_{ki} is the weight from hidden unit i to output unit k.

We have used two output units in all experiments. We have trained the model to display activation values (10) in these units when the window in the input layer is representing a *negative pattern*, that means, a rhythmic pattern which is not a case of segmentation. We have also trained it to display values (01) when the window is representing a *positive pattern*, a rhythmic pattern which is a case of segmentation.

5 First experiment

The first experiment was on recognizing cases of segmentation given by pauses. We have randomly generated three sets of patterns, each set containing 2000 patterns. The first set was the training set. Every 20 epochs, training was halted, and the model was tested on the second set. When the total error stopped decreasing, training was ended, and the model was tested on the third set. We could thus evaluate different net configurations to find the optimum number of hidden units.

The best performance was given by a configuration with 20 hidden units. The window in the input layer held 10 pairs of units. The initial weights were set randomly, and it was trained for 80 epochs.

A fourth set containing 8 positive and 73 negative patterns was also randomly generated. Principal component analysis (PCA) was performed on the activations of the hidden units given by each pattern in the set. Figures 5 and 6 plot the two first principal components for the negative and positive patterns respectively. We can verify that there is no correlation between the negative and positive patterns, for

[1] As the output units are now linear, the existence of the bias is no longer necessary, although we have decided to keep them.

they are placed in different clusters. Therefore, the internal representations in the hidden units make a distinction between the two types of patterns.

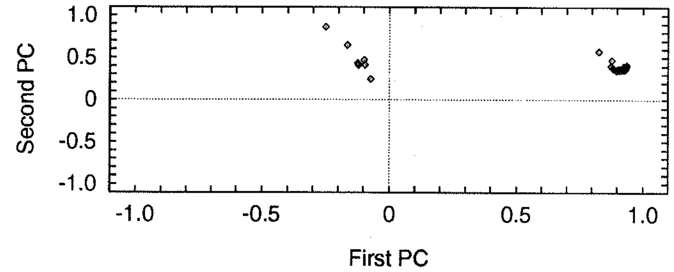


Figure 5: The two first principal components of the negative patterns in the first experiment

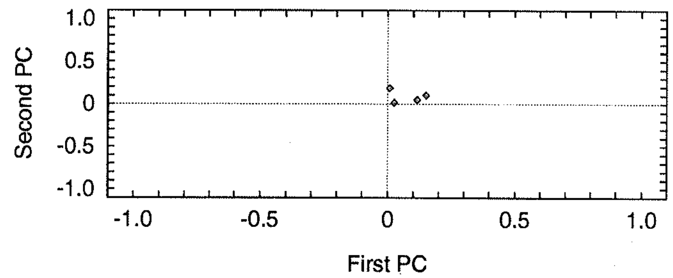


Figure 6: The two first principal components of the positive patterns in the first experiment

The model was evaluated on six musical pieces from Bach. The first two pieces were the nineth and thirteenth two-part inventions in F minor and A minor (Bach, 1970). The third and fourth were the third and fourteenth three-part inventions in D major and B flat major (Bach, 1970). The last two pieces were the fourth and seventeenth fugues in C sharp minor and A flat major of the Well-Tempered Clavier (Bach, 1989). Each part[2] of each piece was input separately. The size of the window was not wide enough to cover all instances of segmentation present in the pieces, yet it was wide enough to cover most of them. The results are displayed in table 1. The percentage of misclassifications is low. We think that these misclassifications can be reduced or even avoided by increasing the number of patterns in the training set.

6 Second experiment

The second experiment was on recognizing cases of segmentation given by longer durations. Again, we have randomly generated three sets of patterns, each set containing 3000 patterns. The training method was identical to that followed in the first experiment.

The best performance was given by a configuration with 10 hidden units. The window in the input layer held 19 pairs of units. Initial weights were set randomly, with training lasting for 240 epochs.

A fourth set containing 110 positive and 440 negative patterns was also randomly generated. Again, PCA was applied to the activations of the hidden units given by each pattern in the set. Figures 7 and 8 plot the two first principal components for the negative and positive patterns respectively. As in the first experiment, we verified that there is no correlation between the negative and positive patterns.

[2] Also known as voice.

Table 1: Results of the first experiment — #NP: number of negative patterns; #PP: number of positive patterns; %NM: percentage of misclassified negative patterns; %PM: percentage of misclassified positive patterns;

Pieces	#NP	#PP	%NM	%PM
1	814	2	0	0
2	784	16	0	0
3	1188	12	0	0
4	2293	10	0	60
5	4556	44	0	14
6	2204	36	0	0

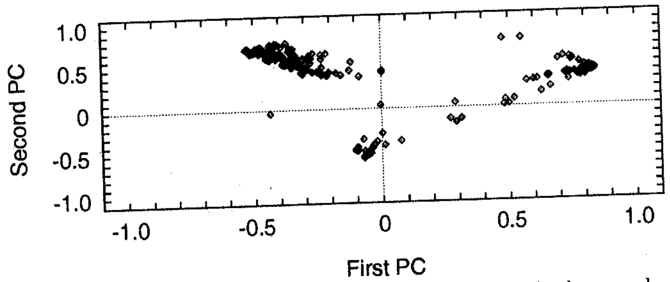


Figure 7: The two first principal components of the negative patterns in the second experiment

The model was evaluated on the same musical pieces as in the first experiment. Each part of each piece was input separately, and the window size was also wide enough to cover most instances of segmentation present in the pieces. As displayed in table 2, there is a low percentage of misclassifications. Once again, we believe that these misclassifications can be reduced by increasing the number of patterns in the training set.

7 Third experiment

The third experiment was on recognizing cases of segmentation given by breaks of similarity. Once again, we have randomly generated three sets of patterns, each set containing 3000 patterns. The training method was identical to that followed in the first two experiments.

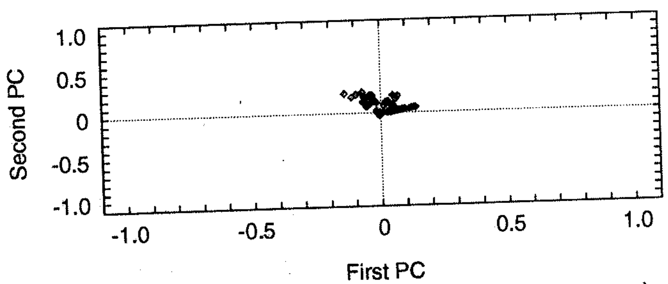


Figure 8: The two first principal components of the positive patterns in the second experiment

Table 2: Results of the second experiment — #NP: number of negative patterns; #PP: number of positive patterns; %NM: percentage of misclassified negative patterns; %PM: percentage of misclassified positive patterns;

Pieces	#NP	#PP	%NM	%PM
1	764	52	2	0
2	790	10	0	0
3	1171	25	10	0
4	2253	39	18	8
5	4483	98	26	15
6	2185	48	22	6

The best performance was given by a configuration with 5 hidden units. The window in the input layer held 16 pairs of units. Random initial weights were trained for 180 epochs.

A fourth set containing 2 positive and 287 negative patterns was also generated randomly. PCA was performed on the activations of the hidden units given by each pattern in the set. Figures 9 and 10 plot the two first principal components for the negative and positive patterns respectively. Unlike the previous experiments, the positive and negative patterns are not completely separated into two different clusters.

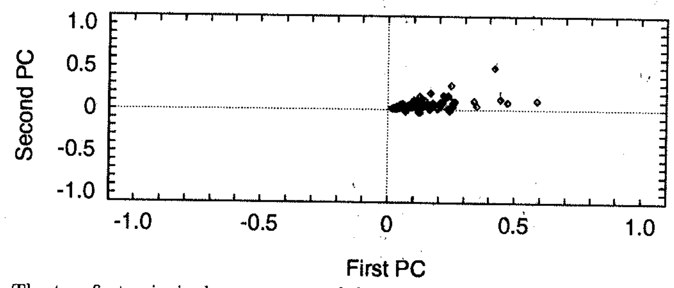


Figure 9: The two first principal components of the negative patterns in the third experiment

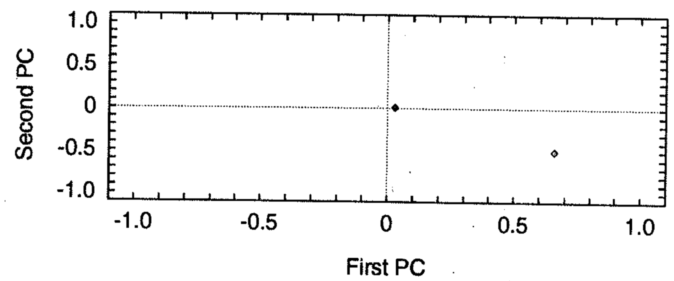


Figure 10: The two first principal components of the positive patterns in the third experiment

The model was evaluated on the same musical pieces as in the former experiments. Each part of each piece was input separately, and the window size was wide enough to cover most instances of segmentation present in the pieces. As displayed in table 3, there is a low percentage of misclassifications. As in the former experiments, we also think that these misclassifications can be reduced by increasing the number of patterns in the training set.

Table 3: Results of the third experiment — #NP: number of negative patterns; #PP: number of positive patterns; %NM: percentage of misclassified negative patterns; %PM: percentage of misclassified positive patterns;

Pieces	#NP	#PP	%NM	%PM
1	813	3	3	0
2	772	28	6	0
3	1196	4	5	0
4	2302	1	0	0
5	4589	8	2	0
6	2233	7	3	0

8 Conclusion

A neural model with an identical topology to that of NETtalk is proposed to segment musical pieces according to three cases of rhythmic segmentation. The model was successfully applied to six musical pieces from Bach. The results presented here suggest that musical segmentation can be accomplished by a neural model with supervised learning.

Acknowledgements

This research was fully supported by CAPES, Brazil.

References

Bach, J. S. (1970). *Inventionen und Sinfonien*. BWV 772–801. Bärenreiter Kassel, Basel, Germany.

Bach, J. S. (1989). *Das Wohltemperierte Klavier*. Vol. 1. BWV 846–869. Bärenreiter Kassel, Basel, Germany.

Drake, C., & Palmer, C. (1993). Accent structures in music performance. *Music Perception, 10*(3), 343–378.

Gabrielsson, A. (1973). Similarity ratings and dimension analyses of auditory rhythm patterns. *Scandinavian Journal of Psychology, 14*, 138–160.

Garner, W. R., & Gottwald, R. L. (1968). The perception and learning of temporal patterns (part 2). *Quaterly Journal of Experimental Psychology, 20*, 97–109.

Kirkpatrick, R. (1984). *Interpreting Bach's Well-Tempered Clavier, A Performer's Discourse of Method.* Yale University Press, London, UK.

Lerdahl, F., & Jackendoff, R. S. (1983). *A Generative Theory of Tonal Music*. The MIT Press, Cambridge, MA.

Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1986). A general framework for parallel distributed processing. In Rumelhart, D. E., McClelland, J. L., & the PDP Research Group (Eds.), *Parallel Distributed Processing*, Vol. 1, chap. 2, pp. 45–76. The MIT Press, Cambridge, MA.

Sejnowski, T. J., & Rosenberg, C. R. (1987). Parallel networks that learn to pronounce english text. *Complex Systems, 1*, 145–168.

Um Modelo Inteligente para Classificação Harmônica Tonal

FÁBIO GHIGNATTI BECKENKAMP*

PAULO MARTINS ENGEL**

CPGCC, INSTITUTO DE INFORMÁTICA, UFRGS

Caixa Postal 15064

91501-970 Porto Alegre - RS- Brazil

ABSTRACT

This work presents an artificial intelligence solution for the harmonic classification problem. The proposed artificial intelligence model divides the harmonic classification problem in subproblems. Intelligent solutions are indicated for each subproblem. The subproblem's solutions interact into the model in the way to find the solution for the harmonic classification problem. The subproblems found are: chord identification, chord classification, chord inversion classification, music tonality classification and harmonic degree's classification. The model indicates connectionist solutions for the chord and tonality classification subproblems, and indicates symbolic solutions for the chord inversions and harmonic degree's classification subproblems. The chord identification problem is partially solved by an algorithm solution. The model was implemented in an appropriately software and hardware that allowed connectionist and symbolic solutions, and the utilization of MIDI interface as music source. The model validation was performed using musical parts from great erudite composers. The model performed an acceptable classification of these music parts showing that cognitive musical problems can be solved by Artificial Intelligence solutions.

1 Introdução

A classificação harmônica consiste em gerar a descrição de uma estrutura audível, formada por um conjunto de tons e suas relações melódicas, rítmicas e métricas que evidenciam uma estrutura de estilo específico. A relação entre acordes e tonalidade é bastante estreita pois é a tonalidade que faz a sonoridade de um acorde com uma função, e ao mesmo tempo, são os acordes, com suas seqüências e relações, que criam a tonalidade.

Dado um conjunto de notas e uma tonalidade específica, pode-se analisar harmonicamente estas notas. Por exemplo: um acorde formado pelas notas dó, mi e sol, possui o grau harmônico I na tonalidade dó maior e possui o grau harmônico V na tonalidade de fá maior. Este é um problema de classificação e a inteligência artificial oferece ferramentas que permitem resolver este tipo de problema como o conexionismo e a abordagem simbólica.

* Student of Master Degree in Computer Science at Universidade Federal do Rio Grande do Sul - UFRGS, E-mail: fgb@inf.ufrgs.br

** Professor at Universidade Federal do Rio Grande do Sul - UFRGS E-mail: engel@inf.ufrgs.br

2 Objetivos

Este trabalho tem como objetivo principal apresentar a solução encontrada para construir o modelo de inteligência artificial proposto em (Beckenkamp & Engel, 1995). Naquele trabalho, o problema da classificação harmônica é apresentado e a teoria envolvida é detalhada. São apresentadas, ainda, possíveis soluções que possibilitariam a construção do modelo através do uso da inteligência artificial.

O modelo construído deve explorar as técnicas da inteligência artificial, envolvendo dois paradigmas da inteligência artificial: o simbólico e o conexionista. A solução da classificação harmônica foi buscada preferencialmente em sistemas conexionistas com o objetivo de experimentar e validar o uso desta ferramenta no domínio da cognição musical. Porém, isto somente foi feito quando a solução via conexionismo mostrou-se adequada ao problema a ser resolvido.

3 O Modelo Proposto e suas Soluções

O modelo de (Beckenkamp & Engel, 1995) propunha a busca da reconstrução do sistema cognitivo envolvido na classificação harmônica através da decomposição desse problema em subproblemas. Os subproblemas foram modelados de forma que os juntando-se possa reconstruir a inteligência do sistema.

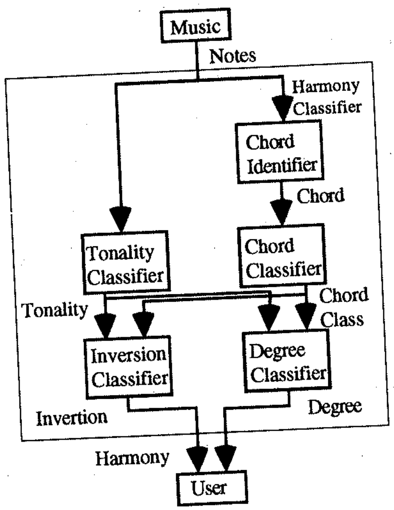


Figura 3.1 Classificação Harmônica

A figura 3.1 acima mostra o modelo proposto para solução do problema da classificação harmônica. A figura mostra o problema, seus subproblemas, bem como o fluxo das informações. Os primeiros subproblemas a serem resolvidos são a identificação e classificação dos acordes. Os acordes devem ser classificados de acordo com sua estrutura intervalar e sua inversão. O modelo deve sugerir uma forma de classificar a tonalidade da música nas 12 tonalidades maiores e nas 12 tonalidades menores o que é fundamental para a classificação dos acordes em graus. Por fim, um classificador de graus deve ser proposto.

No modelo da figura 3.1 acima, a música é fornecida por um usuário, o que pode ser

realizado através de um teclado MIDI ou até mesmo através de um seqüenciador. A música poderá ser fornecida em tempo real ou não, de acordo com a capacidade computacional do ambiente onde for implementado o sistema. As notas da música sendo tocada chegam no classificador de harmonia e são diretamente repassadas ao identificador de acordes e ao classificador de tonalidade. O classificador de tonalidades verifica a tonalidade da música através das notas tocadas. Em qualquer intante de tempo durante a música o classificador deverá ser capaz de fornecer a informação da tonalidade por ele identificada até aquele momento da música. As notas que chegam ao classificador de harmonia também são repassadas diretamente ao identificador de acordes que verifica que conjunto de notas na música formam um acorde. O acorde identificado é passado ao classifcador de acordes que fornece em sua saída a classificação encontrada. O classificador de inversões utiliza-se do acorde classificado e da tonalidade da música para classificar as inversões dos acordes. Por fim, o classificador de graus faz uso da tonalidade e dos acordes para classificar em termos de graus harmônicos os acordes anteriormente classificados por tipo no classificador de acordes. Estas informações poderão, então, ser utilizadas pelo usuário.

Existe na literatura alguns trabalhos que buscam solucões para o problema da classificação harmônica e para os subproblemas anteriormente relacionados. Christoph Lischka apresenta um modelo para classificação harmônica funcional através de Máquina de Boltzmann em (De Poli et al, 1991). Uma solução utilizando uma rede neural não-supervisionada (Kohonen Feature Maps - KFM), para a tarefa de classificação de acordes é apresentada por Barucha em (Todd & Loy, 1991). John Maxwell em (Balaban et al, 1992) busca a solução do problema da classificação de acordes fazendo uma análise de intervalos avaliando consonâncias e dissonâncias para determinar o tipo do acorde. Dois modelos conexionistas para a classificação de acordes, um cognitivo e outro psicoacústico, são apresentados por Berenice Ladem at all em (Todd & Loy, 1991). Para a tarefa da classificação da tonalidade da música tem-se o trabalho de Scarborough at all que implementa uma rede neural de perceptrons (Todd & Loy, 1991). Para a classificação tonal tem-se ainda o trabalho de Marc Leman que também utiliza o modelo KFM (Todd & Loy, 1991).

A bibliografia acima citada indica fortemente a possibilidade de solução dos subproblemas da classificação dos acordes e da classificação da tonalidade via modelos conexionistas. Os subproblemas da classificação de inversões e de graus são problemas classificatórios simples que podem ser facilmente resolvidos através da construção de bases de regras. O subproblema da identificação de acordes é menos trivial do que parece iniciamente pois é dificil determinar quando os três tipos de acordes anteriormente citados (bloco, quebrado e arpejado), formam uma estrutura harmonicamente importante. A seguir serão indicadas maneiras de resolver os subproblemas da classificação harmônica.

3.1 Identificação de acordes

Na literatura sobre harmonia os acordes são as estruturas musicais identificadas para determinar a harmonia. O conceito de acorde diz que três ou mais notas que soam simultaneamente formam um acorde. Baseado na simplicidade do conceito, a identificação do que é um acorde na música seria uma tarefa trivial, porém sabe-se que o que soa simultaneamente pode não ter sido tocado simultaneamente como no caso dos acordes quebrados e arpejados. Isto pode dificultar bastante a identificação dos acordes principalmente se o sistema trabalhar com eventos MIDI, pois uma nota que já foi desligada pode continuar soando. Neste caso, simplesmente a verificação de quais notas estão em *note on* não é suficiente.

Para resolver este problema basta determinar um método de identificar-se, na música, a ocorrência de acordes quebrados e de acordes arpejados. Reconhece-se, no entanto, que isto não é uma tarefa computacionalmente simples pois envolve um problema de explosão combinatória. Para determinar quando notas que são tocadas separadamente formam ou não acordes exige-se combinar todas as notas tocadas num determinado trecho de música a fim de encontrar os possíveis acordes. Um trabalho bastante aprofundado pode ser realizado a fim de determinar que eventos musicais devem ser considerados acordes e quais eventos musicais devem ter importância para a análise harmônica. Este autor não vai apontar nenhuma solução que envolva inteligência artificial para a solução da identificação de acordes, pois este problema somente foi identificado pouco antes da fase de validação do modelo proposto, não permitindo um estudo mais aprofundado do mesmo. A busca de soluções de Inteligência Artificial para a solução deste problema fica apontado aqui como tema para trabalhos futuros. A solução implementada neste trabalho considera como acordes todo conjunto de três ou mais notas tocadas simultaneamente. Mais detalhes sobre esta solução serão apontados na validação.

3.2 Classificação de Acordes

Um estudo sobre classificação de acordes foi realizado por este autor no trabalho entitulado Um Modelo Conexionista para Classificação de Acordes (Beckenkamp & Engel, 1994). O modelo proposto naquele trabalho utilizava uma rede neural supervisionada e foi utilizada neste trabalho com pequenas modificações que serão detalhadas a seguir.

No caso do uso de redes neurais artificiais, o problema da classificação de acordes consiste em submeter um acorde às entradas da rede neural e esta, através de suas unidades de saída, responder qual é o tipo do acorde. A rede neural proposta neste trabalho deve classificar os 13 tipos de acordes definidos em (Beckenkamp & Engel, 1995), que são: as tríades, os acordes de sétima e os acordes de nona. A escolha destes tipos foi feita levando em consideração a sua vasta utilização na música.

A rede neural utilizada foi a rede *Back-propagation* (BPN) que foi formalizada por Werbos, por Parker, e posteriormente por Rumelhart e McClelland (Rumelhart & McClelland, 1986). A BPN é utilizada na resolução de problemas que requeiram o reconhecimento de padrões complexos, e que necessitem realizar mapeamentos de funções não triviais.

A camada de saída da rede foi aumentada de 3 para 5 unidades. O tipo do acorde é determinado através da verificação dos intervalos formados entre as notas que compõem os acordes. O esquema de codificação dos padrões de saída reflete a estrutura dos intervalos do acorde e foi mantida para as três primeiras unidades de saída. A primeira unidade de saída representa o intervalo entre a fundamental e a terça do acorde (1ª terça), e sua saída deve ser 1 se o intervalo for de terça menor e 0 se for de terça maior. A segunda unidade de saída representa o intervalo entre a terça e a quinta do acorde (2ª terça), sua saída deve ser 1 se o intervalo for de terça menor e 0 se for de terça maior. A terceira unidade de saída representa o intervalo entre a fundamental e a quinta do acorde e sua saída deve ser 1 se o intervalo for de quinta justa e 0 se for de quinta diminuta.

A quarta e quinta unidades foram acrescentadas para modelar os acordes que possuem intervalos de sétima e de nona. A quarta unidade representa o intervalo entre a fundamental e a sétima do acorde. Sua saída será 0 quando o acorde nao possuir sétima, 0.3 quando a sétima for diminuta, 0.6 quando for menor, e 1 quando for maior. A quinta unidade de saída representa o intervalo entre a fundamental e a nona do acorde e sua saída deve ser 0 quando o acorde nao possuir nona, 0.5 se o intervalo for menor e 1 se for maior.

A rede do trabalho (Beckenkamp & Engel, 1994) classificava apenas tríades, ou seja, acordes formados por 3 notas. Neste trabalho acordes de 4 e 5 notas também foram considerados. Estes acordes são os de sétima (4 notas) e os de nona (5 notas). Desta forma, o número de tipos de acordes que a rede passou a ser treinada aumentou de 4 para 13.

Para o treinamento passou a ser utilizado um conjunto de 156 acordes, doze de cada tipo.O conjunto de 156 acordes foi subdividido em 4 conjuntos de acordo com a estrutura do acorde. O primeiro conjunto contém as tríades, o segundo conjunto contém os acordes de sétima menor, o terceiro os acordes de sétima maior e o quarto os acordes de nona. Esta divisão foi feita para que houvesse um equilíbrio de treinamento entre os acordes que possuem constituição intervalar diferente. A cada ciclo de treinamento um acorde de um grupo diferente é apresentado à rede. Desta forma, por exemplo, nunca dois acordes que possuem intervalo de nona serão apresentados à rede consecutivamente. Internamente os grupos são organizados de forma que dois acordes de mesmo tipo ou de mesma tonalidade não sejam apresentados consecutivamente.

3.3 Classificação da Tonalidade da Música

Sabe-se muito pouco sobre o processo cognitivo humano realizado na identificação da tonalidade.

Mesmo assim, buscou-se modelar o problema a partir do conhecimento cognitivo através de subsídios obtidos do conhecimento teórico de tonalidade. Existem três domínos na organização tonal: notas, acordes e tonalidades. Uma música sempre possui uma ou mais tonalidades. A tonalidade determina um subconjunto de 7 notas que são extraídas das 12 notas da escala cromática e a cada nota dentro de uma escala é associada uma função. Os acordes e notas são ouvidos de acordo com o contexto provido pela estrutura tonal e a tonalidade da música é inferida por um ouvinte através das notas e acordes que ele está ouvindo. A ordem e freqüência em que as notas são apresentadas podem influenciar na percepção da tonalidade. Além disso, certas notas da tonalidade são mais importantes para a tonalidade. Deve-se ainda acrescentar que mudanças de tonalidade são rapidamente percebidas por um ouvinte.

O modelo de classificação de tonalidade apresentado por Scarborough em (Todd & Loy, 1991) implementa um modelo bastante simples e eficiente pois programa algumas funções tonais diretamente na rede neural. Este modelo serviu como base na construção do modelo de classificação tonal deste trabalho.

Neste trabalho, implementou-se uma rede neural baseada em perceptrons[1] que mede a freqüência com que ocorrem as notas e a importância das notas ocorridas para a percepção da tonalidade. A tonalidade de maior evidência será a que tiver mais notas sendo tocadas e cuja importância for mais significativa. Este processo deve ser dinâmico pois se a tonalidade for modificada durante a música, a rede deverá ser capaz de perceber esta mudança e identificar a nova tonalidade. Esta característica justifica o uso de uma rede neural, pois esta dinâmica é inerente à sua estrutura. A tonalidade com maior ativação será a escolhida, podendo esta ativação mudar de uma tonalidade para outra.

A camada de entrada da rede neural proposta possui 12 neurônios, cada um correspondendo a uma nota da escala cromática. Quando uma nota é tocada, o neurônio correspondente recebe uma ativação. A segunda camada é formada por 24 neurônios correspondendo às doze tríades maiores e às doze tríades menores da música tonal.

As ligações sinápticas entre a camada de entrada e a camada de acordes é determinada pela existência ou não da nota no acorde. Ex.: Os neurônios de entrada referentes às notas Dó-Mi-Sol possuem ligação sináptica válida com o neurônio do acorde Dó maior. Ou seja, o neurônio correspondente a cada acorde receberá ativação somente dos neurônios correspondentes às notas que o formam.

Por último, modelou-se uma camada de 24 neurônios correspondendo às 24 tonalidades da música tonal ocidental, 12 maiores e 12 menores. Os neurônios de tonalidade são ativados pelos neurônios de acordes que são mais freqüentes para a tonalidade. Estes acordes são os acordes de tônica, dominante e subdominante. Neste caso, a tonalidade de Dó maior receberá ativação dos neurônios correspondentes aos acordes Dó, Fá e Sol maior.

A distribuição dos pesos na rede foi realizada de forma intuitiva procurando valorizar o que teóricamente é mais importante. A nota fundamental do acorde e o acorde tônico da tonalidade são os mais estáveis de uma tonalidade e indicam a tonalidade da música. As sinapses que ligam as fundamentais e as tônicas da tonalidade receberam peso 2. Ex.: No acorde Dó maior a nota Dó recebeu sinapse com peso 2 e as demais notas, Mi e Sol, receberam peso sináptico igual a 1. Na tonalidade Dó maior o acorde de Dó maior recebeu peso sináptico 2 e os acordes de Fá maior e Sol maior outros pesos sinápticos. As sinapses, tanto de notas quanto de acordes, que não contribuem para a evidência da tonalidade recebem peso 0. Pela experiência, verificou-se que é importante também valorizar as dominantes dos acordes e os acordes de subdominante, para evitar que o sistema não consiga encontrar uma tonalidade no caso de ocorrência de sucessivos acordes de tônica e dominante muito comuns na música. Suas ligações sinápticas foram colocadas em 1,5.

A partir do momento em que uma nota é tocada já se pode verificar o valor das ativações dos neurônios de saída, o neurônio que possuir maior ativação determinará a tonalidade. Em qualquer ponto da música a tonalidade pode ser verificada através do neurônio de tonalidade que possuir maior ativação, o que faz com que o modelo suporte modulações.

A rede neural percebe a tonalidade a partir das notas que são tocadas. A medida em que a música vai sendo tocada mais notas da mesma tonalidade são tocadas, aumentando a evidência da tonalidade. Porém a música pode possuir modulações, o que determina que notas de outra tonalidade aparecerão e deverão modificar a evidência da tonalidade anterior para a nova tonalidade. Para que a rede neural perceba a

[1] Mais informações sobre perceptrons pode ser obtidas no livro (Freeman & Skapura, 1992).

tonalidade a medida em que as notas vão acontecendo, é preciso que se crie uma forma de memória das notas que foram tocadas. No entanto, esta memória deve ter menos importância para o reconhecimento da tonalidade do que as notas mais recentes de forma que a rede possa rapidamente perceber modulações.

Este sistema de memória do que já foi tocado pode ser realizado através do controle das ativações que as notas geram quando são tocadas. Neste momento, o neurônio correspondente recebe a ativação máxima e permanece com ativação durante algum tempo após ser desacionada. A permanência da ativação após a nota ter sido desligada faz com que as notas que já foram tocadas continuem sendo importantes na escolha da tonalidade. Caso não fosse implementado um sistema de memória, possivelmente uma seqüência de notas resultaria numa seqüência de tonalidades. Este trabalho sugere três maneiras de controlar a ativação dos neurônios de entrada: linear, com decaimento e com ataque e decaimento.

No controle linear, quando uma nota é tocada, o seu neurônio correspondente recebe a ativação máxima (igual a 1). A ativação permanece a mesma durante o tempo em que a nota estiver sendo tocada. Depois que a nota é desligada a ativação também permanece igual durante um tempo que pode ser determinado pelo usuário. Após o tempo estipulado, a ativação cai para zero.

No controle com decaimento, o neurônio recebe ativação máxima durante o tempo em que a nota for tocada. Quando a nota é desacionada, a sua ativação não é imediatamente levada a zero, ela entra em processo de decaimento ao longo do tempo.

No controle com ataque e decaimento a duração da nota influencia na quantidade de ativação aplicada ao neurônio. À medida em que a nota vai sendo sustentada, aumenta o valor da ativação por ela gerada (ataque). Isto faz com que as notas tocadas durante mais tempo influenciem mais na ativação final.

Com a implementação do ataque e do decaimento, as notas mais recentes terão maior influência na ativação da rede, fazendo com que haja a possibilidade de modulações durante a música. Este mecanismo implementa automaticamente um controle de grau de certeza de qual é a tonalidade da música. Quanto mais tempo notas da mesma tonalidade permanecerem sendo ativadas, maior será o grau de certeza de que a tonalidade da música é aquela. Para implementar o ataque e o decaimento basta implementar funções exponenciais na ativação de cada nota.

3.4 Inversões de Acordes

A solução encontrada para resolver o problema das inversões de acordes foi construir um sistema de regras. Para os acordes não simétricos basta verificar a distância do baixo do acorde com as demais notas do acorde. As inversões dos acordes são determinadas por qual nota formadora do acorde é o baixo. Dependendo da nota do acorde que é o baixo, são formados intervalos diferentes entre as notas do acorde. Por exemplo: o acorde dó-mi-sol possui um intervalo de terça maior entre o baixo, a nota dó, e a segunda nota do acorde, a nota mi. Além disso possui um intervalo de quinta justa entre o baixo (dó), e a terceira nota (sol). Caso este acorde estivese na primeira inversão teria a seguinte estrutura: mi-sol-dó. A distância entre o baixo, agora a nota mi, e a segunda nota do acorde (sol), é de uma terça menor e o intervalo entre o baixo (mi), e a terceira nota (dó), é de uma sexta.

Para ter-se um controle mais preciso destes intervalos deve-se verificar os intervalos na forma de distância em semitons. Isto é necessário pois cada intervalo pode assumir vários tipos (maior, menor, justo, diminuto e aumentado). Deve-se ter cuidado para verificar com exatidão o tipo de intervalo formado para cada tipo de acorde em cada inversão. Por exemplo verificar que tipo de intervalo de sexta tem-se no acorde mi-sol-dó, do exemplo anterior. A Tabela 3.4.1 a seguir, apresenta as distâncias em semitons existentes entre o baixo e as demais notas que formam um acorde do tipo perfeito maior nas três posições possíveis. Os números que aparecem em negrito determinam quais os intervalos que devem ser levados em consideração para que se possa classificar as inversões do acorde.

No modelo proposto neste trabalho o classificador de inversões deve classificar as inversões dos acordes previamente selecionados e classificados pelo classificador de acordes. Tendo-se a informação do tipo do acorde, pode-se verificar os intervalos em semitons formadores do acorde. Para cada tipo de acorde pode-se construir um conjunto de regras que classificam a inversão do acorde. Estas regras podem ser construídas a partir de tabelas como a Tabela 3.4.1 abaixo.

Tabela 3.4.1 Acorde PM

Inversão	1º Intervalo	2º Intervalo
Fundamental	**4**	7
1ª Inversão	**3**	8
2ª Inversão	**5**	9

Analisando-se a Tabela 3.4.1, verifica-se que para acordes perfeitos maiores basta identificar o tipo do intervalo formado entre o baixo e a 1ª nota do acorde. Caso o intervalo seja de 4 semitons sabe-se que o acorde está em posição fundamental, 3 semitons em 1ª inversão e 5 semitons em 2ª inversão. Ou seja, o intervalo entre o baixo e a segunda nota pode ser desprezado. Porém isto não ocorre para todos os tipos de acordes. Na maioria dos casos deve-se verificar mais de um intervalo para que se possa distinguir as inversões. Regras de intervalos para todos os tipos de acorde considerados neste trabalho podem ser facilmente formadas. Um mecanismo de aprendizado pode ser implementado afim de permitir que o conhecimento seja incrementado através da evolução das regras.

Através deste método também não se resolve o problema dos acordes simétricos, pois os intervalos formados entre o baixo e as demais notas do acorde são sempre os mesmos para quaisquer inversões. A solução encontrada foi construir algumas regras levando em consideração o grau em que os acordes simétricos podem ocorrer na harmonia tonal. Estes acordes acontecem somente na tonalidade menor e somente em um grau da escala. O acorde de 5ª aumentada ocorre somente para o terceiro grau da escala menor e o acorde de sétima diminuta somente no sétimo grau da escala menor. Tendo-se a informação de qual escala menor a música está sendo tocada, quando ocorrer um dos acordes simétricos, pode-se verificar facilmente qual é o terceiro ou o sétimo grau da escala. Por exemplo: caso a música esteja em lá menor e acontecer um acorde de 5ª aumentada que este acorde terá grau III. Sendo assim, conclui-se que a nota dó, que é o grau 3 da escala de lá menor, é a nota fundamental do acorde tocado. Verificando-se a distância entre o baixo e a fundamental do acorde, no exemplo a nota dó, pode-se concluir qual a inversão dos acordes simétricos.

3.5 Classificação dos graus

Na literatura sobre harmonia encontram-se discordâncias entre os autores com relação aos graus em que ocorrem cada tipo de acorde. Apesar disto, a classificação dos graus dentro do modelo proposto neste trabalho tornou-se uma tarefa bastante simples, não justificando a busca de soluções complexas como o uso de redes neurais. Neste trabalho buscou-se unificar as informações fornecidas nas várias literaturas consultadas criando-se regras que mapeiam em que graus de cada escala ocorrem os tipos de acordes considerados neste trabalho. Um mecanismo de aprendizado também pode ser implementado para a classificação de graus, permitindo que as regras que definem em que graus cada acorde ocorre possam evoluir com o uso do sistema.

Alguns acordes podem ser mapeados diretamente para um sistema de regras, pois eles ocorrem somente em um grau de determinada tonalidade, como, por exemplo, o acorde de 7ª diminuta que sempre será o grau VII. Porém para aqueles acordes que podem ocorrer em vários graus de uma mesma tonalidade deve-se construir um sistema de regras que leve em consideração informações complementares que possam eliminar a ambigüidade. Por exemplo, o acorde PM pode ser tanto o grau I como o grau IV ou ainda o grau V em tonalidades maiores. Para eliminar-se esta ambigüidade através de regras, pode-se verificar as notas do acorde que somente acontecem em um determinado grau. Por exemplo, se a tonalidade for dó maior e o acorde for dó-mi-sol (PM) sabe-se que este acorde é o grau I, pois a nota mi está presente e ela não ocorre nos acordes de IV e V graus da tonalidade de dó maior (fá-lá-dó e sol-si-ré respectivamente). Pode-se generalizar esta regra para todos os acordes PM em tonalidade maior se considerarmos que a nota mi é o terceiro grau da escala de dó maior. Sabendo-se a tonalidade da música, pode-se saber que nota é cada grau desta escala e com posse das notas que formam o acorde a ser classificado, pode-se verificar se

determinado grau da escala pertence ao acorde.
As regras criadas mapeando em que graus de cada escala ocorrem os acordes podem ser facilmente transformadas num sistema de regras que resolverá a classificação dos graus para as escalas maiores e menores de uma maneira bastante simples. No caso das escalas menores o sistema de regras ficará mais complicado, pois têm-se três tipos de escalas menores, o que aumenta a ambigüidade da classificação. A diferença entre estas três escalas está no sexto e sétimo graus que podem aparecer alterados. Estas alterações determinam que outros tipos de acordes podem ser contruídos nestas escalas. A presença destes graus alterados pode ser utilizada como diferenciadora na classificação dos graus.

4 Validação

A implementação do modelo proposto exigè a escolha de uma plataforma de hardware e software que permita realizar diferentes tarefas que possuem diferentes necessidades tanto em hardware quanto em software. O hardware deve ter boa velocidade de processamento numérico a fim de facilitar o treinamento das redes neurais. Além disso, o hardware deve possuir interface MIDI o que permite a entrada da música a ser analisada. O software deve ser capaz de trabalhar com MIDI e oferecer ferramentas para a implementação de sistemas de inteligência artificial que envolvam tanto o paradigma conexionista, quanto o simbólico.

Para o desenvolvimento de programas de redes neurais a linguagem C foi escolhida, pois este tipo de programa exige uma linguagem que aproveite ao máximo a velocidade de processamento do hardware disponível. Devido à necessidade de um ambiente de software que facilite a manipulação de MIDI e que também permita o desenvolvimento de programas em linguagem C, o software Max foi escolhido. Além disso, a linguagem Max também permite a criação de programas de Inteligência Artificial baseados em regras, pois possui uma ferramenta chamada Pyrite que implementa uma linguagem baseada em Lisp.

Os subproblemas foram desenvolvidos na forma de objetos Max. Programas Max foram desenvolvidos para testá-los separadamente afim de se verificar o seu desempenho para a realização do subproblema a que foi destinado. Da mesma forma, foi criado um programa Max para avaliar o desempenho do modelo proposto, onde os objetos Max que implementam os subproblemas, interagiram a fim de realizar a classificação harmônica. Abaixo os resultados dos testes das implementações dos subproblemas individualmente e, por último, o teste do modelo proposto.

Tabela 4.1 Resultados do modelo de Laden

Rede	Unidades Intermediárias	Arquitetura	η	α	Codificação da Saída	Porcentual de acerto	Iterações
6	3	Adjacente[2]	0,6	0,6	Intervalar	44	10000
7	25	Adjacente	0,8	0,8	Intervalar	94	3100
8	25	Adjacente	0,9	0,2/0,9	Intervalar	58	5000

O subproblema da classificação de graus depende muito fortemete dos dados fornecidos pelos outros subproblemas. Em conseqüência disto, torna-se difícil testá-lo individualmente. Testes foram realizados somente com o intuito de verificar se a programação das regras foram corretamente implementadas. O subproblema da classificação de inversão de acordes também requereu apenas um teste, a fim de verificar a correta implementação das regras.

Já o classificador de acordes mereceu atenção especial em sua validação. Para ter-se uma noção correta do desempenho do classificador de acordes, foram comparados os resultados da rede neural proposta com os melhores resultados obtidos pelo modelo conexionista cognitivo apresentado por Berenice Laden em (Todd & Loy, 1991), que foram transpostos para a Tabela 4.1 acima. Na Tabela 4.2 abaixo é mostrado o melhor resultado obtido no treinamento da rede proposta.

2 A arquitetura chamada de adjacente por Laden é mais conhecida por *feed-forward*.

Tabela 4.2 Resultados do modelo proposto

Rede	Unidades Intermediárias	Arquitetura	η	α	Codificação da Saída	Porcentual de acerto	Iterações
1	25	Adjacente	0,9	0,25	Intervalar	100	146133

A rede número 1 da Tabela 4.2 apresentou um resultado plenamente satisfatório bastante superior aos resultados encontrados na bibliografia pesquisada. A taxa de aprendizado (η) utilizada para o treinamento foi de 0,25 e o momentum (α) foi de 0,9.

O sucesso alcançado no treinamento do modelo provou que a classificação de acordes via modelo conexionista cognitivo por intervalos é viável. O modelo proposto é capaz de classificar todos os intervalos possíveis para acordes da música tonal considerados no estudo de harmonia.

Para validar o classificador de tonalidade e o classificador de harmonia, foram selecionados no livro (Kostka & Payne, 1984), trechos de música previamente classificada por seus autores. Desta forma, basta confrontar os resultados do sistema com a classificação dada pelos autores. Os trechos de música escolhidos abrangem uma boa parcela da teoria envolvida na classificação harmônica considerada por este trabalho. São eles: Exemplo 7-7, Exemplo 7-10 e Exemplo 7-11. Abaixo, a Figura 4.1 mostra o trecho 7-7 com as classificações do livro.

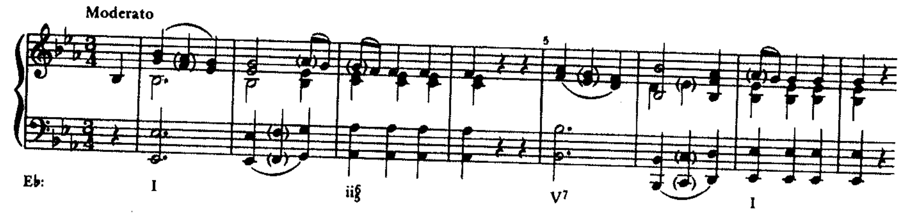


Figura 4.1 Beethoven, Minuet

Para que o sistema classifique os trechos de música é necessário que estes sejam fornecidos ao sistema na forma de eventos MIDI. Os trechos de música podem ser tocados no controlador MIDI A-30 que é um gerador de eventos MIDI. Os eventos gerados pelo A-30 são enviados para o módulo de som SC-55 que os repassa para o macintosh. Este procedimento de teste é cansativo pois toda vez que se quiser testar o programa, deve-se tocar o trecho de música. Uma alternativa é seqüenciar o trecho de música num software apropriado. O seqüenciador deve ser conectado ao programa de classificação de harmonia de maneira que os eventos MIDI previamente seqüenciados possam ser avaliados em tempo real, quantas vezes se deseje e sem a necessidade de tocá-los.

Os trechos de música foram seqüenciados no Cakewalk para Windows. O PC 486 Dx4 do laboratório de música e computação possui uma placa MIDI. Através da saída MIDI da placa os eventos seqüenciados são passados para a entrada MIDI do módulo de som SC-55, que por sua vez está conectado ao computador macintosh. Como já foi explicado anteriormente, estes dados MIDI chegam no macintosh e são interceptados pelo programa Max. O programa desenvolvido na linguagem Max pode então manipular os eventos MIDI livremente.

A classificação da tonalidade de todos os trechos de música testados foi correta. Os controladores de ataque e decaimento foram utilizados de várias maneiras para verificar o melhor resultado para cada trecho analisado. As características de ritmo e andamento de cada trecho de música determinam que parâmentros são os mais adequados para a classificação da tonalidade. Com alguma experiência, facilmente se identifica quais parâmetros são os mais adequados para cada música. Normalmente o classificador de tonalidade demora aproximadamente um compasso para começar a classificar corretamente a tonalidade. O tempo em que o classificador leva para perceber a modulação também fica em torno de um compasso.

A eficiência do classificador de tonalidade depende sensivelmente dos pesos da rede e dos controles de ataque e decaimento. O modelo percebe corretamente a tonalidade em vários contextos musicais como escalas, acordes, cadências, arpejos, etc; e classifica rapidamente, com grau de acerto aceitável e consistente com relação a modulações. A percepção de algumas regras musicais que também dão pistas para a tonalidade poderiam ser consideradas pelo modelo. Ex.: intervalos de segunda menor ascendente resolvem na fundamental.

O objeto classificador de harmonia classificou satisfatoriamente os três trechos de música selecionados. Os dois primeiros trechos testados foram corretamente classificados e apenas dois erros foram cometidos no terceiro trecho devido a aparição de acordes incompletos que não formam classificados corretamente. No trecho da Figura 4.1, os dois primeiros compassos foram classificados com grau I. No compasso 3, o primeiro acorde é classificado como grau IV. Porém, assim que ocorre a nota fá, a classificação muda para o grau ii na segunda inversão, como indicado no livro. O mesmo ocorre no quinto compasso, onde aparece um acorde quebrado. Somente quando todas as notas formadoras do acorde são tocadas é que o sistema chega à classificação harmônica definitiva, que é o grau V. Como se pode verificar, o sistema classificou corretamente acordes importantes, como os acordes de grau I, V e ii.

Problemas como os acima citados já foram exaustivamente analisados pelos estudos de análise harmônica e por certo podem ser facilmente isolados. O modelo deste trabalho possui flexibilidade suficiente para poder incorporar soluções para estes problemas quer seja via redes neurais, como via soluções híbridas.

5 Conclusão

Os resultados obtidos são positivos e motivadores, porque mostram que o problema da classificação harmônica tonal pode ser resolvido via modelos de inteligência artificial. A solução da classificação harmônica tonal foi buscada através de uma modelagem que agrupasse soluções para determinadas partes do problema. O paradigma conexionista apresentou-se como solução natural e adequada para problemas cognitivos devido as suas propriedades de tolerância a falhas e generalização. A solução dos problemas de inversões de acordes e classificação de graus através de bases de regras, mostrou-se bem dimensionada diante da complexidade destes problemas. O modelo criado proporcionou uma solução rápida e eficiente para a classificação harmônica.

A implementação do modelo foi possível devido à acertada opção pelo seu desenvolvimento através da linguagem Max que permitiu o desenvolvimento dos programas de redes neurais e bases de regras, facilitando também o trabalho com música na forma de eventos MIDI. O uso de um computador Macintosh Quadra é necessário para que o classificador de harmonia tenha um desempenho que permita realizar sua tarefa em tempo real. Os resultados obtidos e a flexibilidade de programação da linguagem Max permitem que os objetos criados para a solução dos subproblemas da classificação harmônica e o próprio programa de classificação harmônica possam ser utilizados em novos projetos e programas que busquem resolver outros problemas da música.

O modelo desenvolvido neste trabalho fornece resultados analíticos, pois realiza automaticamente a classificação da harmonia tonal. Estes resultados podem ser utilizados em sistemas que requeiram o conhecimento da harmonia, como sistemas de acompanhamento ou composição automática. Estes sistemas auxiliam o músico no aperfeiçoamento de seus conhecimentos e de seu trabalho. Eles também possibilitam a criação de interfaces inteligentes entre o computador e o músico, permitem novas formas de estudo, organização e utilização do material musical, e reduzem a complexidade do estudo e do aprendizado da música.

É importante salientar que o estudo do problema não encerra aqui, pois o modelo criado implementa parte do conhecimento sobre harmonia, podendo evoluir através da inclusão de mais conhecimento, a fim de aumentar a precisão e rapidez da classificação harmônica. Este conhecimento pode continuar a ser buscado na teoria e nos especialistas, mas também pode ser buscado através de soluções que modelem partes do conhecimento, como este trabalho. O modelo proposto mostrou ser abrangente e conciso, formando um alicerce bastante forte para trabalhos futuros.

A teoria que envolve a análise harmônica certamente é bastante ampla, não podendo ser totalmente considerada neste trabalho. No entanto, ele aponta caminhos que ainda não foram totalmente explorados e que, certamente, podem ajudar a entender e a resolver muitos dos desafios da análise harmônica. Espera-se que estes caminhos possam continuar a ser trilhados tanto por pesquisadores da música, como da ciência da

computação.

Bibliografia

Balaban, Mira; Ebcioglu, Kemal; Laske Otto (1992). *Undestanding Music and IA*: perspectives on music cognition. MIT Press, Cambridge, MA, 1992.

Beckenkamp, Fábio Ghignatti; Engel, Paulo Martins (1994). A Connectionist Model for Chord Classification. *Proceedings of* I Simpósio Brasileiro de Computação e Música. Caxambu, MG, Brazil, August 1994.

Beckenkamp, Fábio Ghignatti; Engel, Paulo Martins (1995). A Proposal of a Harmonic Classification Artificial Intelligence Model. *Proceedings of* International Joint Conference on Artificial Intelligence 1995. Montréal, Québec, Canada, August 1995.

De Poli, Geovanni; Piccialli, Aldo and Roads, Curtis (1991). *Representations of Music Signals*. MIT Press, London, England.

Freeman, J.; Skapura (1992), D. *Neural Networks:* Algorithms, Aplications, and Programing Technics. Addison-Wesley Publishing Company, p. 89-124.

Kostka, Stefan; Payne, Dorothy (1984). *Tonal Harmony,* with an introduction to the twentieth-century music. Alfred A. Knopf Inc, New York, NY, v. 1, p. vii-x, 17-21, 39-42, 44-46, 54-57, 60-62, 241, 421-422.

Rumelhart, David; McClelland, James (1986). *Parallel Distributed Processing*. MIT Press, Cambridge, MA, v.1, p. 318-334.

Todd, Peter; Loy D. Gareth (1991). *Music and Connectionism*.. MIT Press, Cambridge, MA.

Digital Waveguide Modeling of Air-Driven Reed Generators for the Synthesis of Brass and Woodwind Instrument Sounds

GARY P. SCAVONE

Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305 USA
gary@ccrma.stanford.edu

Abstract

This paper reviews past digital waveguide methods for reproducing non-linear "reed" excitations as well as introducing a new method incorporating reed dynamics. This model is based on a mass-spring-damper system and a non-linear flow control mechanism. In this way, a more physical system is attained which provides better approximations to the control parameters of real musical instruments. Further, it is shown that this model of the reed can be modified to represent the lips of a brass player, and the similarities and differences between these two systems are briefly examined.

1 Introduction

Musical instruments are most clearly distinguished from one another by their transient characteristics, which in turn are defined by a particular method of excitation. Among wind-blown instruments, for example, the various air-driven excitation methods distinguish saxophones from trumpets or flutes. In the context of digital waveguide modeling of musical instruments, highly accurate models of these non-linear excitation methods have proven difficult to produce. Two effective digital waveguide reed woodwind excitation methods have previously been presented (Smith 1986) (Cook 1992), though these models lack the *physical* control parameters associated with their real world counterparts. A dynamic waveguide reed model incorporating a mass-spring-damper system and non-linear flow control is presented here. The modeling of the reed in this way for waveguide applications was previously discussed in (Hirschman 1991) for woodwinds and (Cook 1991*b*) for brasses, though the flow control mechanisms implemented were different from that discussed in this paper.

2 Acoustical Aspects of Air-Driven Reed Generators

The acoustical properties of reed generators have been extensively studied (see Fletcher & Rossing 1991 for references). Two distinct types of reed generators exist those in which the reed valve is initially closed and then blown open (as with a brass player's lips) and those in which the reed valve is initially open and then blown closed (as for clarinets and saxophones). In most cases, the reed itself is modeled as a linear oscillator, and the pressure on the reed is taken equal to the difference in oral cavity and bore pressures. The position of the reed in turn governs the volume flow through the reed aperture, for which Bernoulli's flow equation forms a first approximation. Possible modifications to the flow equation include terms to compensate for reed channel inertia and the physical motion of the reed surface. Recent fluid-dynamic studies of flow through a reed aperture have suggested the need to account for viscous flow (Hirschberg et al. 1990). Non-linearity of the reed stiffness has also been discussed (Gilbert et al. 1990).

In woodwind instruments the reed resonance is normally high compared to the operating frequency of the reed. A mass-spring system driven at a frequency well below resonance is said to be stiffness dominated and its displacement amplitude will approach f/k, where k is the spring constant and f is the applied force. Thus, a common simplification for woodwind instruments has been to neglect the effect of the mass altogether and to simply model the reed system as a memory-less non-linearity. Assuming

the force on the reed is equal to $A \cdot p_\Delta$, where p_Δ is the difference in oral cavity and bore pressures and A is the approximate surface area of the reed exposed to p_Δ, the displacement of the reed from its equilibrium position (x_0) is given by Hooke's Law

$$x = \frac{A \cdot p_\Delta}{k} + x_0. \tag{1}$$

The area of the reed aperture is assumed to be proportional to x. Bernoulli's equation for steady volume flow through the reed aperture gives $u = \gamma |x| p_\Delta^{1/2}$, where γ is a constant dependent on the density of air and the area of the reed aperture. Combining this expression and Eq. (1), the volume flow is

$$u = \gamma \left| \frac{A p_\Delta^{3/2}}{k} + x_0 p_\Delta^{1/2} \right|. \tag{2}$$

Figure 1 displays the non-linear volume flow characteristic given by Eq. (2). An initial increase of p_Δ from zero results in a rapid increase in u. However, a continued increase of p_Δ begins to force the reed toward the mouthpiece lay, resulting in a decrease in volume flow. When $p_\Delta = p_{closure}$, the reed valve is completely closed.

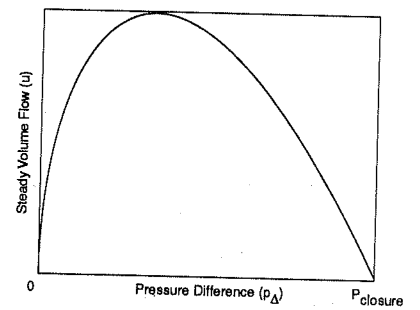


Figure 1: *Steady Flow Through a Pressure Controlled Valve Blown Closed*

3 Digital Waveguide Reed Generator Models

3.1 The Pressure Dependent Reflection Coefficient

McIntyre, Schumacher, & Woodhouse (1983) discussed the time-domain synthesis of clarinet sounds and incorporated a non-linear volume flow characteristic similar to that of Figure 1. Their procedure assumed continuity of volume velocity at the reed/bore junction and involved the simultaneous solution of a linear equation relating pressure to volume flow in the bore and a nonlinear approximation to Eq. (2), which related pressure to volume flow through the reed. As an efficient alternative to this process within the context of digital waveguide modeling, Smith (1986) proposed modeling the reed/bore boundary with a reflection coefficient that varies in response to the difference in oral cavity (p_{oc}) and bore pressures (p_b). The essential non-linear behavior of the reed is attained using an extremely simple calculation, though "higher order" reed behavior is sacrificed.

The pressure dependent reflection coefficient is derived by assuming continuity of volume velocity at the reed/bore junction,

$$\frac{p_\Delta}{Z_{oc}(p_\Delta)} = \frac{p_b^+ - p_b^-}{Z_b}, \qquad p_\Delta \triangleq p_{oc} - \left[p_b^+ + p_b^-\right] \tag{3}$$

and defining the reflection coefficient

$$\rho(p_\Delta) \triangleq \frac{1+r(p_\Delta)}{1-r(p_\Delta)}, \qquad r(p_\Delta) \triangleq \frac{Z_b}{Z_m(p_\Delta)}$$

Eq. (3) can then be solved for the reflected bore pressure at the junction, p_b^-

$$p_b^- = \rho(p_\Delta)p_b^+ + \frac{1-\rho(p_\Delta)}{2}p_{oc}. \tag{4}$$

Unfortunately, p_Δ is dependent on p_b^- and in order to solve Eq. (4) it is necessary to find an approximation to p_Δ. In a recursive, discrete-time calculation, it is possible to approximate $p_\Delta(n)$ by $p_\Delta(n-1)$ or to calculate $p_\Delta(n)$ using $p_b^-(n-1)$. Further, current values of either quantity could be extrapolated from previous values. The approach taken here is to define a new term, $p_\Delta^+ = p_{oc} - 2p_b^+$, which is independent of p_b^- and substitute this into Eq. (4) to obtain

$$p_b^- = \frac{p_{oc}}{2} - \rho(p_\Delta^+)\frac{p_\Delta^+}{2}. \tag{5}$$

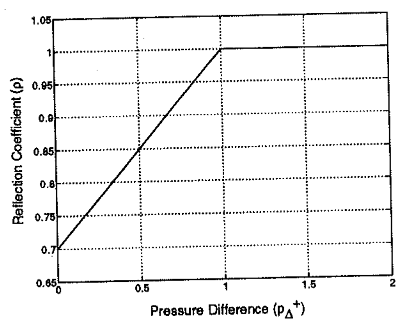


Figure 2: *Sample Reflection Coefficient Table*

The pressure dependent reflection coefficient is normally implemented using a look-up table, thereby saving one multiply and one addition per sample. Figure 2 displays a sample reflection coefficient table that has been used in synthesizing clarinet sounds. This particular table is based on normalized oral cavity pressure. Values of p_Δ^+ greater than 1.0 correspond to beating of the reed against the mouthpiece lay and complete reflection of incoming bore pressure. Values of differential pressure less than 1.0 correspond to partial reflection of p_b^+ and partial transmission of p_{oc} into the bore.

3.2 The Reed Reflection Polynomial

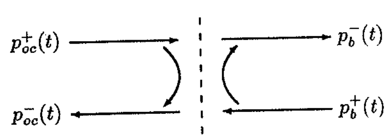


Figure 3: *Reed/Bore Scattering Junction*

The reed reflection polynomial incorporates the concept of a pressure dependent reflection coefficient but makes the assumption that p_Δ can be approximated by $p_{oc}^+ - p_b^+$. The polynomial model is derived by considering the reed/bore junction as shown in Figure 3. The portion of p_b^+ reflected back into the bore is given by $p_b^+ \cdot \rho(p_\Delta)$, while the portion of the oral cavity pressure which is transmitted into the bore is given by $p_{oc}^+(1-\rho(p_\Delta))$. Then p_b^- is given by

$$p_b^- = p_{oc}^+ - \left[p_{oc}^+ - p_b^+\right]\rho(p_\Delta). \tag{6}$$

Using the above stated approximation for p_Δ and approximating $\rho(p_\Delta)$ by a second order polynomial function, Eq. (6) becomes

$$p_b^- \approx p_{oc}^+ - \left[c_1(p_{oc}^+ - p_b^+) + c_2(p_{oc}^+ - p_b^+)^2 + c_3(p_{oc}^+ - p_b^+)^3\right]. \tag{7}$$

This reed implementation method has proven efficient and effective for real-time DSP synthesis. Unfortunately, the process of determining appropriate polynomial coefficients is rather arbitrary. It is possible to relate the polynomial coefficients to a polynomial approximation of the pressure dependent reflection coefficient through a matrix transformation (Cook 1991*a*).

3.3 The Dynamic Woodwind Reed Model

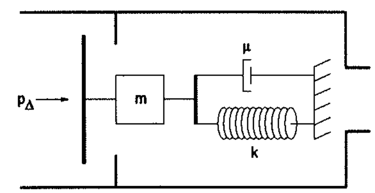


Figure 4: *Dynamic Woodwind Reed Model*

In contrast to the reed models previously discussed all of which ignored the mass of the reed the dynamic reed model seeks to accurately model the motion of the reed and its beating against the mouthpiece lay. In functioning as a pressure controlled valve, the position of the reed at any instant governs the volume flow that is injected at the reed/bore junction. The reed is represented by a linear mass-spring-damper system which is acted upon by the difference in oral cavity and bore pressures, as shown in Figure 4. The relationship between applied force and displacement, and the corresponding Laplace transform is given by

$$f_r(t) = m\frac{d^2x}{dt^2} + \mu\frac{dx}{dt} + kx \iff F_r(s) = \left[ms^2 + \mu s + k\right]X(s). \tag{8}$$

Both the oral-cavity and the bore pressures act upon the reed, so that the resultant force on the reed is

$$F_r(s) = A \cdot P_\Delta(s) = A \cdot [P_{oc}(s) - P_b(s)], \tag{9}$$

where A is the approximate surface area of the reed exposed to P_Δ. A is typically bounded by the width of the reed at its tip and the distance from the reed tip to the player's lower lip.

The transfer function that relates reed displacement to applied force is found from Eq. (8) as

$$\begin{aligned} \frac{X(s)}{F_r(s)} = H(s) &= \frac{1}{ms^2 + \mu s + k} \\ &= \frac{1/m}{s^2 + (\mu/m)s + \omega_o^2} \end{aligned} \tag{10}$$

where $\omega_o^2 = k/m$ is the natural frequency of the mass-spring system in the absence of damping. Using the bi-linear transform to convert from continuous to discrete time, the following digital transfer function results:

$$\begin{aligned} \frac{X(z)}{F_r(z)} = H(z) &= \frac{(1 + 2z^{-1} + z^{-2})}{(k + \alpha^2 m + \alpha\mu) + 2(k - \alpha^2 m)z^{-1} + (k + \alpha^2 m - \alpha\mu)z^{-2}} \\ &= \frac{(1 + 2z^{-1} + z^{-2})}{(m\omega_o^2 + \alpha^2 m + \alpha\mu) + 2m(\omega_o^2 - \alpha^2)z^{-1} + (m\omega_o^2 + \alpha^2 m - \alpha\mu)z^{-2}} \end{aligned} \tag{11}$$

where α is the bilinear transform constant used to control the frequency warping. The displacement found by passing $A \cdot p_\Delta(n)$ through this biquad section is subtracted from the reed's equilibrium position

(x_0) to produce the aperture spacing. Inelastic beating of the reed is assumed, such that the reed is forced against the lay and held there until the driving force decreases below $k \cdot x_0$, the force necessary to hold the spring stretched by x_0. The digital filter of Eq. (11) must be reset with the appropriate initial conditions each time this occurs. Figure 5 represents a transposed direct form II biquad filter structure that could be used to implement the reed filter. The appropriate internal state values to be used when the reed first begins to separate from the lay can be determined by inspection of the filter structure, given that the previous filter inputs and outputs are assumed to be $k \cdot x_0$ and x_0, respectively. In this case, the initializing values for the filter's internal states should equal $k \cdot x_0\,(b1 + b2) - x_0\,(a1 + a2)$ and $k \cdot x_0 \cdot b2 - x_0 \cdot a2$.

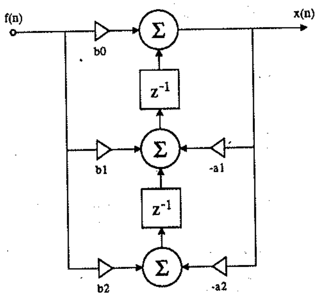


Figure 5: *Second Order Reed Filter*

Assuming that the Bernoulli flow equation applies to the given situation, the volume flow through the reed aperture is found from

$$\begin{aligned} u(t) &= A_r(t) \cdot v(t) \\ &= A_r(t) \cdot \left[\frac{2p_\Delta(t)}{\rho}\right]^{\frac{1}{2}} \end{aligned} \qquad (12)$$

where $A_r(t) = w \cdot x(t)$ is the time-varying area of the reed aperture, w is the width of the reed, and ρ is the density of air.

Finally, we assume continuity of volume velocity at the reed/bore junction and calculate the new traveling wave component of pressure entering the bore as,

$$u(t) = u_b^+(t) + u_b^-(t) = \frac{p_b^+(t) - p_b^-(t)}{Z_b} \qquad (13)$$

where $Z_b = \rho c / A_b$ is the constant acoustic characteristic impedance of the bore. $p_b^-(t)$ represents the traveling wave component of pressure entering the bore while $p_b^+(t)$ represents the traveling wave component of pressure leaving the bore. Solving for $p_b^-(t)$, we have

$$\begin{aligned} p_b^-(t) &= u(t) \cdot Z_b + p_b^+(t) \\ &= A_r(t) \cdot \left[\frac{2p_\Delta(t)}{\rho}\right]^{\frac{1}{2}} \cdot \frac{\rho c}{A_b} + p_b^+(t) \\ &= A_r(t) \cdot \frac{c}{A_b}\,[2\rho p_\Delta(t)]^{\frac{1}{2}} + p_b^+(t) \end{aligned} \qquad (14)$$

Measurements on a clarinet reed and mouthpiece (Backus 1963) have shown the exponent value in Eq. (14) to be on the order of $\frac{2}{3}$. This calculation can be simplified for real-time DSP implementation by the use of a look-up table. In order to implement Eq. (14) in discrete-time, it is necessary to use an approximation to $p_\Delta(n)$ because of its dependence on $p_b^-(n)$. Given sufficiently high sampling rates and the fact that the bore oscillations are low-pass filtered by the bell/tonehole filter, a reasonable approximation that has produced acceptable results when implemented is $p_\Delta(n) = p_{oc}(n) - \left[p_b^+(n) + p_b^-(n-1)\right]$.

3.4 Modeling a Brass Player's Lips

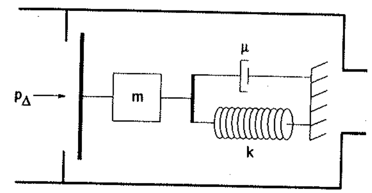


Figure 6: *Air-Driven Lip Model*

Figure 6 represents a basic air-driven lip model that can be used in brass instrument synthesis. In contrast to the woodwind reed model, this system is blown open and will oscillate near its resonance frequency. The derivation of Section 3.3 remains valid, though the lips do not beat against the mouthpiece. A similar model for brass instruments was developed in (Cook 1991*b*), though the discrete-time derivation and the flow control equations were different. In the woodwind model, frequency variation is principally controlled by adjustment of the resonator (delay-line) length, and the reed oscillations adjust appropriately. For the brass instrument model, however, both the acoustic resonator length and the lip model variables (mass, stiffness, and damping) control the sounding frequency. For a fixed resonator length corresponding to a fundamental frequency f_0, modification of the lip variables will cause the lip oscillations to become entrained at various partials of f_0. Likewise, holding the lip variables constant and varying the resonator length will result in a sounding frequency which fluctuates slightly about the resonance frequency of the lips. In essence, it is necessary that the lip resonance and the resonator length be modified in conjunction, as is required in the performance of real brass instruments. For these reasons, performance of waveguide brass instrument models is more difficult than for their woodwind counterparts and requires better control mechanisms.

4 Results & Future Refinements

The dynamic woodwind reed model presented here has been successfully implemented in digital waveguide woodwind instrument models and produces realistic transient and steady-state behaviors. Time-varying control of the reed parameters (mass, spring constant, and damping) is being explored. In particular, it is desired that the reed stiffness be variable over the course of the reed's motion. This is currently possible using three look-up tables for the filter coefficients, though more efficient methods for time-varying control are desired. The brass instrument model has been implemented only in *Matlab*, though similar real-time models by Perry R. Cook have previously been demonstrated. Current work is underway to implement all models in a real-time environment.

References

Backus, J. (1963). Small-vibration theory of the clarinet. *The Journal of the Acoustical Society of America*, 35(3), 305 313.

Cook, P. R. (1991*a*). Non-linear periodic prediction for on-line identification of oscillator characteristics in woodwind instruments. In (ICMC 1991), pp. 157 160.

Cook, P. R. (1991*b*). Tbone: An interactive waveguide brass instrument synthesis workbench for the NeXT machine. In (ICMC 1991), pp. 297 299.

Cook, P. R. (1992). A meta-wind-instrument physical model, and a meta-controller for real time performance control. In *Proceedings of the 1992 International Computer Music Conference*, pp. 273 276, San Jose, California.

Fletcher, N. H. & Rossing, T. D. (1991). *The Physics of Musical Instruments*. Springer-Verlag, New York.

Gilbert, J., Meynial, X., & Kergomard, J. (1990). The influence of the reed and the reed table on the fundamental frequency of a single-reed system. In *Journal de Physique II*, number C2 in Supplément au Journal de physique, pp. 833 836.

Hirschberg, A., Dane, H. J., Houtsma, A. J. M., Kruijswijk, S. G., van de Laar, R. W. A., Marrou-Maurières, J. P., & Wijnands, P. J. (1990). A quasi-stationary model of air flow in the reed channel of single-reed woodwind instruments. *Acustica*, 70(2), 146 154.

Hirschman, S. E. (1991). *Digital Waveguide Modeling and Simulation of Reed Woodwind Instruments*. Ph.D. thesis, Department of Electrical Engineering, Stanford University.

ICMC (1991). *Proceedings of the 1991 International Computer Music Conference*, Montreal, Canada.

McIntyre, M. E., Schumacher, R. T., & Woodhouse, J. (1983). On the oscillations of musical instruments. *The Journal of the Acoustical Society of America*, 74(5), 1325 1345.

Smith, J. O. (1986). Efficient simulation of the reed-bore and bow-string mechanisms. In *Proceedings of the 1986 International Computer Music Conference*, pp. 275 280, The Hague, Netherlands.

MUSICAL SCORE RECOGNITION OF "DON CUCO EL GUAPO" PIANIST ROBOT

A. Pedroza, H. Vargas, D. Vera and R. Fournier

Dept. of Microelectronics, Universidad Autónoma de Puebla
Boulevard Valsequillo y 14 sur A.P. J–32 Sn Manuel–72570
Puebla, Pue. México

Abstract

Don Cuco el Guapo is the first Mexican pianist robot, which was designed and built at the Departament of Microelectronics of the UAP. The project was based on multidisciplinary participation, where physicists, electronic engineers, computer scientists, musicians and designers converged.

The musical score recognition sytem was implemented through the following steps: frame grabbing, image processing, pattern recognition and interpretation or analysis of scene. The vision system of Don Cuco el Guapo is capable of reading musical score from a template.

Frame Grabbing

Frame grabbing is the process through which a visual image is taken from the three dimensional world. The frame grabbing involves different methods in order to reduce the graphic complexity, increasing the necessary information for object detection and extraction. These methods consist in the precise definition of the object to be captured, that is, what form characteristics does our object have so that the camera set up (focal distance, iris opening and focus) establishes a correspondence between the object (real image) and the plane image (digital image).

An ELECTRIM EDC−1000 camera was used for frame grabbing; it´s main characteristics include:

- CCD sensor
- High sensitivity
- Distorsionless image
- Fast response
- Resolution 192(h)x165(v)
- Monochromatic 8 bits
- Spectral range 400−1000 nm

Focal length was taken at one meter, with a variable iris for different illmination conditions. The visual information is converted to electric signal by the sensor CCD. When these signals are sampled and quantized, we obtain a digital image.

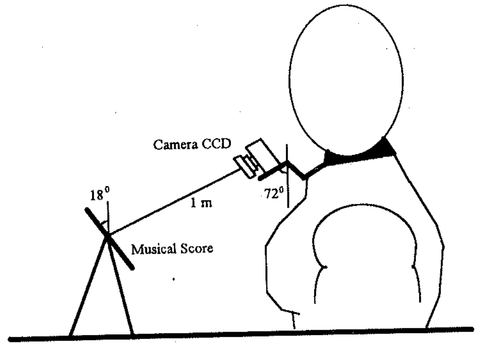


Fig 1: Focal length

The digital image can be represented by the following matrix:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N-1) \\ f(1,0) & f(1,1) & \dots & f(1, N-1) \\ \dots & \dots & \dots & \dots \\ f(M,0) & f(M,1) & \dots & f(M, N-1) \end{bmatrix}$$

where x and y are discreet variables. Each element of the matrix is called a pixel. For our case the dimension of the matrix is M=165 and N=192.

(a) Musical score format

Don Cuco el Guapo can reach two octaves as seen on the keyboard shown below, along with their representation on the pentagram. From this figure it is possible to observe that a notes position on the pentagram determines a corresponding key, and also the note determines the duration for a given beat.

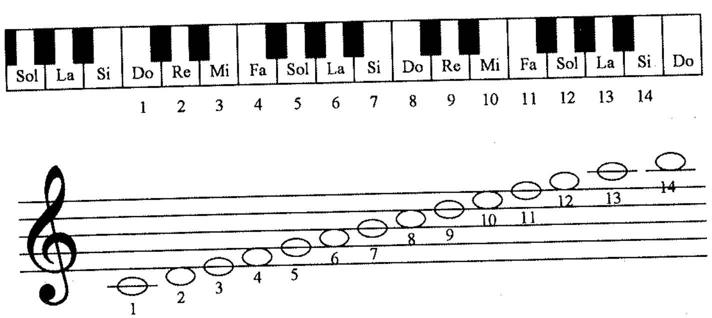
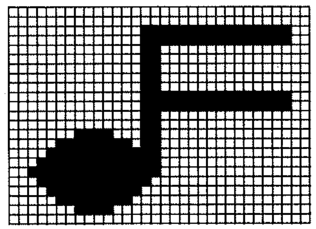


Fig 2: Relation between the keyboard and the working space for each pianist robot arm.

In order to define a musical format that could be adapted to the cameras field of view, the following points were considered:

- Focal length was set equal to 1 meter.
- A field of view equal to 16.4 cm.
- A digital image with 165x192 pixels.
- Each column of the musical score has 14 entries.

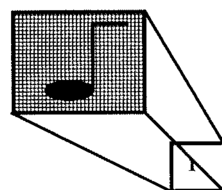
All symbols will be defined on the following grid:



23 x 32 (rows x column)

this arrangement provides sufficient definition for later recognition; with consideration having been made of information loss due to image capture and/or processing.

With the chosen parameters, we'll have 6 symbols for each row and 7 symbols for each column, distribuited thruought the field of view. The following figure shows three columns of a beat, and their relation to keyboard positions.



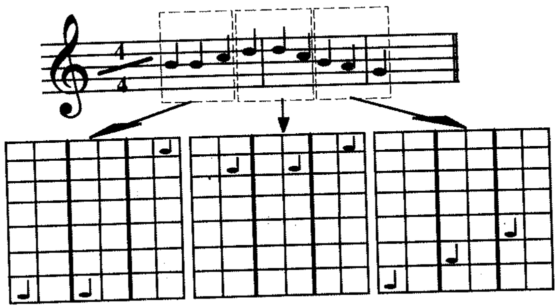
1	8	1	8	1	8
2	9	2	9	2	9
3	10	3	10	3	10
4	11	4	11	4	11
5	12	5	12	5	12
6	13	6	13	6	13
7	14	7	14	7	14

Field of view

Fig 3: The lines that divide the grid are imaginary. The frame that encloses the field of view acts as a reference for camera perception.

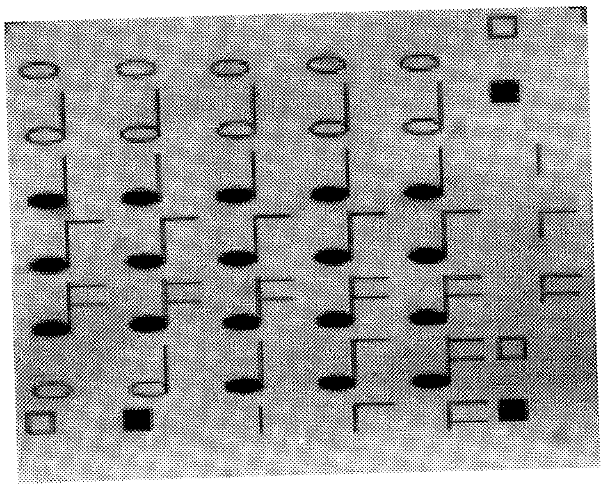
The following is an example of how the process is implemented.

Standard musical score



Musical score adapted to Don Cuco el Guapo's field of view

(b) Example of an image for frame grabbing



This image shows all the symbols allowed by the system, along with their keyboard positions. This image will be the same one used as input to all image processing algorithms.

Image Processing

Image processing is concerned with images generated from existing ones. The new image is the result of applying operations to reduce noise and other artifacts, that may be present as a result of sampling or perturbations in the system.

After analyzing different image processing methods, the Laplace function was chosen for noise reduction. The Laplacian is a second order operator defined as follows

$$L[\,f(x, y)\,] = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial x^2}$$

For digital images the Laplacian is defined as

$$L[\,f(x, y)\,] = [\,f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)\,] - 4 f(x, y)$$

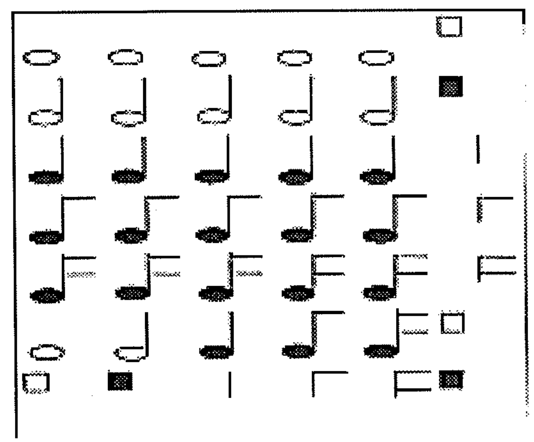
This digital Laplace formula yields zero in zones of constant intensity and in edge ramps, determining if a pixel is on the dark or illuminted side of the edge. Consequently, the Laplacian is used for intensity transitions, and rarely for edge detection.

Equation evaluation may be made using the array shown in fig. 4

0	1	0
1	-4	1
0	1	0

Fig 4: Array used to find the Laplace operator.

The following image results from applying the Laplacian to the input image, being the algorithm that resulted in maximum noise reduction.



Laplace: neighboring surroundings of 3x3.

For our case, a grayscale image is converted to a binary one, by using a non-negative threshold T, where this variable will separate the values in the equation, as follows

$$g(x, y) = \begin{cases} 1 & \text{si} \quad L[f(x, y)] > T \\ 0 & \text{si} \quad L[f(x, y)] \leq T \end{cases}$$

Thus, this equation may be seen as a procedure that extracts only those pixels characterized by significant intensity transitions (as set by T). This new binary image will be the entry data for the recognition algorithm.

Pattern Recognition

By Pattern Recognition we understand a process by which it's possible to determine the importance of each feature described in a certain object or phenomenon with respect to it's own characterization, and also to a given class.

(a) Segmentation

The image dimensions are 165x192 pixels (rows x columns), and we divide the image in 6 columns by 7 rows, resulting in a submatrix of 23x32 pixels (rows x columns), leaving 4 rows unnoccupied.

The submatrix is well defined, and it´s area is given by

$$\sum_{i=0}^{n} \sum_{j=0}^{m} A_{ij} \quad \text{where } m = 32 \text{ y } n = 23$$

Evaluation of this submatrix yields the existence or absence of an object in the field of vision of the input grid.

The criterion for determining the existence or absence of an object is given by

$$f(A_{ij}) = \begin{cases} 1 & \text{if} \quad \delta \leq \sum_{i=0}^{n} \sum_{j=0}^{m} A_{ij} \leq \varepsilon \\ 0 & \text{if} \quad \delta > \sum_{i=0}^{n} \sum_{j=0}^{m} A_{ij} \\ -1 & \text{if} \quad \sum_{i=0}^{n} \sum_{j=0}^{m} A_{ij} > \varepsilon \end{cases}$$

where n=23 and m=32. δ and ε are presence estimators, where d is least and e is most. If the decision function results equal to one, this means a symbol exists as input for that area in the field of vision; if it's equal to 0, this means there is no input, and when it's equal to -1 the input is undefined for the system symbols.

The decision function can determine which note on the pentagram is represented by the object from it's position on the field of view (or on the corresponding grid); thus, if a symbol exists, we determine the note which is it's first parameter (key number). Note duration, which is it's second parameter, will be obtained by the method of description and recognition.

(b) Description

Once segmented, and considering that a symbol was found on an image grid square, then the next step is to make a symbol description, that is, to obtain all the characteristics that define it. Having determined these parameters, we procede with the recognition.

A solution is obtained by dividing the matrix from the corresponding grid, which contains the object, into submatrices of different dimensions (as shown in fig. 5), and which will represent the characteristic features of the object.

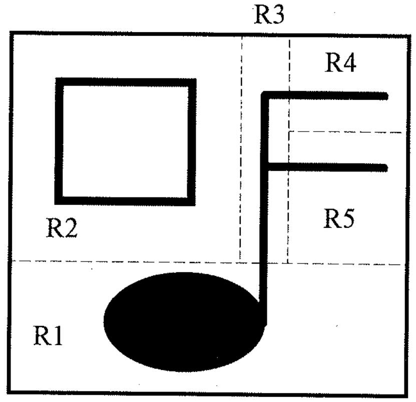


Fig 5: The grid is divided in different submatrices, each one representing an object feature or characteristic. This diagram shows two musical symbols that give an idea of how the different features are discriminated.

All the features for the description procedure have been defined, and they are given below:

- R1 represents the elipse
- R2 represents the round silence or white
- R3 represents the escrow of the symbol
- R4 represents a quaver of the symbol
- R5 represents a quaver of the symbol
- R6 indicates if the elipse is white or black

The result of the description procedure is a vector, which will contain as it's inputs the features of the object, this is

$$\mathbf{v} = (R1, R2, R3, R4, R5, R6)$$

this vector is generated by the following boolean function

$$f(D_i) = R_i = \begin{cases} 1 & \text{if} \quad \delta_i \leq \sum_{i=0}^{n} \sum_{j=0}^{m} Dij \leq \varepsilon_i \\ 0 & \text{other case} \end{cases}$$

where D_i coresponds to the submatrix of the ith feature, i = 0, 1, 2, 3, 4, 5. The submatrix dimensions are given below

For D1, n = 32 and m = 10 For D2, n = 12 and m = 13
For D3, n = 8 and m = 13 For D4, n = 12 and m = 7
For D5, n = 12 and m = 7 For D6, verifies if its white or black

ε_i is a tolerance constant. If the submatrix of the evaluated feature in the function results equal to 1, this means that the object has that feature, if the opposite is true and the result is equal to 0, this means that the object does not have the feature. Then, the description vector will be given by

$$\mathbf{v} = (f(D_0), f(D_1), f(D_2), f(D_3), f(D_4), f(D_5))$$

this vector only yields information for the features found in the object. The decision if the vector has the characteristics of a symbol defined for the system, is given by the recognition algorithm. Consequently, this vector will be the input data for the recognition algorithm.

(c) Recognition

The vector obtained in the preceding section is evaluated in table 1; and will decide if the vector information corresponds to a system defined symbol, ie

1 2 3 4 5 6 7 8 9 10

Fig 5: Sytem defined symbol

Table 1: Object properties.

TOP	R1	R2	R3	R4	R5	R6	VALOR
1	1	0	0	0	0	0	32
2	1	0	1	0	0	0	40
3	1	0	1	0	0	1	41
4	1	0	1	1	0	1	45
5	1	0	1	1	1	1	47
6	0	1	0	0.	0	0	16
7	0	1	0	0	0	1	17
8	0	0	1	0	0	0	8
9	0	0	1	1	0	0	12
10	0	0	1	1	1	0	14

It can be clearly observed that all objects have different inputs for their features. The VALUE column represents in decimal code, the value of the object row, and this value will determine if an object belongs to this class or not.

Solution:

Let $\mathbf{v}$ = (R1, R2, R3, R4, R5, R6) the vector that contains all the features of the object. Let f be a function that converts the vector in decimal representation, defined as

$$f(\mathbf{v}) = R_1R_2R_3R_4R_5R_{6\,\mathbf{b}} = n$$

where

$$n \in S \subset N$$

and

$$R_i = \{ 0 \text{ or } 1 \} \quad i = 1, 2, 3, 4, 5, 6.$$

Now, to decide if the object belongs to the class of the musical symbols, use can be made of the decision function

$$g(n) \rightarrow VALUE_i$$

where VALUEi is a value from table 1, this means that if the number n evaluated from function g is equal to one of the values of the table, the object is in the class, otherwise the object does not belong to the class.

Consequently, from this last function we determine the second symbol parameter: note duration. At this point, we recognize the symbol with all it's characteristics.

Also, at this stage, the mechanical limits of the pianist robot are verified; that is, if a chord goes beyond an interval of five adjacent notes, an error is generated, allowing the user to make any necessary corrections.

As the musical symbols are recognized, a doubly linked structure is generated, which will contain the information corresponding to the note or chord, and will be stored in a file.

Interpretation

The performance of an artificial vision system is determined by its capacity to extract meaningfull information from a scene with a wide margin of conditions. Thus, the interpretation, includes all those methods that are related to scene comprehension. In this way, interpretation associates a meaning or an action to a set of recognized objects.

In the pianist robot's vision system, the interpretation algorithm will understand all the musical file, up to the the start of robot's control system, for the the code execution. In the interpretation process , note digitization is made first, that is, notes are assigned to the corresponding fingers and optimization is made of the robot's arm movements. This new code is then translated to another one which will be delivered to the interface.

After this the whole system is activated, and the robot produces computer music, by playing the keyboard with a mechanical sensitivity. In the robot's presentations, science, art and technology converge.

Example of musical score for "Don Cuco el Guapo" pianist robot

Recordando a Bach

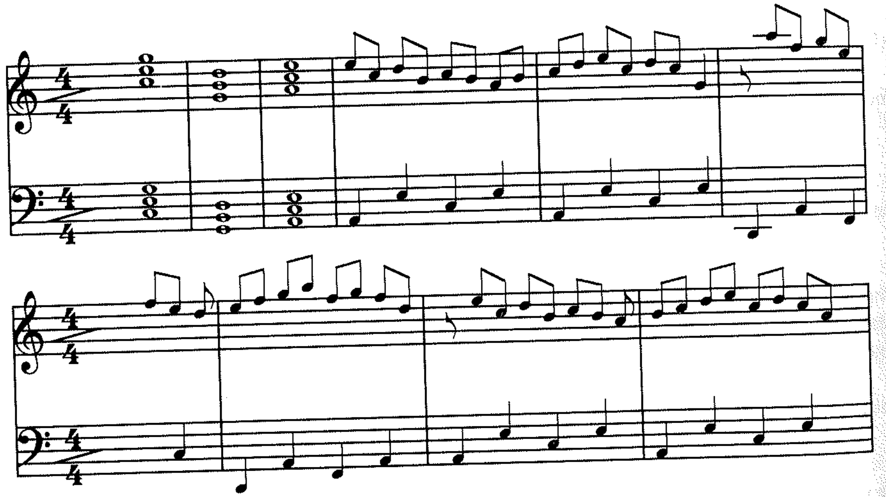
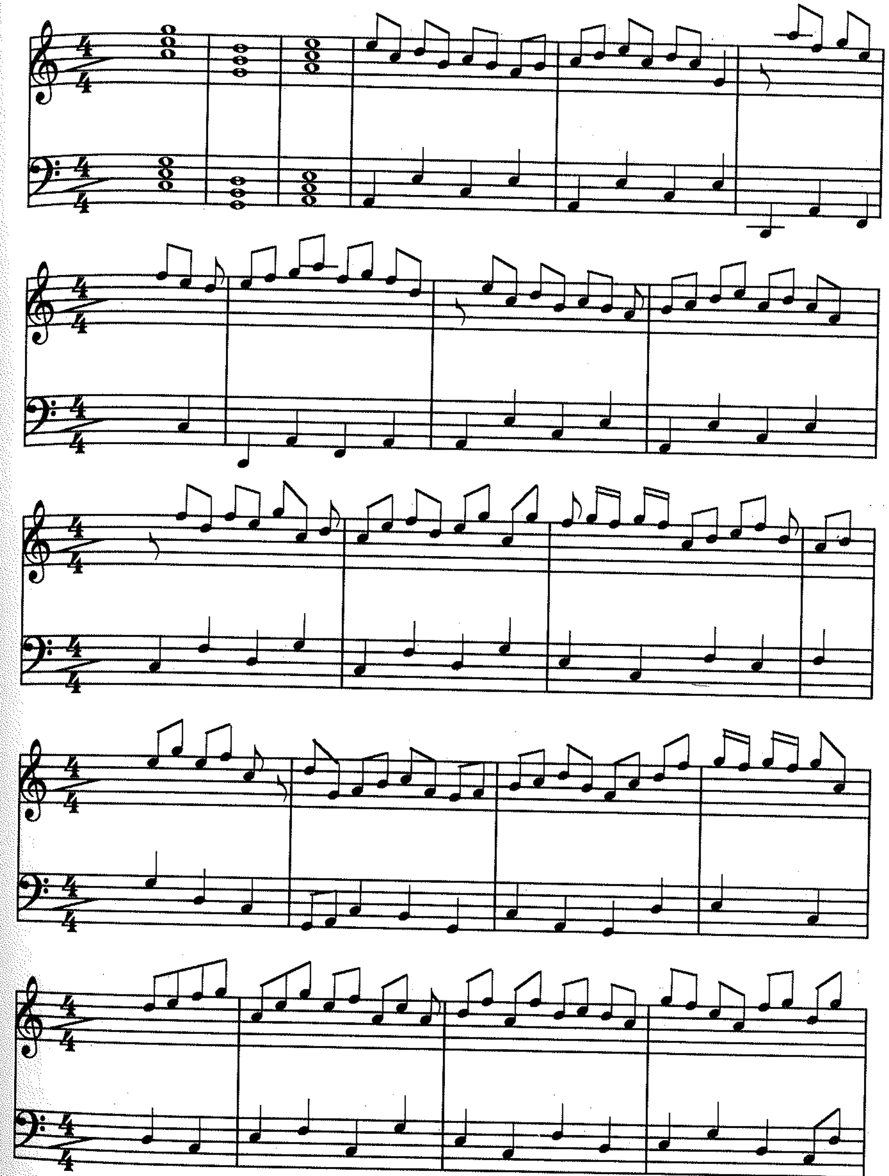
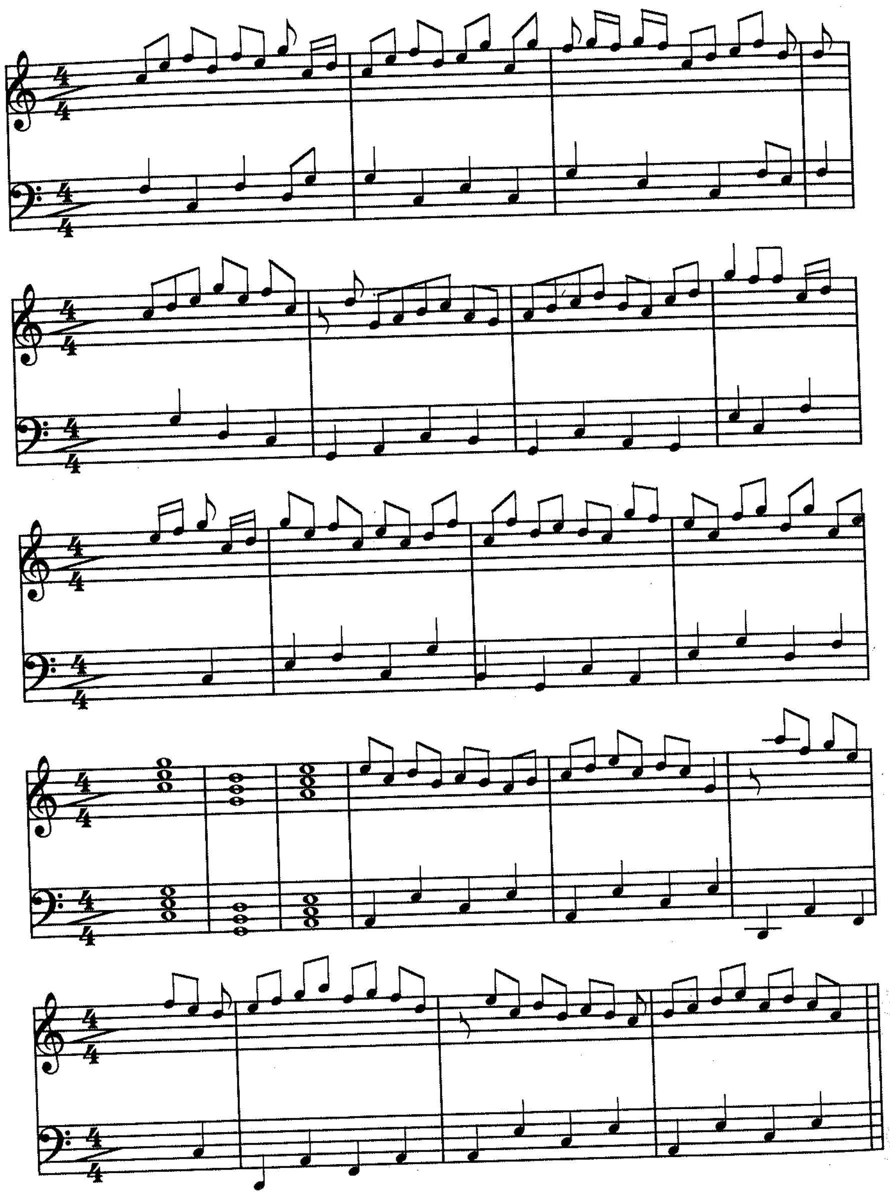
Alejandro Pedroza Meléndez

Bibliography

Image Forming Optics - Instrumentals Optics
by Simclair, Douglas.
Image Processing and Computer - aided design in electronics optics
P.W. Hawkes. Academic Press 1973
Image Analysis and Mathematical Morphology
Serra J. Academic Press (London)
Introductory computer Vision and Image Processing
Adrian Low
Otsu, N. (1979) A threshold selection method from grey-level histogram,
IEEE Transaction Systems, Man and Cybernetics, 9(1), 62-66 January.
Advances in Digital Image Processing
Theory, Application, Implementation
Edited by Peter Stucki
Machine - Aided
Image Analysis 1978
Institute of Physics Conference Series Number 44
Pattern Recognition Introduction and Fundations
by Sklansky Jack
Pattern Recognition Principles
by Tou, J.T. & Gonzáles, R.C.
Pattern Classification and Scena Analysis
Duda, Richard O. & Hart, Peter E., John Wiley & Sons
Inc. 1973
Modelos Matemáticos para el Reconocimiento de Patrones
José Ruiz Shulcloper
Instituto de Cibernética, Matemática y Física
Academia de Ciencias de Cuba
Robot Vision
The MIT Electrical Engineering and Computer Science Series
Berthold Klaus, Paul Horn
Fundamentals for Control of Robotics Manipulators
Antti J. Koivo
Robotica: Control, Detección, Visión e Inteligencia
K.S. Fu, R.C. González, C.S.G. Lee
A. Bazerghi, A. A. Goldenberg, J. Apkarian
"An Exact Kinematic Model of PUMA 600 Manipulator"
IEEE Transactions on Systems, Man, and Cybernetics,
Vol. SM-14, No. 3, pp 483 - 487, May - June 1984
C.P. Neuman and J.J. Murray
"Symbolically Efficient Formulations for Computational Robot Dynamics"
Journal of Robotic Systems, Vol. 4, No. 6, pp 743 - 769 1987.
M.W. Walker and D.E. Orin

"Efficient Dynamic Computer Simulations of Robotic Mechanisms"
Journal of Dynamical System, Measurments and Control, Vol. 104,
pp 205 - 211, 1982.
DC Motors, Speed Control Servo Systems
An Engineering Handbook
prepared by Electro - Craft Corporation, USA

MIDISCAN

the program for reading and processing musical notation

WLADYSLAW HOMENDA(*)(**), ALFREDO CRISTOBAL(**), MERCEDES ARELLANO(**)

Department of Electronics and Telecommunication, Applied Physics Division
Centro de Investigacion Cientifica y de Education Superior de Ensenada
Km. 107 Carretera Tijuana/Ensenada, Ensenada, B.C., Mexico 22860

ABSTRACT

The paper presents problems related to automated recognition of printed music notation. Music notation recognition is a challenging problem in both fields: pattern recognition and knowledge representation. Music notation symbols, though well characterized by their features, are arranged in elaborated way in real music notation, which makes recognition task very difficult and still open for new ideas. On the other hand, the aim of the system, i.e. application of acquired printed music into further processing requires special representation of music data. Due to complexity of music nature and music notation, music representation is one of the key issue in music notation recognition and music processing. The problems of pattern recognition and knowledge representation in context of music processing are discussed in this paper. MIDISCAN, the computer system for music notation recognition and music processing, is presented.

Keywords: music notation recognition, knowledge representation, music representation, MIDI format.

1. INTRODUCTION

There are many ways in which computers have been involved in the world of music. One is the score edition, where musicians can develop a score using an editor to produce a digital output file. This strategy might be a good idea, but it is a non-traditional way to develop a score. Another computer application is music processing that can help musicians in music creation process: automatic composition of music, analysis of musical style and so on.

(*) ***On leave from:*** Institute of Mathemaics, Warsaw University of Technology, Warsaw, Poland

(**) ***Correspondence to:*** Wladyslaw Homenda, Mercedes Arellano, Alfredo Cristobal,
- CICESE Research Center / Applied Physics, P.O.Box 434944 San Diego, CA 92143-4944, USA
  e-mails to: {homenda,arellano,asalas}@cicese.mx
- Wladyslaw Homenda, Instytut Matematyki, Politechnika Warszawska, pl. Politechniki 1, 00-661 Warszawa, Poland
  e-mails to: homenda@plwatu21.bitnet

In this paper we are going to deal with automatic recognition of printed music and its conversion to some digital forms. Here, musicians can continue developing the scores as they used to, and introduce these printed scores into a computer to convert them to a digital output file.

Over the last years, efforts have been focused on automatic musical notation recognition. Several research centers and universities have developed many programs to recognize printed music, c.f. (Computing 1994, Fujinaga, 1988, Itagaki, et al., 1992, Kato et al., 1992). Although some results of these works are quite good, improvement is needed. The obstacles that developers of these systems have had to face seem to have increased and most of them are too complex to be solved by classical methods in comparably short execution time of the system. To show this fact, a couple of the most important problems are listed next:

1. Problems related with the nature of music represented in printed form: here composers feel free to invent new symbols or contractions of them and this freedom has increased the number of musical dialects around the world, most of which are not standardized.
2. Problems related with recognition technology: in this area it is common to find symbols with different sizes and shapes, symbols that overlap one over the other, distortions of the image, etc.

These difficulties have not permitted us to make a universal printed music recognizer system. So, this area is still open to ideas that can improve the methods already applied or create new ones to have a better recognition process.

Some problems mentioned above were solved by MIDISCAN. MIDISCAN is a semi-automatic printed music recognition system which accepts scanned scores and produces a playable file (MIDI file). Such file can be played on synthesizers, computers, and other equipment that can accept MIDI files. Unfortunately, the conversion is not so direct, because of that MIDISCAN represents the recognized scores in an intermediate format called MNOD, cf. (Homenda, 1995).

2. MUSIC NOTATION RECOGNITION - PROBLEM STATEMENT

Music notation can be interpreted as a language allowing to document musical information in a legible, archival form. Recognition of music notation can of course be modeled as mappings between the printed notation and the information it represents, cf. Figure 1, cf. (Blostain & Baird, 1992). This general formulation of the task of music notation recognition does not reflects conceptional and technical problems developers of recognition systems are faced.

First of all, music notation does not have a universal definition. Although attempts to codify printing standard for music notation have been undertaken, c.f. (Ross, 1970, Stone 1980), in practice composers and publishers feel free to adopt different rules and invent new ones. Though most of scores keep the standard, they still can vary in details. Moreover, music notation, as a subject of human creative activity, constantly develops and probably will be unrestricted by any codified set of rules. Thus, it may not be possible to built a universal recognition system accepting all dialects of printed music notation. Furthermore, the nature and structure of music, even that printed one, is much more complicated than the structure of a text, so representation of music is comparably much more difficult than, for example, representation of printed text. As the result, comparing music notation recognition with text recognition, there are several applicable computer systems for automated text recognition with considerably high rate of recognition (which can be calculated as ratio of recognized characters to all characters in the text). As to music notation recognition, commercial systems are still very rare despite the fact that several research systems of music notation recognition have already been developed, c.f. (Blostain & Baird, 1992, Fujinaga, 1988, Itagaki, et al., 1992, Kato et al., 1992).

On the other hand, it is not quite clear, what *music information* - the goal of recognition task means, cf. Figure 1. In fact, meaning of this term may depend on the application of recognition task. For some applications, e.g. from musician point of view, a complete solution to the music recognition problem is the specification of: which notes are present, what order they are played in, their time values or durations, and volume, tempo, and interpretation. Thus, in this case the scanning units are generally quite small, for example, a clef, one or more key signature, accidentals, a time signature, note heads in a chord, and following music symbols are not included in scanning units: staff lines, beams, slurs ties brackets, text, and crescendo or

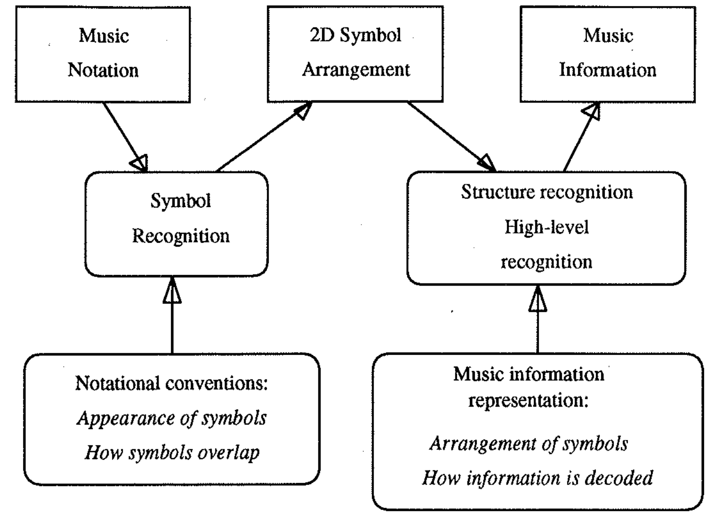


Figure 1. Music notation recognition

decrescendo signs. However, this set of information is not sufficient for producing parts from a score. As it was stated in (Blostain & Baird, 1992), these digital files can be used for many purposes:

1. To adapt existing works to other instrumentations.
2. To convert existing scores to Braille to aid blind musicians.
3. To read works in old editions and produce a new printing.
4. To store scores in digital formats and construct a kind of music library which one can see and hear the music with multimedia equipment, or even change the instruments and features of the score.
5. To print newly written music automatically.
6. To analyze musical structures and styles.

Finally, it should be stated, that designing and developing universal recognition system seems to be currently beyond the horizon. A reasonable strategy for developing recognition system should be based on the application of such a system. This assumption would be useful in restriction both, the class of acceptable notation dialects as well as the format of music information acquired. For example, if recognition of old notations and producing a new printings is considered as a goal of recognition task, it will be effective in fixing the set of rules defining both, old and new notations.

3. MIDISCAN - RECOGNITION PROGRAM

3.1. Overview of the program

The transformation of data given as printed music into playable MIDI format is the main idea of MIDISCAN software. This transformation is intended to be as far automated as possible. Unfortunately, at present stage of development of both fields: methods of recognition of printed music notation and representation of music data, it is impossible to built fully automated system which could recognize music notation and create playable music data correctly performed with the electronic instrument. Errors appearing in recognition process, even a few of them, cause that correction of recognized music notation is necessary

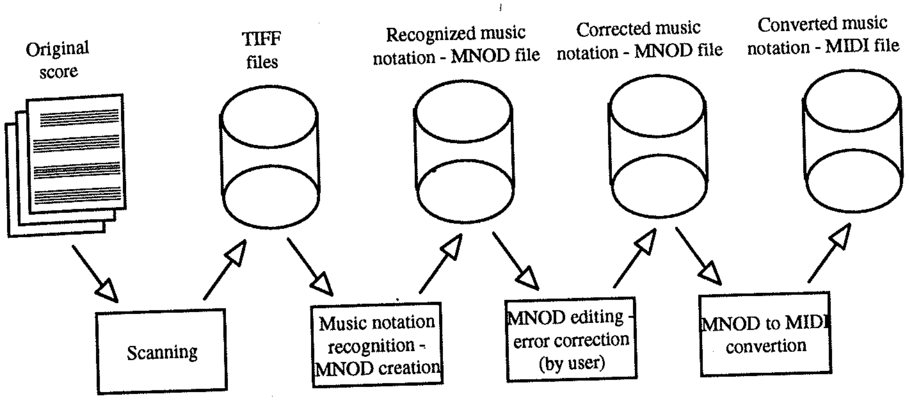


Figure 1.

before it is performed with the instrument. Thus, a correction of acquired music data is necessary. The correction may be done on output, playable data (e.g. MIDI format) or in the middle of the road: before acquired data are converted to playable format.

The first option, correction of a final MIDI format, is regarded as to be less convenient than correction of recognized data before MIDI conversion. The reason is quite clear, for example, key signature correction needs only a few operations before MIDI conversion while pitches of many notes should be corrected if wrong key signature was assumed in MIDI conversion. But this assumption requires the music notation to be represented in a format allowing for editing, correction and, then, MIDI conversion. This assumption implied the necessity of acquired data to be represented in the special intermediate format called MNOD (Music Notation Object Description).

The idea of MIDISCAN is outlined in Figure 1.

Once the score is scanned and stored on a hard disk as a set of TIFF files, MIDISCAN can start its task, i.e. recognition of music notation, writing acquired data as a MNOD file format, editing MNOD format and presenting it to the user for corrections (if necessary), conversion of MNOD file format to MIDI file format.

3.2. Score definition.

Two types of music notation may be processed by MIDISCAN: ensemble and part scores. MIDISCAN does not detect the type of a processed score, it must be defined by user. In case of ensemble score, i.e. score with all voices linked together into systems, user only chooses the score type and defines the sequence of pages of the score (i.e. pages of the original notation scanned and stored in the form of TIFF files). When part score is processed, i.e. score with voices separated from each other, the number of voices and the number of pages for every voice must be described before the sequence of TIFF files is defined. The sequence of pages must keep the order of respective pages of the score.

3.3. System location.

Automated stave location is performed for the whole score before recognition process is started. Simultaneously, the structure of the score is detected, i.e. the way systems are located. The number of staves in every system is detected for the whole score of ensemble type and for every part of the score of part type. Let us recall that the term "system" is used here in the meaning of:

- all staves performed simultaneously and
- joined together in the score or part of the score.

The score for voice and piano with three staves in the system is an example of ensemble type of score. If voice part is missing in the first and second system, the score has two irregular systems with two staves and the other systems are regular with three staves.

Part type score: only scores with constant number of staves in systems for every part are accepted. Up to 3 staves per system are permitted (e.g. organ part of the score consists of 3 staves, other voices consist of no more staves). A score for string quartet with separated voices is an example of part type score with four parts and one stave in part system for every part.

3.4. Stave location.

Stave location algorithms are based on horizontal projections. Theoretically, for non-distorted and non-skewed images, methods based on projections should be fully effective: high pass filtering gives clear image of a stave as five equidistant picks with the height equal to the length of the stave.

Unfortunately, in real images staff lines are distorted too much to give so clear projections, especially when projection is done for page width or even for wide region. Scanned image of a sheet of music is often skewed, staff line thickness differs for different lines and different parts of stave, staff lines are not equidistant and are often curved, especially in both endings of the stave, for ensemble type of score staves may have different sizes, etc. These problems cause that projections done in wide region are useless for stave location.

On the other hand, projections in narrow region are distorted by notation objects such as ledger lines, dynamic 'hairpin' markings, slurs, note heads, etc. Thus, simple filtering does not give information sufficient for stave location. In MIDISCAN program, horizontal projections in several narrow regions are analyzed for obtaining vertical placement of the staves. Once vertical placement of the stave is located, both endings of it are detected. Projections in relatively narrow regions are used in stave endings location task. Iterative process based on classical bisection algorithm is employed in this process - the process starts in the middle of the stave and then goes in the directions of both endings of the stave.

An advantage of applied methods is that distortions such as non-equidistant staff lines, varying thickness of staff lines, skewed images (skew angle up to 10-15 degrees) and stave curvature, do not influence the process of stave location as well as the process of notation recognition in an observable way.

It is worth mentioning that stave location process is not fully automatic. In some cases, especially for low quality images or for very dense notations, automatic stave location is mistaken and must be corrected manually. Fortunately, the program is able to detect problems it has and only if automated correction of given problem is not possible, location process is suspended unless user fixes the problem.

3.5. Recognition.

Music notation is built around staves. The position and size of symbols are restricted and determined by the stave. So, location and identification of staves must be the first stage of recognition process. Having staves and systems located, the program starts fully automated recognition of music. Recognition is done for every stave, and then, after notation is recognized and analyzed for given stave, the acquired music data are filled into MNOD format.

Recognition strategy can be seen as three step process:

- object location,
- feature extraction
- classification.

The first step - object location - is aimed at preparing a list of located symbols of music notation. Bounding boxes embodying symbols to be recognized are defined for located symbols. The process of object location is based on projection analysis. First, the projection of whole region of given stave on OX axis is processed. The location process is mainly based on the analysis of a derivative the projection, c.f. (Fujinaga, 1988,). The derivative analysis gives the first approximation of object location. Then, for every rough located object, a projection on OY axis is analyzed to obtain vertical location of the object and to improve its horizontal location. The most important difficulties are related to objects which cannot be separated by horizontal and vertical projections. Also wide objects as slurs, dynamic 'hairpin' signs, etc. are hardly located.

The next two steps of recognition process are based on the list of located objects. both steps: feature extraction and classification overlap each other and it is not possible to separate them. Feature extraction starts from extracting the simplest and most obvious features as height and width of the bounding box containing given object. Examining of such simple features allows for classification only in a few cases. In most cases additional features must be extracted and context analysis must be done. The extraction of features is based on filtering of projections in the bounding box, analysis of chosen columns and rows of pixels, etc. Several classification methods are applied for final classification of object including context analysis, decision trees, and syntactical methods.

Only limited set of music notation objects can be processed in MIDISCAN. This set includes notes, chords, rests, accidentals, clefs, bar lines, ties, key signatures, time signatures, change of key and time signature. Rhythmic grouping can also be inserted into acquired music data, though they are not recognized. Other symbols are going to be recognized and represented in future versions of the program.

Recognition confidence depends on many features of a printed score: font, printing quality, image quality, notation density, etc. The obvious, general rule may be formulated that the higher quality of printed music and scanned image, the higher the rate of recognition.

Recognition efficiency of MIDISCAN program may be estimated as 95% for good quality of image and 80-85% for low quality of image. Precise calculation of recognition rate is strictly related to applied calculation method (c.f. note in chapter 1.), but the scope of the paper does not allow to discuss extensively this problem.

4. MUSIC REPRESENTATION

Making an analogy between computer processing of a printed text and a printed music, it is worth underlining that, as to text processing, design and implementation of widely accepted data representation is considerably easy. Rich Text Format (RTF) format is an examples of such a representation.

Music data representation is far more difficult and, up to now, there is no universal representation widely used and commonly accepted. Music data formats used in computer systems are intended more for particular task rather than for common use. Even if a particular format is widely spread and commonly applied, it is used for special tasks rather than for any purpose. For example, "MIDI (Music Instrument Digital Interface) data format was established as a hardware and software specification which would make it possible to exchange information between different musical instruments or other devices such as sequencers, computers, lighting controllers, mixers, etc. This ability to transmit and receive data was originally conceived for live performances, although subsequent developments have had enormous impact in recording studios, audio and video production, and composition environments" (c.f. (MIDI, 1990)). Nevertheless, MIDI format, as performance oriented, is not an universal one. E.g., it is very difficult or even impossible to represent in MIDI format graphical features related to music notation. On the other hand, format used by notation programs are notation oriented and cannot be easily used as universal format of music representation.

Because recognition confidence in MIDISCAN is not satisfactory enough for direct conversion of recognized music notation into MIDI format, it is necessary to edit acquired data, correct it and then convert into MIDI format. These tasks need acquired data to be stored in some form suitable for both editing and conversion. For this particular aim, a special format was developed. This format is called MNOD format (Music Notation Object Description). It plays the role of an intermediate format between printed music notation and playable MIDI format.

It was assumed that recognized music notation would be edited for correction in the form compatible with original score. This assumption allows for simultaneous displaying of original score and acquired data. It makes editing and checking correctness of acquired data as easy as comparing two notations which should be identical. All differences can be easily corrected, even if user is not familiar with music and music notation.

MNOD format applied in MIDISCAN for data representation meets all these requirements. Its main features give the possibility of data interpretation form both perspectives: notation oriented and performance oriented, that makes music data easily accessible for both purposes: editing and MIDI conversion. Unfortunately, MNOD format applied in MIDISCAN does not represent all commonly used notational symbols. Only symbols edited in MNOD editor (see section 3.5. for the list of the symbols) are represented in the format.

MNOD format is structured hierarchically. The levels in this hierarchy reflects data accessibility for above purposes. MNOD format may be regarded as two different structures permeated each other.

The notation oriented structure may be seen in the following levels:

- score,
- page,
- stave on page,
- objects of music notation

while performance oriented structured may be outlined as below:

- score,
- part of score (applicable to the scores of part type),
- system,
- stave in system,
- vertical events,
- objects of performed music.

This comparison gives only general view on differences between those attempts. Extended discussion on this topic is out of the scope of this paper.

It is worth mentioning that levels of both structures differ in their meaning even if they are called similarly. E.g., notation structure reflects sequential organization of staves on page while performance structure organizes staves according to the systems of the score, regardless of their order on the page. Similarly, objects of music notation are seen differently in both structures. For example, in notation structure, notes must have such features as their position on the page, while their relative position to each other is unimportant. On the other hand, relative placement of notes is significant for performance structure, while their position on the page is of less importance.

The approach to music representation applied in MIDISCAN is flexible and easy to control: displaying data in graphical form on the screen, converting music data to MIDI format and independent music data processing.

Acquiring contextual information from recognized music notation and checking correctness of recognized notation is the aim of independent data processing applied in the program. This processing allows for locating of pickup/close-out measures, analyzing voice lines, verifying bar lines or change of key and time signature consistency, monitoring data integrity. In general, the possibility of independent music data processing considered in wider context creates a lot of research problems related to knowledge representation, that makes music representation interesting from more general point of view.

5. EXPERIMENTAL RESULTS

MIDISCAN was developed and ported on PC 386 and compatible computers in WINDOWS environment. Hardware requirements are similar as for WINDOWS environment. Digitizing resolution of 300 dpi is suggested for acquired binary images of the size A4. It gives TIFF file of the size approximately equal to 1MB. Both orientation of a page: portrait and landscape is accepted. Pages/files of bigger size can be effectively processed on computers with at least 8 MB RAM. Processing time, which is different for different notations, depends on resolution of the image and on notation density.

For example, it took about 75 minutes to recognize first movement of Bach's Brandenburg Concerto no. 6. The experiment was done on PC 386 / 8 MB RAM, 33 MHz clock. The score consisted of 15 pages of A4 format, 18 staves per page, 6 staves per system, scanned at 300 dpi resolution. The printing quality of the score was rather low, music notation density - medium. The recognition efficiency for this score was considerably high. Three staves were incorrectly localized and, what was important, system detected all three errors and signalized them to the user. Estimated recognition rate overheaded 90%. It was calculated as the ratio of missing or mistaken objects or their features, to all objects.

Another experiment done for Beethoven's "Fur Elise" gave similar result in speed processing (i.e. comparable recognition time per stave), but the rate of recognition was higher due to higher printing quality of tested score. All 36 staves of this score were located accurately, score structure was also correctly detected, estimated recognition rate was over 95%.

More experiments done for different scores confirmed that context information implied form music notation, such as pickup and close-out measures and voice lines analysis and location, was correctly and comparably easily acquired.

MNOD to MIDI conversion was very fast and took about one minute for Bach's score and a few seconds for that of Beethoven's.

To compare similar parameters of other recognition systems see (Blostain & Baird, 1992, Fujinaga, 1988, Itagaki, et al., 1992, Kato et al., 1992)..

6. CONCLUSIONS

The paper describes MIDISCAN - a recognition system for printed music notation. Music notation recognition is a challenging problem in both fields: pattern recognition and knowledge representation. Music notation symbols, though simple and well characterized by their features, are arranged in sophisticated and confusing way in real music notation, which makes recognition task highly difficult and still open for new ideas, as for example, fuzzy sets application in skew correction and stave location. On the other hand, the aim of the system: conversion of acquired printed music into playable MIDI format, requires special representation of music data. This representation should be adequate for both: source - notation oriented, and target - performance oriented music data. Regarding further development of this system, the effort should be put on following tasks: improving recognition methods, extending class of recognized objects, improving and extending music representation format.

See also (Aikin, 1994, Computing, 1994, Homere, 1994, Lindstrom, 1994) for reviews of MIDISCAN.

REFERENCES

Aikin J. (1994): Musitek Midiscan: Sheet music recognition software, *Keyboard*, March 1994, pp. 136-140.

Blostein D. & Baird H. S. (1992), A Critical Survey of Music Image Analysis, *Structured Document Analysis, H.S.Baird, H.Bunke, K.Yamamoto (Eds)*, pp. 405-434, Springer-Verlag, 1992

Computing (1994), *Computing in Musicology*, vol. 9, 1993-1994, pp. 109-166.

Fujinaga I. (1988), Optical music recognition using projections, *Master's thesis*, McGill University, Montreal, Canada, September 1988.

Homenda W. (1995), Optical pattern recognition for printed music notation", *Proc. of the SPIE's Intern. Symp. on Optical Pattern Recognition VI*, April 1995, Orlando, Florida.

Homere S. (1994), Musitek Midiscan: Donnez la parole a vos partitions, *Keyboards Magazine*, 1994.

Itagaki T. et al. (1992), Automatic Recognition of Several Types of Music Notation, *Structured Document Analysis, H.S.Baird, H.Bunke, K.Yamamoto (Eds)*, pp. 405-434, Springer-Verlag, 1992

Kato H. & Inokuchi S. (1992), A Recognition System for Printed Piano Music Using Musical Knowledge and Constraints, *Structured Document Analysis, H.S.Baird, H.Bunke, K.Yamamoto (Eds)*, pp. 405-434, Springer-Verlag, 1992

Lindstrom B. (1994), Musitek Midiscan: Optical Music Recognition is finally a reality, *Electronic Musician*, February 1994, pp. 151-154.

MIDI (1990), *MIDI 1.0, Detailed Specification*, Document version 4.1.1, February 1990.

Ross T. (1970), *The Art of Music Engraving and Processing*, Hansen Books, Miami, 1970.

Stone K. (1980), *Music Notation in Twentieth Century: A Practical Guidebook*, W.W.Norton & Co.,New York, 1980.

A computer-aided object-oriented analysis

Didier GUIGUE
DEPARTAMENTO DE MÚSICA/CCHLA
UNIVERSIDADE FEDERAL DA PARAÍBA
Home add.:Rua Antônio P. da Rocha, 90
58045 - 380 João Pessoa - PB - BRAZIL
e-mail: didier @ brufpb.bitnet

Most of analytic tools developped during this century are exclusively devoted to pitch structure. It is however an evidence that, at least since Debussy, forming and linking complex abstract sound-objects are too decisive composing features. This paper exposes the basis of a computer-aided information and evaluation method which may bring out significant data for an objective analysis of the formal functions of these sound-objects. The Prelude La Cathédrale engloutie , by Debussy, exemplifies the way this method can contribute to high-level structure understanding.

1. The goal

Most of analytic tools developped during this century are exclusively devoted to the basic low-level components of music writing: pitches (or pitches classes). It is however an evidence that, at least since Debussy, forming and linking complex abstract sound-objects are too decisive composing features. Musicology and music theory generally approach object-oriented formal structures in a empirical way, asserting, without demonstrating, that they depend on the laws structuring pitches, considered as the most important, or, indeed, the *unique* formal dimension. This paper exposes the basis of a computer-aided method of information and evaluation of the components of the musical score which may bring out all the necessary data for an objective analysis of the formal functions of this sound-objects. Restricted — at least for the moment — to the piano literature, it is aided by a set of algorithms realized inside the *Patchwork* environment developped at the IRCAM Institute, Paris [1]. as part of a PhD research on the 20th Century piano music.

2. The score reading

The source score has to be previously segmented in a sequence of units, basically defined by an homogeneous content; this homogeneity may be mainly due to the invariance of (most of) its elements, or to continuous teleological changes. The segmentation process is reached threw a continuity break scanning of the score, at any desired level: macro-formal stops (rests, fermatas), interruptions of pedal sustains and/or phrase slurs, continuity breaks in intensities, registers, rhythmical outlines, densities, a.o. These rupture criteria do not depend on the thematical/periodical structure (phrases and other time-groupings), although coincidences may be founded, especially at higher hierarchical levels. Nevertheless, independence is fundamental to check connections between the object-level organization and lower ones. Each unit is thus a sole (written) sound-object [See fig.1 for a segmentation sample of the last bars of Debussy's La Cathédrale engloutie]. Once segmented, the score is stored in several Midi files (one per segment, i.e. one per sound-object) to be exported to Patchwork [see, in the sample *Patchwork* window fig.3, the Midi-import patches and Midi-object storage].

3. The sound-objects evaluation

Each object is analysed by a set of specific algorithmic patches, we name *interpreters*, connected in a hierarchically structured "frequency-modulation" bidimensional network [fig.2], where qualitative evaluations (in round boxes) modulate quantitative ones (square boxes), producing "synthetical" interpreters at a subsequent level.

The "S" dimension is exclusively concerned with achronic aspects of the musical writing: inside the "*SPACE* " sub-group of evaluation patches, the *AMBITUS* single interpreter calculates the range-filling rate of the source file (related to the whole range of the piano), a quantitative value; and the *REGISTER* algorithm computes the number of filled registers (we count seven acoustic-defined registers in the piano), a qualitative distribution choice of the composer [the fig.3 shows how the *SPACE* interpreters do work]. The synthetic value of this two interpreters (*SPACE* output) is then submitted to minute analysis of vertical tone-distribution modalities ("*S-FILL* " patches): distribution complexity rate (according to linear — i.e. equidistant — and/or harmonic intervallic object-construction models), perceptive and/or cognitive consonance/dissonance rate, density rate. The second-level synthetic ouput value (*SPECTRUM*) is only resynthesized if the *EXOGENICS* interpreter is active (for instance if the sustain or *una corda* pedal is used, or any global sound transformation, as in the pieces for prepared piano by John Cage). The last output value (collected in the S-box), a rate of achronic sound complexity for the source object, is only a part of the analysis of the observed segment of the score.

The "T" network value rate acts as a qualitative modulator of the S contents and properties, according to the way the composer has distributed them in the object time-span. While the *SPAN* algorithm group quantify the relative duration of the object, by comparing it to the longuest one (or the whole piece or section), the *T-FILL* group evaluates some decisive aspects of the time outlining of the sound-object: the time-density rate (the number of sequential events *versus* a maximum fullness-paradigm value), the velocity (i.e. relative intensity, or dynamics) dispersion rate (related to the most frequent observed value in the source), the linearity rate (based on a equidistant onsetting model — the most regular rythmic distribution of events), the pitch-direction (a relative real number, function of the global directionnality of pitch-profile), a.o. As for the S dimension, *T-FILL* is a quality modulator of *SPAN*, producing the final T value (collected in the T-box), which interacts with the S one to give a synthetic global sound-complexity rate for the source object [fig.2].

A whole segmented piece can be stored in a *Patchwork* sequence of buffers, as showned in fig.3 (left side, sample for 10 objects). A *Patchwork* specific set of boxes allows the connection of one or several evaluation patches, as desired. The ouput data is a numerical list for each connected interpreter. Each list correspond to the sequential evaluations of the connected sound-objects [see fig.4]. This lists contain significant information for analytic purposes, as it will be shown now to conclude.

4. The output data and the formal analysis

The figs. 5 and 6 are graphic representations of part of the lists table of the fig.4. The first graph displays the results of the evaluation of four interpreters for the last five sound-objects of La Cathédrale engloutie [see the musical score fig.1]. It shows how the composer realize the formal *kinesis* of the end of this Prelude by dialectically linking various parameters of the sound-objects shape: the progressive increase of the sound-space [see the growing evolution of range-filling and register-density rates, reaching maxima values in the three last objects], is correlated to the slowing down of the music pulse [see how the time-density rate strongly drops between object o39 and o40], the slight scarcering of the pitch-density rate [space-density rate reads (0.45, 0.32, 0.34, 0.20, 0.20) for the sequence]. It must be observed, too, that the vertical pitch-content tends to become more and more harmonic [the lowest the harmonicity rate, the most harmonic the pitch distribution; the *HARMONICITY* algorithm reads: (0.29, 0.33, 0.12, 0.00, 0.00) for the sequence].

The fig.6 shows the kind of relevant analytic data the program may return. It appears that Debussy, in this Prelude, systematically correlates the harmonic distribution of tones to the range-filling rate, in a way that wide-spaced objects tend to simulate an harmonic structure, while narrow-spaced ones have a strong inharmonic tone distribution [2]. There is no doubt that an object-oriented analysis of a significant number of similar pieces by Debussy, or other composers, has to bring out significant, objective informations about the new ways the 20th Century composers deal with the formal structuring and the function of the sound-object concept as a form-defining dimension [3].

[1] *Patchwork* is an interactive environment for computer assisted composition. It consists of a set of tools that help the composer generate and manipulate musical objects. Because of its aptitude to manage and to interact with the musical knowledge, it may allow the critical processing of information.

[2] For a more compreeensive study of the space organization in this Prelude , see GUIGUE, D. (1995): "Sonorité, espace et forme dans La Cathédrale engloutie de Debussy". São Paulo: *Revista Música*, 5 (2).

[3] The software, although still in a experimental phase, is freely available for *Patchwork* owners.

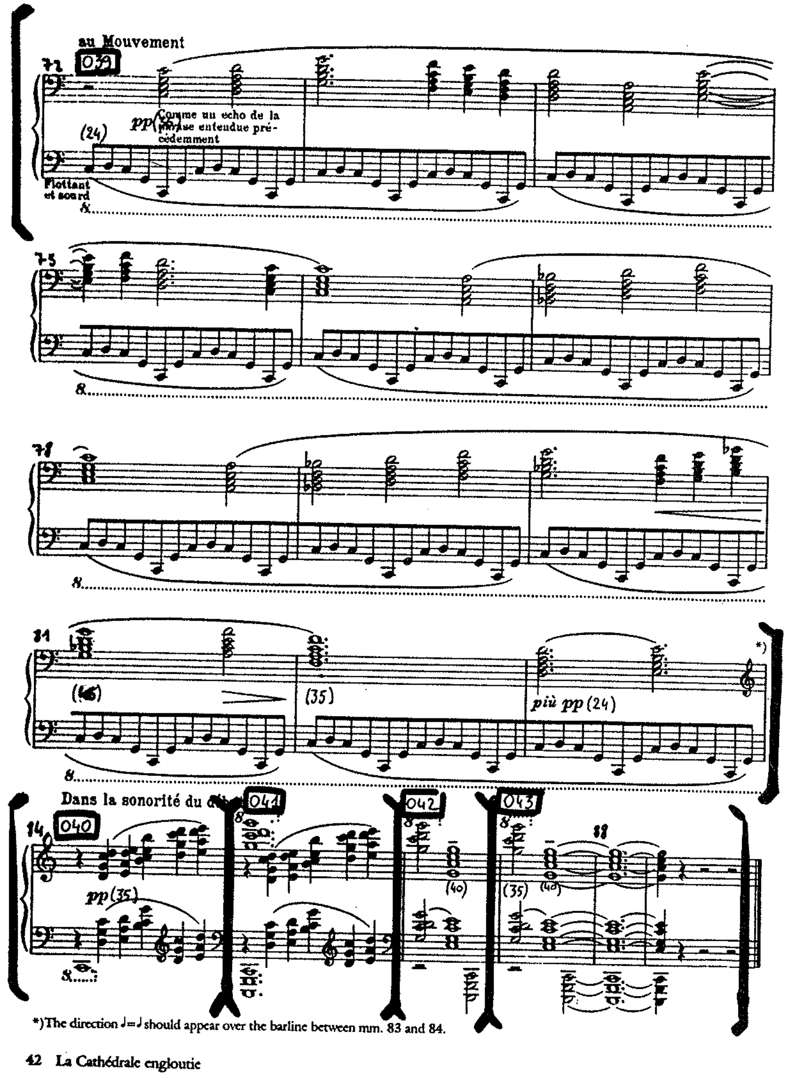


fig. 1: The final section of Debussy's La Cathédrale Engloutie, segmented in five object-units (o39 to o43).

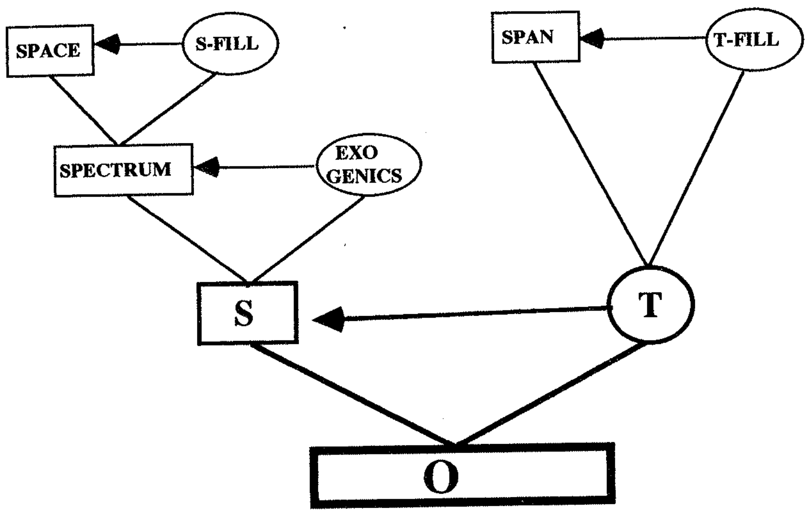


fig.2: the upper-level network of the patches: S = "spatial" (i.e. achronic) evaluation group; T = "time" (i.e. diachronic) evaluation group; "O" is the source/target object (a Midi file).

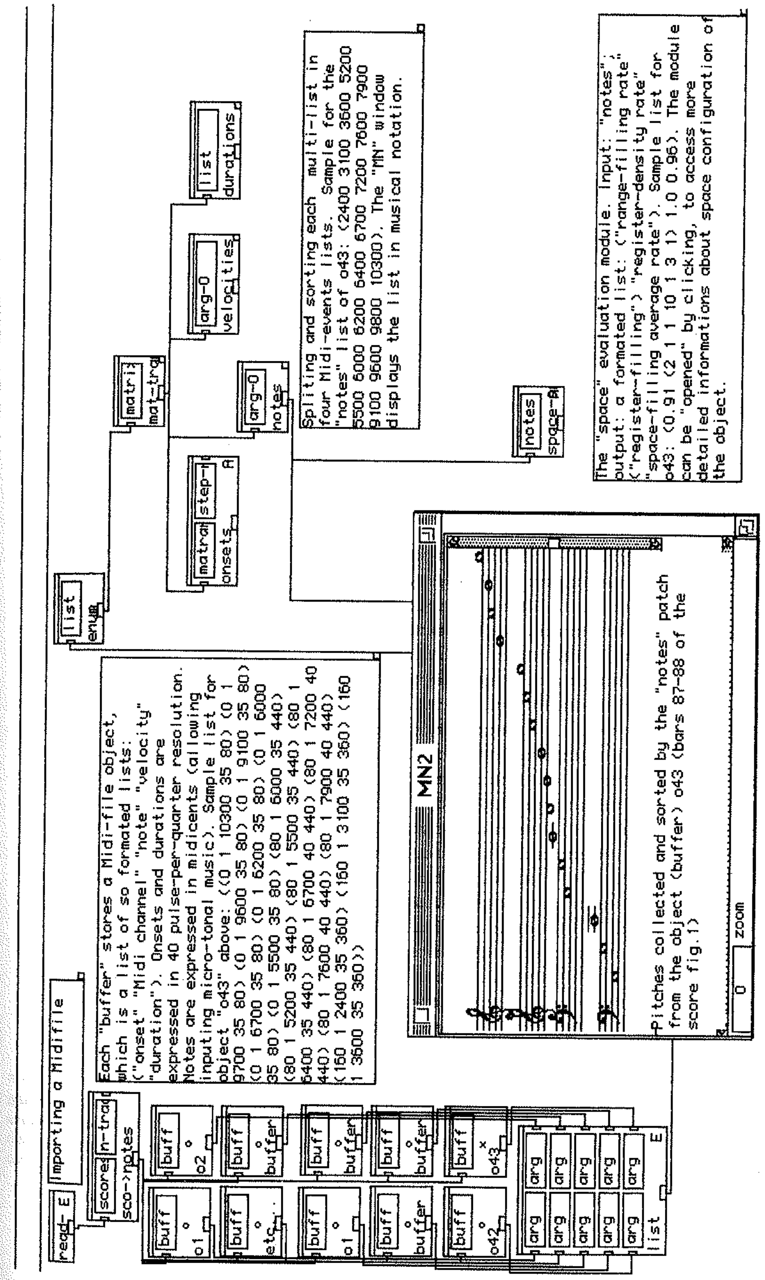


fig.3: The main Patchwork window of the program.

0	int	amb	reg	reg-d	SPACE	S-DENS	S-LIN	HM
1	,30	,77	,75	6	,76	,32	,18	,20
2	,30	,79	,75	6	,77	,31	,20	,33
3	,30	,80	,75	6	,77	,30	,18	,22
4	,30	,41	,25	3	,33	,43	,05	,40
5	,40	,37	,25	3	,31	,55	,09	,50
6	,30	,79	,75	6	,77	,16	,11	,50
7	,30	,82	,75	6	,78	,31	,20	,22
8	,30	,60	,50	4	,55	,26	,36	,17
9	,30	,79	,75	6	,77	,26	,21	,17
10	,30	,79	,75	6	,77	,26	,21	,17
11	,40	,69	,75	6	,72	,30	,23	,10
12	,51	,60	,58	5	,59	,32	,37	,10
13	,59	,74	,58	5	,66	,31	,28	,10
14	,71	,71	,50	4	,61	,32	,31	,10
15	,80	,69	,67	5	,68	,28	,24	,12
16	,80	,49	,25	3	,37	,52	,17	,33
17	,85	,55	,50	4	,52	,47	,26	,50
18	,90	,63	,58	4	,61	,43	,49	,50
19	,90	,61	,50	4	,56	,43	,41	,45
20	,90	,67	,58	5	,62	,44	,48	,47
21	1,00	,61	,50	4	,56	,41	,56	,44
21	,90	,57	,50	4	,53	,22	,46	0
22	,40	,74	,50	5	,62	,20	,20	,50
23	,34	,74	,50	5	,62	,20	,20	,50
24	,30	,76	,50	5	,63	,19	,22	,38
25	,21	,76	,50	5	,63	,19	,22	,38
26	,30	,28	,25	3	,26	,32	,51	,57
27	,30	,36	,25	3	,31	,38	,48	,64
28	,30	,63	,42	4	,52	,27	,52	,36
29	,30	,67	,50	5	,59	,27	,23	,50
30	,51	,69	,50	5	,60	,31	,27	,43
31	,80	,75	,42	4	,58	,26	,31	,42
31	,80	,72	,50	5	,61	,30	,29	,54
32	,90	,75	,42	4	,58	,32	,38	,50
33	,59	,36	,08	2	,22	,78	,27	,47
34	,40	,33	,25	3	,29	,47	,24	,36
35	,30	,47	,25	3	,36	,36	,48	,50
36	,30	,07	,42	2	,24	,43	,50	1,00
37	,30	,07	,42	2	,24	,43	,50	1,00
38	,30	,09	,42	2	,25	,44	,27	1,00
39	,30	,53	,50	4	,51	,45	,49	,29
40	,30	,74	,58	5	,66	,32	,57	,33
41	,30	,91	1,00	7	,96	,34	,27	,12
42	,30	,91	1,00	7	,96	,20	,13	0
43	,30	,91	1,00	7	,96	,20	,13	0

fig.4: a listing ouput. O = label for the source files (i.e. the 43 sequential objects of La Cathédrale engloutie); int = intensity (i.e.`Midi velocity, scaled (0.0-1.0)); amb = range-filling rate; reg = register-density rate; reg-d: this density (an integer corresponding to the number of filled piano registers); space = average value (amb x reg); s-dens = space-density

(cont. next page)

SON	S-DIS	S-FILL	T-DENS	T-LIN	V-ENV	T-DIS	T-FILL
,80	,50	,41	,22	,21	0	,10	,16
,81	,57	,44	,22	,21	0	,10	,16
,78	,50	,40	,22	,44	0	,22	,22
,80	,60	,51	,20	,08	,14	,11	,16
,93	,72	,63	,25	0	,18	,09	,17
,43	,47	,31	,30	,20	,04	,12	,21
,78	,50	,41	,25	0	0	0	,12
,76	,47	,36	,27	,06	0	,03	,15
,70	,44	,35	,78	,06	0	,03	,40
,70	,44	,35	,78	,06	0	,03	,40
,71	,41	,35	,78	,07	,01	,04	,41
,80	,45	,38	,78	,07	,04	,05	,42
,84	,47	,39	,78	,11	0	,05	,42
,84	,47	,39	,91	,11	0	,05	,48
,70	,41	,34	,91	0	0	0	,46
,93	,63	,57	,33	,08	,05	,06	,20
,91	,71	,59	,50	0	,04	,02	,26
,95	,73	,58	,53	,09	,04	,06	,30
,95	,70	,56	,53	,20	0	,10	,31
,94	,70	,57	,30	,10	0	,05	,18
,95	,69	,55	,32	,53	,01	,27	,29
,68	,34	,28	,25	0	,10	,05	,15
,61	,56	,38	,30	0	0	0	,15
,61	,56	,38	,50	0	0	0	,25
,63	,50	,35	,50	0	0	0	,25
,63	,50	,35	,50	0	0	0	,25
,88	,73	,52	,50	,17	,03	,10	,30
,91	,77	,58	,20	,44	0	,22	,21
,81	,59	,43	,19	0	0	0	,09
,72	,61	,44	,30	,20	0	,10	,20
,77	,60	,46	,36	,32	0	,16	,26
,71	,56	,41	,33	0	0	0	,17
,78	,66	,48	,38	0	0	0	,19
,84	,67	,49	,60	0	0	0	,30
,99	,73	,75	,31	,14	,04	,09	,20
,90	,63	,55	,38	0	,01	0	,19
,87	,69	,52	,50	,46	,01	,23	,37
1,00	1,00	,72	,20	0	0	0	,10
1,00	1,00	,72	,50	0	0	0	,10
1,00	1,00	,72	,50	0	0	0	,25
,92	,61	,53	1,00	0	,10	,05	,52
,89	,61	,47	,30	0	0	0	,15
,84	,48	,41	,30	0	0	0	,15
,58	,29	,25	,30	0	0	0	,15
,58	,29	,25	,19	0	0	,39	,29

rate; s-lin = vertical linearity distribution rate; hm = harmonic distribution rate; son = consonance/dissonance rate; s-dis = average value (hm x son); s-fill = average value (s-dens x s-dis); t-dens = time-density rate; t-lin = time-linearity distribution rate; v-env = velocity dispersion rate; t-dis = average value (t-lin x v-env); t-fill = average value (t-dens x t-env). Other available interpreters (as *SPAN* a.o.) not shown.

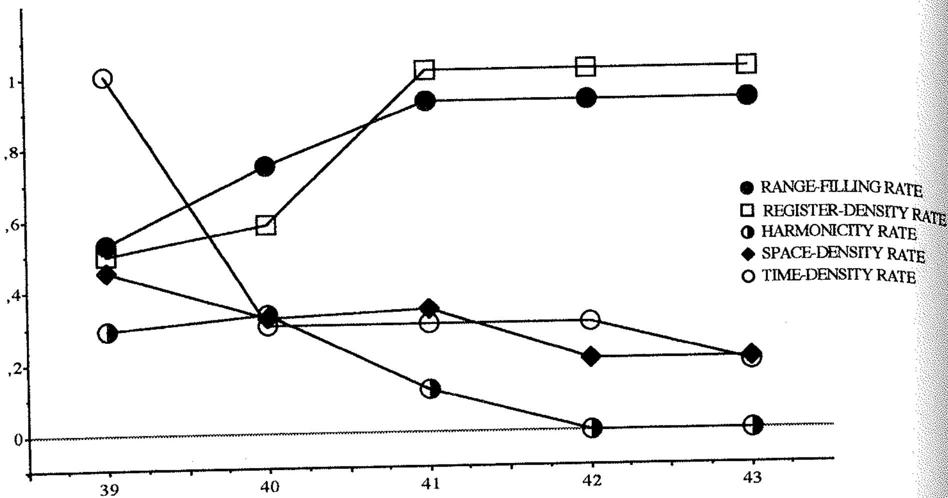


fig.5: The last five sound-objects of "La Cathédrale Engloutie" (objects o39 to o43, x-axis, see score fig.1) as analyzed by five selected interpreters.

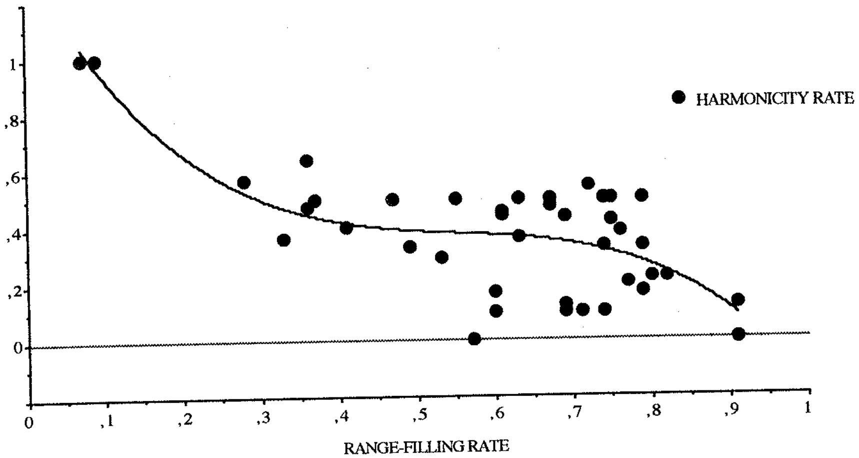


fig.6: This graph shows the impressive negative correlation (factor - 0.73) between the "harmonicity rate" and the "range-filling rate" in "La Cathédrale engloutie" as a whole: the widest the range-filling, the strongest the harmonic distribution rate of the pitch-collections (a high harmonicity rate means an inharmonic structure, and reversely). See fig.4 for the corresponding values ("amb" versus "hm" listings).

Categorial Grammar and Harmonic Analysis

Flávio S. Corrêa da Silva Fábio Kon

Instituto de Matemática e Estatística da Universidade de São Paulo
Cx. Postal 66281 – 05389-970
São Paulo (SP) – Brazil – email: {fcs, kon}@ime.usp.br

Abstract

It is rather commonplace in everyday conversation to refer to the *"Language of Music"*. However, we believe the whole apparatus already built for the analysis of natural language has not been yet as thoroughly used for the analysis of musical phenomena as it could have been. In this article we present some initial ideas towards extending the application of this apparatus for the better understanding of *"Music as Language"*.

In this paper, we apply some techniques from *Categorial Grammar* to represent a simple problem of music theory, which we believe nevertheless to be of widespread interest: functional harmonic analysis. We propose an encoding of the harmonic functions of chords as syntactic categories, and show how the generation of proofs of "harmonic well-formedness" of cadences can be implemented and used as a tool to verify and to display the functional harmonic structuring of cadences.

Keywords: music analysis, harmonic analysis, categorial grammar, syntactic calculus, substructural logics.

1 Introduction

It is commonplace in everyday conversation to refer to the *"Language of Music"*. Indeed, the study of musical phenomena as linguistic objects has been developed by many authors (see e.g. [BCe84, Hol80, LJ83, Sch89]). In this article we present some initial ideas towards extending the application of this apparatus for the better understanding of *"Music as Language"*. More specifically, we employ techniques from *Categorial Grammar* to represent a rather specific and simple problem of music theory, which we believe nevertheless to be of widespread interest: functional harmonic analysis [Bri79].

The aim of Categorial Grammar [Ben87, Ben90, Ben91, Lam58, Lam89] is the analysis of syntactic well-formedness of sentences. The fundamental concept underlying Categorial Grammar is that of *syntactic categories*, which are classes to which words in a sentence must belong. Syntactic categories can be organised as formulae of some substructural logic – e.g. the so-called *Lambek Calculus* [Lam58] – in such way that syntactic well-formedness can be checked via an appropriate *proof theory* related to the logic.

In this paper we propose an encoding of the harmonic functions of chords as syntactic categories and show how the generation of proofs of "harmonic well-foundedness" of cadences can be implemented and used as a tool to verify and to display the functional harmonic structuring of cadences.

In section 2 we briefly review the concepts of Lambek Calculus that we need to use in the rest of the paper. In section 3 we introduce our encoding of harmonic functions of chords as syntactic categories, and show how they can be used to check and to display the functional harmonic structuring of cadences. In section 4 we present a simple PROLOG implementation for checking the harmonic well-foundedness of cadences and displaying functions of chords.

Finally, in section 5 we discuss these results, present some conclusions and propose some future work.

2 The Lambek Calculus

J. Lambek introduced the Syntactic Calculus – most usually called Lambek Calculus nowadays – in [Lam58], as a tool to encode the English grammar, such that well-formedness of sentences could be tested deductively.

Essentially, the Lambek Calculus corresponds to classical propositional logic devoid of any structural rule, in which implication is factorised in two non-commutative connectives. Here, we consider only the implicative fragment of that Calculus, which is sufficient for what we intend to present.

A Gentzen-style presentation of implicative Lambek Calculus can be given by the following rules, in which x, y, z are syntactic categories generated by the members of a set $\mathcal{S}$ of basic syntactic categories and $\Gamma, \overline{\Gamma}, \Delta$ are sequences of syntactic categories. The sequences $\overline{\Gamma}$ are assumed to be non-empty.

axiom: $x \vdash x$.

right-inclusion: left-inclusion:

$$\frac{\overline{\Gamma}, y \vdash x}{\overline{\Gamma} \vdash x/y}. \qquad \frac{\overline{\Gamma} \vdash y; \Gamma, x, \Delta \vdash z}{\Gamma, x/y, \overline{\Gamma}, \Delta \vdash z}.$$

$$\frac{y, \overline{\Gamma} \vdash x}{\overline{\Gamma} \vdash y \setminus x}. \qquad \frac{\overline{\Gamma} \vdash y; \Gamma, x, \Delta \vdash z}{\Gamma, \overline{\Gamma}, y \setminus x, \Delta \vdash z}.$$

For example, Lambek assumes that $\mathcal{S} = \{n, s\}$, in which n stands for "noun" and s stands for "sentence". The words of the English language are attached as labels to formulae, in such way that they can only occur in specific sequences from which the category "s[entence]" can be derived.

Giving a more specific example, if we assume the words `John` and `milk` to be of category "n[oun]", the word `fresh` to be of category "n/n" (a qualifier – must precede the noun it is qualifying to produce a qualified noun) and the word `likes` to be of category $n\backslash s/n$ (a transitive verb – forms a sentence if preceded by a noun – the subject – and followed by another noun – the object of the sentence), we can prove the well-formedness of `John likes fresh milk` with the deduction tree presented in figure 1 (we abbreviate `John`, `likes`, `fresh`, and `milk` by their initials `J`, `l`, `f`, and `m`).

$$\dfrac{\dfrac{\mathtt{m}\colon n \vdash \mathtt{m}\colon n \qquad \dfrac{\mathtt{fm}\colon n \vdash \mathtt{fm}\colon n \quad \mathtt{Jlfm}\colon s \vdash \mathtt{Jlfm}\colon s}{\mathtt{Jl}\colon s/n,\ \mathtt{fm}\colon n \vdash \mathtt{Jlfm}\colon s}}{\mathtt{Jl}\colon s/n,\ \mathtt{f}\colon n/n,\ \mathtt{m}\colon n \vdash \mathtt{Jlfm}\colon s} \qquad \mathtt{J}\colon n \vdash \mathtt{J}\colon n}{\mathtt{J}\colon n,\ \mathtt{l}\colon n\backslash s/n,\ \mathtt{f}\colon n/n,\ \mathtt{m}\colon n \vdash \mathtt{Jlfm}\colon s}$$

Figure 1: Deduction for "John likes fresh milk"

This deduction proves that from the sequence `J`: n, `l`: $n\backslash s/n$, `f`: n/n, `m`: n we can derive the well-formed sentence `Jlfm`: s.

We have employed this Calculus to encode the functional grammar of tonal chords, as detailed in the following sections.

3 Tonal Chords and Syntactic Categories

The set of basic syntactic categories for functional harmony must be large enough to permit the characterisation of all different functions each chord may have in a cadence. We have employed a set of three basic syntactic categories $\mathcal{S} = \{a, b, c\}$, a and c loosely corresponding to the concepts of *tonic* and *cadence*, related to Lambek's *noun* and *sentence* functions, and b corresponding to an intermediate concept leading to the idea of *subdominant*. Intuitively, we have a as tonic, $a \setminus b$ as subdominant (fulfilled when preceded by something of category a) and $b \setminus c/a$ as a full cadence (fulfilled when preceded by some chain of chords of category b and followed by something of category a). In order to present our proposed encoding of chords as representatives of syntactic categories, we must introduce some notation.

We have adopted the (first twelve) MIDI codes for pitch values, and hence the notes **C**, **C♯**, **D** ... are denoted respectively as $0, 1, 2, \ldots$. The syntactic categories of the functions of chords can then be encoded in a dictionary like the one presented in table 1. In this dictionary, $i = 0, 1, \ldots, 11$, and these numbers are operated modulo 12, i.e. $6 + 5 = 11, 6 + 6 = 0, 6 + 7 = 1$ etc. (and, of course, in table 1 we have only a small fragment of one such dictionary).

Major Mode

entry	chord	tonality						
		i	i+1	i+3	i+5	i+7	i+8	i+10
i^1	i,i+4,i+7	a			$b\backslash c/a$	$a\backslash b$		
i^2	i,i+3,i+7, i+10							$a\backslash b$
i^d3	i,i+4,i+7, i+10				$b\backslash c/a$			
i^4	i,i+4,i+7, i+11	a				$a\backslash b$		
i^5	i,i+4,i+7, i+11,i+2	a						
i^7	i,i+3,i+8		$b\backslash c/a$	$a\backslash b$			a	

Minor Mode

entry	chord	tonality					
		i	i+1	i+4	i+5	i+7	i+9
i^1	i,i+4,i+7				$b\backslash c/a$		
i^3	i,i+4,i+7, i+10				$b\backslash c/a$		
i^6	i,i+3,i+7	a				$a\backslash b$	
i^7	i,i+3,i+8		$b\backslash c/a$				
i^8	i,i+4,i+9			$a\backslash b$			a

Table 1: Dictionary of Syntactic Categories of Chords ($i = 1, \ldots, 11$ is the root of the chord)

It should be observed that syntactic categories refer to specific tonalities and modes. We avoid referring explicitly to tonalities and to modes in our deduction trees to preserve our notation as simple as possible. Now, using the notation of table 1, if we attach the perfect major triads $0^1, 5^1, 7^1$ as labels to the categories $a, a\backslash b$ and $b\backslash c/a$, we can derive the well-formedness of the perfect cadence $\{0^1, 5^1, 7^1, 0^1\}$ (figure 2).

$$\dfrac{\dfrac{0^1 : a \vdash 0^1 : a \qquad \dfrac{0^15^1 : b \vdash 0^15^1 : b \quad 0^15^17^10^1 : c \vdash 0^15^17^10^1 : c}{0^15^1 : b, 7^10^1 : b\backslash c \vdash 0^15^17^10^1 : c}}{0^1 : a, 5^1 : a\backslash b, 7^10^1 : b\backslash c \vdash 0^15^17^10^1 : c} \qquad 0^1 : a \vdash 0^1 : a}{0^1 : a, 5^1 : a\backslash b, 7^1 : b\backslash c/a, 0^1 : a \vdash 0^15^17^10^1 : c}$$

Figure 2: Deduction for the perfect cadence

```
| ?- harmon([[c,e,g], [d,f,a], [g,b,d], [c,e,g]], X).

X = [[c,maj]] ?

yes

| ?-
```

Figure 3: Using the Theorem Prover

A theorem prover for this Calculus can be implemented in PROLOG, and in the following section we present a very simple implementation for it.

4 Functional Harmony in PROLOG

The PROLOG code for a simple implementation of a theorem prover for the Calculus presented above is introduced in the appendix following this paper. This program works as follows: given a sequence of chords $[C_1, ..., C_n]$, the procedure **genseq** converts this sequence into a sequence of sets of harmonic functions $\mathcal{F}_i$ that each chord C_i can have. From these, the procedure **cadence** selects the functions $f_i \in \mathcal{F}_i$ such that from $f_i : i = 1, ..., n$ we can derive the function c of any tone and mode. These functions are then presented as solutions, with the corresponding tone and mode of the derived cadence.

For example, if we want to check the well-formedness of the sequence of chords in figure 2, we obtain the following (figure 3). This output indicates that, for the fragment of tonal functional harmony encoded above, the only syntactic category of type "c" that can be derived from the given sequence of chords is that of **C** major.

5 Conclusions and Future Work

In this paper we presented an encoding of the harmonic functions of chords as syntactic categories, and showed how the generation of proofs of "harmonic well-foundedness" of cadences could be used as a tool to verify and to display the harmonic functional structuring of cadences. We have also presented an implementation of a theorem prover for automating this verification.

Clearly, there is still much to be done on turning Categorial Grammar applied to functional harmonic analysis a more friendly tool for musicians. Nonetheless, our initial experiments suggest that this can be a useful tool, not only for analysis but also for generation of cadences upon certain constraints, e.g. when building accompaniments for given melodies.

Immediate future work shall include the study of applicability of this tool in practical situations of interest for musicians and for students of music, and the extension of our "dictionary" to encompass richer harmonic cadences. It shall also be interesting to further analyse the mathematical properties of tonal harmony under the viewpoint of Lambek Calculus, and to study what the (musical) consequences could be of altering some of these mathematical properties (e.g. by adding some structural rules or different connectives to the Calculus).

The program presented here is also available by ftp at
`ftp.ime.usp.br:/pub/music/lambek`, or directly from the authors.

Acknowledgements: this work has been partially supported by FAPESP grant 93/0603-01, and CNPq grant 300041/93-4.

References

[BCe84] M. Baroni and L. Callegari (eds.). *Musical Grammars and Computer Analysis.* Univ. degli Studi di Bologna e Modena, 1984.

[Ben87] J. Benthem. A Linguistic Turn: New Directions in Logic. In *Logic, Methodology and Philosophy of Science VII*, pages 205–240. Elsevier, 1987.

[Ben90] J. Benthem. Categorial Grammar and Type Theory. *Journal of Philosophical Logic*, 19:115–168, 1990.

[Ben91] J. Benthem. Language in Action. *Journal of Philosophical Logic*, 20:225–263, 1991.

[Bri79] C. M. Brisolla. *Princípios de Harmonia Funcional.* Novas Metas, 1979.

[Hol80] S. R. Holtzman. A Generative Grammar Definition Language for Music. *Interface*, 9:1–48, 1980.

[Lam58] J. Lambek. The Mathematics of Sentence Structure. *American Mathematics Monthly*, 65:154–169, 1958.

[Lam89] J. Lambek. Multicategories Revisited. *Contemporary Mathematics*, 92:217–239, 1989.

[LJ83] F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music.* MIT Press, 1983.

[Sch89] E. F. Schurmann. *A Música como Linguagem.* Brasiliense / CNPq, 1989.

Appendix: A Theorem Prover for Functions of Chords

```
/*******************************************************************/
/* harmon(L, A) - the collection of harmonic justifications for    */
/*              sequence of chords L is A                          */
/*******************************************************************/

harmon(L, A) :- miditransl(L, Ln), genseq(Ln, F),
    setof([X,Y], cadence(F, X, Y), An), midiback(An, A).

miditransl([H|T], [Hn|Tn]) :- transchord(H, Hn), miditransl(T, Tn).
miditransl([], []).

transchord([H|T], [Hn|Tn]) :- transnote(H, Hn), transchord(T, Tn).
transchord([], []).

transnote(c, 0). transnote(c_sharp, 1). transnote(d_flat, 1).
transnote(d, 2). transnote(d_sharp, 3). transnote(e_flat, 3).
transnote(e, 4).
transnote(f, 5). transnote(f_sharp, 6). transnote(g_flat, 6).
transnote(g, 7). transnote(g_sharp, 8). transnote(a_flat, 8).
transnote(a, 9). transnote(a_sharp, 10).transnote(b_flat, 10).
transnote(b, 11).

midiback([H|T], [Hn|Tn]) :- noteback(H, Hn), midiback(T, Tn).
midiback([], []).

noteback([0, M], [c, M]). noteback([1, M], [c_sharp_d_flat, M]).
noteback([2, M], [d, M]). noteback([3, M], [d_sharp_e_flat, M]).
noteback([4, M], [e, M]).
noteback([5, M], [f, M]). noteback([6, M], [f_sharp_g_flat, M]).
noteback([7, M], [g, M]). noteback([8, M], [g_sharp_a_flat, M]).
noteback([9, M], [a, M]). noteback([10, M], [a_sharp_b_flat, M]).
noteback([11, M], [b, M]).

/*******************************************************************/
/* genseq(S, L) - the collection of candidate sequences of         */
/*              harmonic functions for S is L                      */
/*******************************************************************/

genseq(S, L) :- genfunct(S, F), remap(F, L).

genfunct([H|T], L) :- genfunct(T, T2), setof(F, function(H, F), S),
    append([S], T2, L).
genfunct([], []).

function([H|T], [Y, Fun, Mod]) :- funct(Lf, [Z, Fun, Mod]),
    match(H, [H|T], Lf), Y is ((Z + H) mod 12).
```



```
/*******************************************************************/
/* funct(H0, F0) - dictionary of harmonic functions                */
/*******************************************************************/

funct([0,4,7], [0, [a], maj]). funct([0,4,7], [5, [b,e,c,d,a], maj]).
funct([0,4,7], [7, [a,e,b], maj]).
funct([0,3,7,10], [10, [a,e,b], maj]).
funct([0,4,7,10], [5, [b,e,c,d,a], maj]).
funct([0,4,7,11], [0, [a], maj]). funct([0,4,7,11], [7, [a,e,b], maj]).
funct([0,4,7,11,2], [0, [a], maj]).
funct([0,3,8], [1, [b,e,c,d,a], maj]).
funct([0,3,8], [3, [a,e,b], maj]). funct([0,3,8], [8, [a], maj]).
funct([0,4,7], [5, [b,e,c,d,a], min]).
funct([0,4,7,10], [5, [b,e,c,d,a], min]).
funct([0,3,7], [0, [a], min]). funct([0,3,7], [7, [a,e,b], min]).
funct([0,3,8], [1, [b,e,c,d,a], min]).
funct([0,4,9], [4, [a,e,b], min]). funct([0,4,9], [9, [a], min]).

/*******************************************************************/

match(X, [H1|T1], [H2|T2]) :- H1 is ((X + H2) mod 12), match(X, T1, T2).
match(_, [], []).

remap([H|T], L) :- remap(T, T2), combine(H, T2, L).
remap([T], L) :- combine(T, [[]], L).

combine([H|T], L1, L2) :- combine(T, L1, T2), comb(H, L1, H2),
    append(H2, T2, L2).
combine([], _, []).

comb(A, [H1|T1], [H2|T2]) :- comb(A, T1, T2), append([A], H1, H2).
comb(_, [], []).

append([H|T], L1, [H|T2]) :- append(T, L1, T2).
append([], L, L).
```

```
/*********************************************************************/
/* cadence(L, Ton, Mod) - L forms a cadence of Ton - Mod            */
/*********************************************************************/

cadence([H|_], X, Y) :- theor(H, [X, [c], Y]).
cadence([_|T], X, Y) :- cadence(T, X, Y).

theor([H|T], [X, F, Y]) :- theor(T, [X, L, Y]),
    prove(H, [X, L, Y], [X, F, Y]).
theor([[X, F, Y]], [X, F, Y]).

prove([X, L1, Y], [X, [F|T2], Y], [X, L2, Y]) :-
    invert(L1, [F, d|T1]), invert(T1, T1i), append(T1i, T2, L2).
prove([X, L1, Y], [X, [F, e|T2], Y], [X, L2, Y]) :-
    invert(L1, [F|T1]), invert(T1, T1i), append(T1i, T2, L2).

invert([H|T], L) :- invert(T, Ti), append(Ti, [H], L).
invert([], []).

/*********************************************************************/
```

Figure 4: A Theorem Prover for Functions of Chords

An Optimized Method for Storage, Transmission and Display of Digital Audio Data

CARLOS AUGUSTO PAIVA DA SILVA MARTINS
RUGGERO ANDREA RUSCHIONI
SÉRGIO TAKEO KOFUJI
[capsm, roger, kofuji]@lsi.usp.br
Escola Politécnica da Universidade de São Paulo
LSI - Laboratório de Sistemas Integráveis
Divisão de Sistemas Digitais
Av. Professor Luciano Gualberto, 158, Travessa 3
05508-900 - São Paulo - SP - Brasil

Abstract

This paper analyses the application of a new reconstruction method for digital audio signals. The method is called Normalized Sampled Finite Sync Reconstructor (NSFSR) and its behavior is closer to the ideal reconstructor used in the original sampling theorem. When compared to the reconstruction method currently used, the Zero Order Reconstructor (ZOR), it substantially reduces storage size, transmission bandwidth and synthesis time. Both qualitative and quantitative analysis of the methods are presented.

Introduction

This paper analyses the application of a new method of digital audio signal reconstruction. The goal is to reduce the storage size and consequently the transmission bandwidth the synthesis time. The reconstruction method was designed with the intent of surpassing the quality of the popular Zero Order Reconstructor (ZOR), which produces jag effects in one dimensional signals [Martins, 1994].

Digital audio signals are currently used in many computer applications, including entertainment and telecommunications. In addition our research is important for future applications such as computer music, virtual reality, distributed multimedia, scientific visualization and digital television, among others.

Most existing signals of importance to the humans, such as audio or static and dynamic images, may be considered analog at the macroscopic level. Nonetheless, for these signals to be manipulated and processed by digital computers, they need to be transformed into digital signals through sampling, quantizing and codification. Since our final objective is the display of the analog signal, after all the digital signal processing the data must be converted back to the analog domain.

According to the sampling theorem [Shannon, 1949], the minimum sampling frequency must be twice the frequency of the highest component in the analog signal. However it is common practice to use sampling frequencies well above this minimum. Reconstructors often fail to meet theoretical performance levels, and higher sample rates simplify the design of the analog lowpass filter used after digital to analog conversion.

Currently almost all systems that work with digital signals use digital to analog converters (reconstructors) connected to the output devices, that can be modeled as a zero order reconstuctor (ZOR). Since this is not the ideal reconstructor used in the sampling theorem, errors occur in the reconstruction.

Our main objective is to show it is possible to use sampling frequencies closer to the minimum and obtain better results than by employing the ZOR, using the NSFSR.

We must remember that audio signals are a subclass of the one dimensional signals which present unique features related to human perception. What is an essential characteristic of audio signals is not essential in other one dimensional signals. In this paper we will focus more on the general characteristics of one dimensional signals which are also valid for audio signals, such as storage size requirements, bandwidth for transmission and the quality of the generated signal.

We compare the use of the ZOR and the NSFSR on audio signals through the analysis of the reconstructed signals in the time and the frequency domains.

Results are analyzed qualitatively, comparing the reconstructed signals' wave form and sound, and quantitatively comparing size and the complexities of the reconstruction methods.

Reconstruction Method

The reconstruction method presented here is named "Normalized Sampled Finite Sync Reconstructor" (NSFSR). It is a new reconstructor which matches more precisely the ideal reconstructor proposed in the sampling theorem (Shannon, 1949) than the Zero Order Reconstructor (ZOR) used in practice.

It is very important to remember that the sampling process corresponds to a time domain multiplication between the original signal and the sampling impulse train. In an equivalent form it corresponds to a convolution between the Fourier transforms of the original signal and the sampling impulse train. In this way the sampled signal in the frequency domain is formed by copies of the original analog signal in the frequency domain. This copies are centered at points which are multiples of the sampling frequency employed.

The reconstruction process from the sampled signal corresponds to a time domain convolution between the sampled signal and the reconstruction filter. Equivalently it corresponds to a multiplication between the Fourier transforms of the sampled signal and of the reconstruction filter. According to the sampling theorem, using a sampling frequency above Nyquist limits and using the ideal reconstructor the reconstructed sampled signal should be identical to the original. The demonstration of this affirmation is the proof of the sampling theorem. Detailed mathematical analyses may be found in [Brigham, 1988], [Oppenheim, 1983], [Oppenheim, 1989] and [Shannon, 1949].

The ideal reconstructor, named time domain sync, has infinite support, and cannot be implemented in practice. The NSFSR is generated by sampling the ideal sync reconstructor in the time domain with a finite number of points; it is later on normalized by dividing each sample by the sum of the values of all samples. Once the number of samples is defined in the reconstructor, on each side of the interpolated point will fall half the points which form the ideal sync reconstructor, sampled and normalized. A detailed analysis of this reconstructor is presented in [Martins, 1994], chapter 7.

Results

We applied the ZOR and NSFSR in one dimensional signals and analyzed them. For the reconstructors implementation, the software package Matlab was used; for the analyses of the results of the reconstructors in the signals Matlab and the Khoros environments were used. Here we present the analyses of a low frequency sine wave. Nevertheless it is important to mention that the conclusions are also valid for other complex signals as presented in more extensive tests involving different signals in [Martins, 1994], chapter 6. In the analyses we considered both subjective quality factors and quantity factors in time and frequency domain.

Qualitative analyses

Proceeding with the quality analyses, disregarding sound perception factors, we may observe some important features in the reconstructed signals employing the ZOR and the NSFSR. Figures 1 to 4 show us the zoom of the original analog signal and the time domain reconstruction. We see that the signals generated by the reconstructors NSFSR 2 points and NSFSR 6 points do not show any perceivable jag effect as in the one generated by ZOR. Analyzing visually the images 3 and 4 we see that there is no jag effect for the NSFSR 2 points in figure 3 and for the NSFSR 6 points in figure 4, seeming that NSFSR 2 points shows better results.

Analyzing figures 5 to 8, which show the original signal and the reconstructed signal in the frequency domain, we notice that in the signal generated by the ZOR the frequency components in the repetition points generated by sampling, have magnitude approximately 100 times greater than the ones generated by the NSFSR 2 points and approximately 20 times greater than the ones generated by the NSFSR 6 points. These high frequency components, of quite significant magnitude in the replication points generated by sampling, are what characterizes the existence of jag effects in the time domain.

The reconstruction error introduces high frequency components not present in the original signal, changing the spectrum of the original signal and eventually its timbre. The additional frequency components are a consequence of a non ideal lowpass filtering, because we used a zero order reconstructor, while sampling errors (aliasing) introduces low frequencies.[Brigham, 1988; Oppenheim, 1975].

Differences between the original signal and the reconstructed with the family of NSFSR in the frequency domain is smaller than between the original and the ZOR signal. The NSFSR 2 points present the smaller magnitude in the replication points.

Observing figures 9 to 11, which show the absolute difference signal between the original and reconstructed signals, we notice that the best results are presented by the NSFSR 2 points which yields a maximum value of about 1.2×10^{-3} We should warn that by itself this measurement is not sufficient to select a better reconstructor.

We concluded by analyzing visually the wave forms that best results are obtained through the NSFSR 2 points, and the analyses by the difference signal also point to this as the best reconstructor.

Also should be stressed that either NSFSRs are far better than the ZOR and apparently do not present oscillation problems (Gibbs effect).

A qualitative analysis based on the sound of the reconstructed signals presented by the ZOR and the two NSFSRs shows that the introduction of high frequency partials on the spectrum of the original signal by the ZOR changes the timbre considerably, while the NSFSR reconstruction preserves the original sound characteristics. Both the original sound and the reconstructed through the NSFSR 2 points are audibly indistinguishable from each other, while the signal reconstructed with the ZOR presents complex spectra due to the artifacts (high partials) introduced by the reconstruction error of the ZOR. So we conclude by perceptive analyses that the NSFSR 2 points also yield the best results.

Quantitative analysis

We carried on some tests to analyze the transmission of audio signals quantitatively on an actual network. The aim of such tests is to verify, in broad terms, the times expended in the transmission of signals with different samples. We simulated the transmission of an audio signal with a number of samples far greater than the minimum established by the sampling theorem and the signal critically sampled. The smallest signal is later interpolated [Oppenheim, 1989] using the ZOR and the NSFSR.

The files used in the process had sizes of 262.144 bytes and 16.384 bytes, the latter is 16 times smaller than the former.

Tests were realized using a local area network configuration (same building) and long distance (two cities 240 Km apart). We transmitted both files under two net traffic conditions, heavy load and light load. We should warn that due to the illustrative purposes of the measurements we didn't monitor the net load during the transmissions, the classification was based on the transmission's time.

Based on the obtained results, we calculated the sample average and using the mean value estimate method calculated the confidence interval for the distributed average, supposed normal. For the calculation of the confidence interval it was used an uncertainty coefficient of $\alpha = 0.05$.

In spite of the limited amount of transmissions, the sampled average obtained is very similar to the probable distribution average. The values for the standard deviation for local transmission were greater than the ones obtained for long distance. This happens because the transmission time in local networks are considerably faster and so more influenced by load variations. For the same reason, on a net under the same load, the transmission times for smaller files present greater deviation than average when compared to files of greater size. The average of the transmission times had a very small variation under conditions of heavy and light load, this suggests that the net did not present definite periods of heavy and light load or that these periods do not follow a known pattern.

The gain values obtained during transmission time, considering the average, were: 5.01 for local net with light load, 5.03 for local net with heavy load, 18.8 for long distance net with light load and 19.5 for long distance net with heavy load.

We noticed the gains should equal 16 if the behaviors of the system were deterministic, which is not true for the net configurations used in this tests. Nevertheless the storage gains are deterministic and always equal 16.

Comparing quantitatively the complexity of the reconstruction algorithms tested, we notice that the NSFSR presents greater complexity in the order of approximately 8 operations per sample against 1 for the ZOR. In software implementations this is a considerable drawback.

At present we are developing an implementation of the NSFSR in hardware, in this case the complexity drawback is unimportant because we use high speed dedicated hardware and optimized parallel structures to operate in real time even for very high sample rates.

Conclusion

The reconstruction error introduces high frequencies in the reconstructed signal, while sampling error (aliasing) introduces low frequencies (Oppenheim, 1975; Oppenheim, 1983; Proakis, 1992).

The quality analyses of the reconstructed signals show that applying the new method of NSFSR in digital audio signals yields superior results when compared to the currently used ZOR. At least for this particular signal the timbre differences were noticeable, we still don't know how a ZOR with a properly designed lowpass filter might affect the reconstruction of complex signals such as those found in recordings of natural instruments. This study provides an indication, though, that there might be timbre distortion introduced by the reconstructor. We plan to do further tests on recordings of natural instruments and compare the quality of the two reconstructors.

The transmission of digital audio data is a crucial point in the above mentioned applications, the gain obtained in the transmission of files with lesser samples are considerable. For most current multimedia applications and net transmissions which use sampling rates as low as 8KHz to 16KHz, the gain in sound quality is expressive. We conclude that the transmission of signals with the minimum number of samples drawn up by the sampling theorem - which is the method used - reduces the importance of this crucial parameter, allowing the practicability of applications which demand greater flux of digital audio signals.

Considering the advantage of transmitting a digital audio signal with a smaller number of samples, which does not violate the minimum sampling value, we still need to choose the method to be used for reconstruction. Even though the NSFSR has a greater algorithmic complexity in relation to the ZOR, the qualitative and quantitative analyses show that the NSFSR is still the best solution. For transmission purposes also we should note that the approach here presented is independent of any compression scheme. Further compression techniques could well be employed over the signal minimizing even further the file size.

So we reached our main objective, showing that by using the NSFSR it is possible to use a lower sampling frequency and obtain better results than the ones we currently obtain using the ZOR.

Furthermore we should also point that using the NSFSR we might not need to use the lowpass filters, which have a complex design. In the ideal case we would not need to use such a lowpass filter after the digital to analog conversion.

Based on all the results, which are expressive and promising, we believe that using the NSFSR on audio signals is desirable and presents various advantages over the current approach. However many tests and development are necessary, this is the beginning of a new research of medium-long term.

Future Works

As said above, audio signals present unique aspects related to perception. In this paper we focused on general aspects valid for most one dimensional signals. It is our intent to carry on deeper research on the psychoacoustic effects of the reconstructor, particularly the effects on timbre. We believe to have enough evidence that the current approach using the ZOR might be introducing unwanted high frequency components in some audio applications. We will proceed on the comparative study of both reconstructors and their effect on recordings of complex sounds such as natural instruments.

The algorithm employed for the NSFSR has greater complexity than the ZOR. In the tests presented in this paper we used a software implementation. We have already started working on a hardware implementation of the NSFSR, in this case the computing drawback is irrelevant since we are using high speed dedicated hardware and optimized parallel structures for real time applications even for high sampling rates.

Lastly, regarding the reconstruction method, we are continuing our research developing other reconstructors better optimized than the NSFSR used in this work.

References

Brigham, E. O. (1988). *The fast Fourier transform and its applications*. Englewood Cliffs, NJ: Prentice-Hall Inc.

Oppenheim, A.V. & Schafer, R. W. (1975). *Digital signal processing*. Englewood Cliffs, NJ: Prentice-Hall Inc.

Oppenheim, A.V. & Willsky, A. S. (1983). *Signals and Systems*. Englewood Cliffs, NJ: Prentice-Hall Inc.

Oppenheim, A.V. (1989). *Discrete-time signal processing*. Englewood Cliffs, NJ: Prentice-Hall Inc.

Martins, C. A. P. S. (1994). *O efeito de serrilhamento em dispositivos matriciais de exibição de imagens como erro de reconstrução*. Dissertação (Mestrado), PPGEE-UFMG, Belo Horizonte.

Proakis, J. G. & Manolakis, D. G. (1992). *Digital signal processing: principles, algorithms, and applications*. 2nd Edition, New York, Macmillan Publishing Company.

Shannon, C. E. (1949). Communication in the Presence of Noise. *Proceedings of the I.R.E.*, jan 1949, 10-21.

Acknowledgments

The reconstruction method used in this paper was developed as part of the research necessary to obtain the title of Master of Science of one of the authors at PPGEE-UFMG. This work was sponsored by CAPES. The authors would like to thank the researchers Luis Gustavo G. Kiatake and Marcelo H. Cintra for their collaboration on the digital audio transmission tests. This research is being developed at LSI-USP, the authors wish to express their gratitude for the support of this laboratory and its staff.

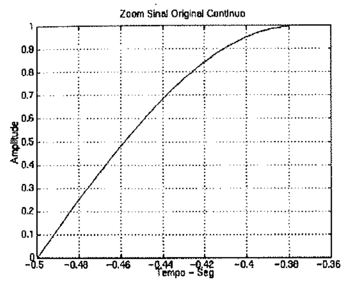


Fig. 1 Original signal zoom

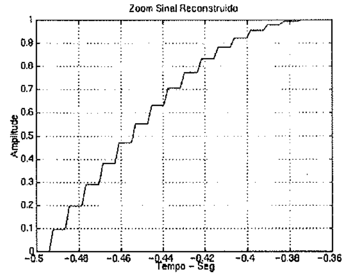


Fig. 2 ZOR signal zoom

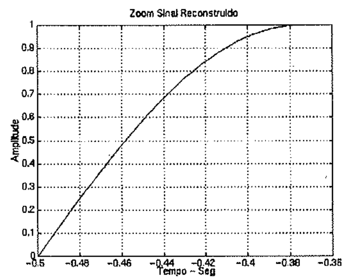


Fig. 3 NSFSR 2p signal zoom

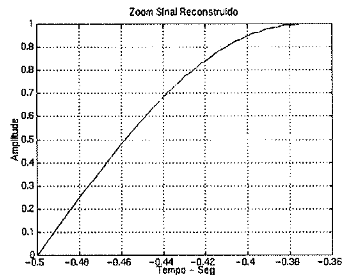


Fig. 4 NSFSR 6p signal zoom

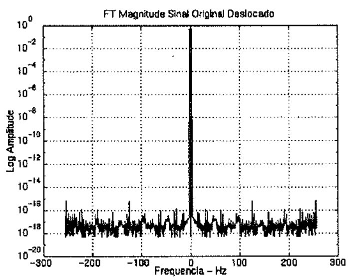


Fig. 5 Original signal FT mag log

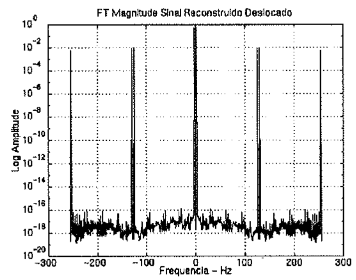


Fig. 6 ZOR signal FT mag log

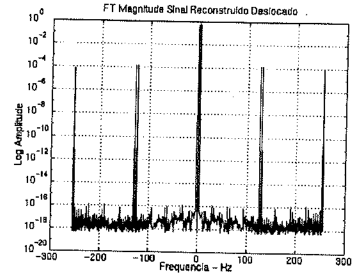


Fig. 7 NSFSR 2p signal FT mag log

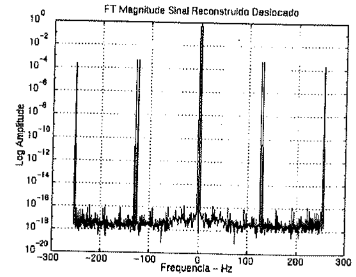


Fig. 8 NSFSR 6p signal FT mag log

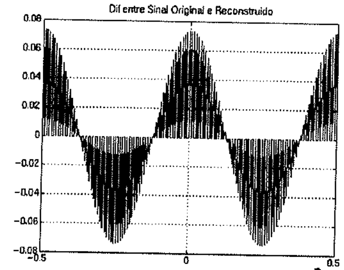


Fig. 9 Original - ZOR dif signal

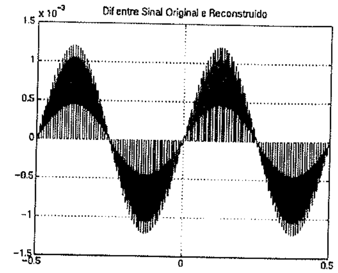


Fig. 10 Original - NSFSR 2p dif signal

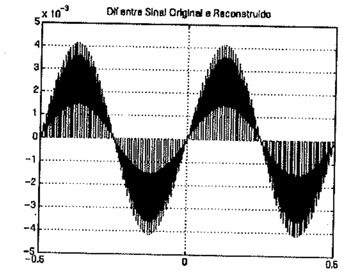


Fig. 11 Original - NSFSR 6p dif signal

Transformações sonoras através de operações timbrais

JOSÉ EDUARDO FORNARI [1] , FURIO DAMIANI [1,2]

1. Depart. Semicondutores Instrumentos e Fotônica - DSIF/FEE/UNICAMP
2. Núcleo Interdisciplinar de Comunicações Sonoras - NICS/UNICAMP
CEP. 13.081-970 Campinas-São Paulo - Brasil
e-mail:: Fornari@dsif.fee.unicamp

Resumo:

Descreve-se o projeto de um sistema de processamento digital para audio (ADSP), em tempo-real. O sinal de áudio de entrada é digitalizado, tem seu espectro de frequência determinado, é submetido a operadores que modificam seu espectro e é transformado em sinal analógico de saída. O espectro de frequência é obtido pela transformada rápida de Fourier (FFT), sendo sua inversa (IFFT) utilizada para gerar o sinal de saída. Os operadores timbrais manipulam as magnitudes dos harmônicos que compõem o espectro, e podem ser parametrizados por controladores externos, bem como pelo "passado" do sinal sonoro de entrada..

1. Do som ao timbre:

O som é constituído por vibrações mecânicas, expansões e compressões longitudinais que se propagam em um meio material com uma velocidade característica e pode ser caracterizado por três grandezas físicas (frequência, intensidade e duração) (Parker). A variação temporal da intensidade pode ser mostrada *macroscopicamente*, pela envoltória da intensidade no tempo (ataque, sustentação e decaimento) e *microscopicamente* pelas variações temporais da amplitude dentro da envoltória, com freqüência aproximadamente periódica, percebendo-se assim o efeito de seus componentes harmônicos [Alles].

O sentido da audição humana possui limites. <u>Limites de intensidade</u>: ouvem-se sons com intensidades entre: 10^{-12} W/m^2 e 10^{-1} W/m^2, correspondentes aos limiares de percepção e de dor. A percepção da variação da intensidade é de cerca de 10% da intensidade sonora total. <u>Limites de frequencia</u>: aproximadamente entre 20 e 20.000 Hz (um piano produz sons com frequências fundamentais entre 27 Hz e 3480 Hz). A maior sensibilidade à variação de frequência fica na faixa de 500 a 4.000 Hz. para a qual é de cerca de 0,3% da freqüênciasendo ouvida, ou 1/20 de semitom [Culver]. <u>Limite de duração</u>: Conhecido como persistência auditiva, é de aproximadamente 50ms.

O timbre é uma percepção subjetiva, e sua conceituação procura definir o que é percebido pela audição que permite reconhecer e distinguir um som. O timbre depende de vários fatores, como a envoltória do espectro de frequência e de suas variações temporais, as variações periódicas da amplitude, a frequência fundamental, se o som for melódico ou percussivo" [Plomp]. A *American Standards Association* o define: "considerando estritamente o estado de regime sonoro, timbre é: o atributo da sensação auditiva pelo qual um indivíduo pode julgar se dois sons complexos e sustentados, com a mesma *altura* e *loudness* são diferentes entre si".

Uma maneira de representar o som de modo a tornar as informações que constituem o timbre observáveis, é através de uma superfície, chamada aqui de superfície sonora ou SS, que representa a variação temporal das magnitudes dos componentes do espectro do som. O conjunto das componentes de frequência de um som num dado instante é o seu *espectro de frequência*. De maneira geral, esse espectro é limitado e contínuo (formado por infinitos componentes dentro dos limites de freqüência da audição), sendo a SS uma superfície contínua com tres dimensões: freqüência, amplitude e tempo. Uma seção da SS ao longo de uma mesma freqüência gera um plano onde a magnitude dessa freqüência é representada em função do tempo, ou *plano morfológico*; uma seção para um dado instante de tempo representa o espectro de freqüência deste instante, ou *plano harmônico*.

A SS pode ser discretizada, dando origem a uma matriz que a pode representar adequadamente desde que os erros de discretização estejam dentro das limitações da percepção auditiva. Isto torna possível o processamento digital de manipilação do timbre.

2. As transformações sonoras

As transformações são operações matemáticas para o mapeamento, ou representação, de um conjunto de pontos de um espaço vetorial R^n em outro espaço vetorial R^m. Podem ser lineares e não-lineares. Uma transformação linear deve satisfazer a: F(u+v) = F(u) + F(v), e F(k.v) = F(v), caso contrário, ela é dita não-linear [Boldrini]. Os equipamentos para transformações sonoras podem ser agrupados em classes, pela grandeza física que transformam. <u>Transformações da duração:</u> É a classe dos equipamentos que atuam na envoltória temporal do som, como reverberadores, retardadores, câmaras de eco. <u>Transformações do espectro</u>: É a classe do que atuam na envoltória de freqüência do som, como os filtros, deslocadores espectrais, *Chorus*. <u>Transformações da intensidade:</u> É a classe dos equipamentos que atuam na amplitude do som, como os prolongadores, equalizadores, compressores de nível. As transformações sonoras nunca transformam exclusivamente uma única grandeza sonora, elas atuam predominantemente nela. As transformações que agem, simultânea e correlacionadamente, em mais de uma grandeza, como as transformações não-lineares, são classificadas como *transformações mistas*.

O som tanto pode ser transformado tanto analógica quanto digitalmente. Respeitando-se os limites de percepção da audição humana, a representação digital do som o pode representar adequadamente (sem perdas perceptíveis) [Chamberlin]. Os sistemas de transformação sonora digital estão sendo cada vez mais utilizados devido a sua grande flexibilidade na manipulação de dados, ao crescente e significativo aumento de suas velocidades de processamento e à diminuição de seu custo. O processador digital de sinais, DSP (Digital Signal Processor), quando especializado para processamento sonoro, é chamado de processador digital de audio, ADSP (Audio DSP).

Um som contínuo possui um espectro de frequência com um número infinito de componentes da análise de Fourier, obteníveis com o uso da transformada de Fourier. No caso de sua representação digital, para uma amostra de N pontos, obtem-se um espectro discreto com N/2 componentes, calculados pela transformada discreta de Fourier. O algoritmo otimizado que os calcula é a transformada rápida de Fourier, FFT. A representação digital da SS será uma matriz cujas linhas correspondem a um dado instante discreto no tempo e cujas colunas correspondem às frequências discretas componentes do espectro, sendo os valores numéricos dos elementos da matriz iguais às magnitudes das componentes desse espectro. Esta matriz será chamada de matriz harmônica, ou MH.

3. A operação timbral

Definimos como transformação do timbre a transformação de cada instante discretizado de um som, re-parametrizada a cada novo instante de tempo. Uma operação timbral realiza uma transformação do espectro, através do controle e da parametrização desta operação em tempo-real. Esta tranformação resulta na modificação do timbre sonoro.O operador proposto é um sistema ADSP que realiza operações timbrais pela manipulação de um espectro discreto contido em um vetor. A operação timbral é aplicada ao espectro do som de entrada e parametrizada por um bloco de controle externo. Este recebe e processa informações de interfaces externas e da MH. O operador timbral, ou OT, realiza todas as operações vetoriais necessárias num intervalo de tempo suficientemente curto para ser considerado pela audição como tempo-real.

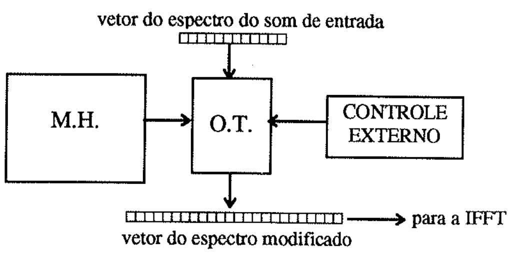


Fig. 1 Estrutura básica de um operador timbral

4. A estrutura do sistema

A entrada e saída são analógicas, sendo o processamento interno digital. Na entrada faz-se o pré-processamento e conversão A/D do sinal, s(t). O espectro de s(n) é obtido pelo algoritmo FFT para sequências de amostras com 50ms. A magnitude da saída complexa da FFT é alocada em um vetor de elementos reais, que representa o controle do operador timbral. O operador timbral manipula o vetor que representa um novo espectro, de acordo com seus parâmetros. Este novo espectro é formado pelas operações timbrais. Todas as operações timbrais ocorrem num ciclo de operação do OT, cujo intervalo de tempo é sincrono com a IFFT. Após cada ciclo do OT, o vetor de saída é escrito na MH e encaminhado para o processamento de sua IFFT e conversão D/A, gerando a saída analógica s'(n). O sistema possui um atraso entre a entrada s(t) e a saída s'(t) aproximadamente igual a $(\tau 1+\tau 2+\tau 3) \leq 50$ms.

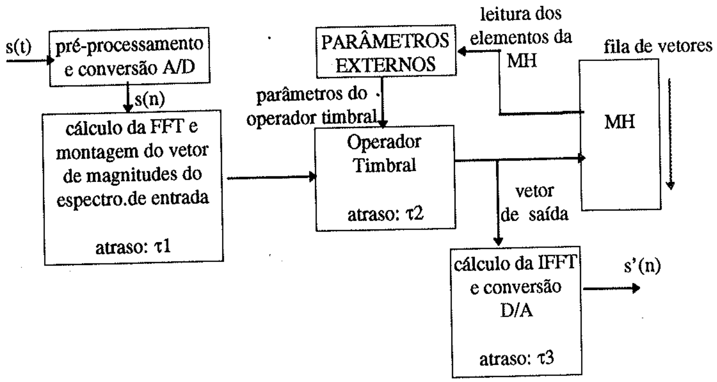


Fig. 2 **Estrutura do sistema de transformação sonora**

5. A simulação do OT

O ambiente de simulação do MATLAB 4.0 foi utilizado para o desenvolvimento de programas de simulação das transformações sonoras através de operações timbrais. Uma vez que se trata de um software, as operações timbrais não ocorrem em tempo-real.

As operações timbrais foram simuladas pelos programas descritos abaixo. Utilizaram-se os seguintes valores: taxa de amostragem Fs=8.192 Hz., pontos de entrada da FFT: N = 256, resolução do eixo da frequência: Fs/N = 32 Hz. e intervalo de tempo correspondente a um espectro: N/Fs = 31ms. Cada vetor do espectro representa a magnitude do harmônicos em N/2 = 128 pontos.

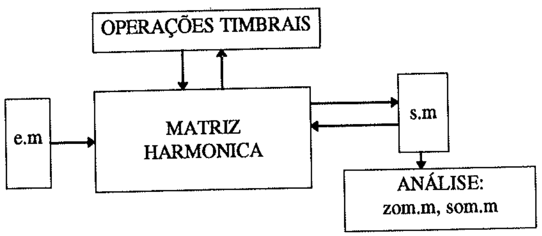


Fig. 3 **Estrutura para a simulação do operador timbral.**

<u>e.m</u>: lê um arquivo de som no padrão u-law, calcula a MH através da FFT e monta gráficos: do som e da SS. <u>s.m</u>: monta um arquivo de som através da IFFT aplicada na MH e monta gráficos da SS modificada e da envoltória temporal do som. <u>zom.m</u>: mostra um detalhe de um intervalo de tempo do som modificado e seu espectro. <u>som.m</u>:

permite a audição do arquivo de som original e modificado. As operações timbrais foram feitas na forma de programas que utilizam operações vetoriais básicas sobre as linhas de MH, tais como: adição, subtração, produto, divisão, inversão, deslocamento dos seus elementos. Através destes foram elaboradas operações timbrais, como: compressão dos níveis máximo e mínimo de superfície sonora, expansão (ou compressão) da magnitude do espectro do som, modificação da envoltória da SS, redução do número de harmônicos do espectro (preserva apenas os n maiores), modificação da *altura* (freqüência fundamental). Observa-se que a transformação sonora realizada através dos operadores timbrais permite uma grande flexibilidade de manipulação e visualização dos resultados no timbre do som. O resultado de algumas simulações feitas é mostrado a seguir.

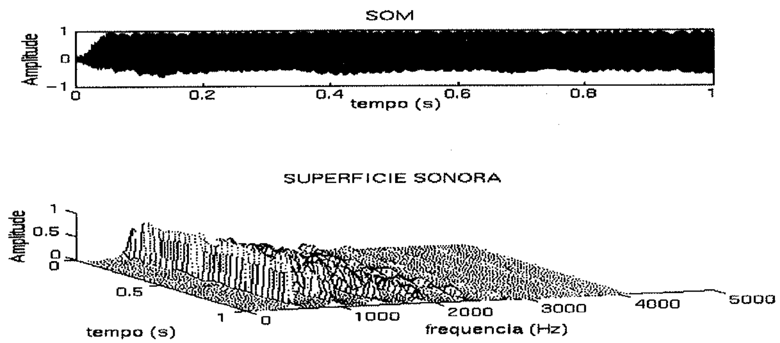


Fig. 4 Amostra de som de violino. Envoltória temporal da amplitude e SS.

Utilizou-se o som amostrado de um violino, emitindo continuamente a nota A4 (440 Hz) durante um intervalo maior que 1 segundo. Observa-se a sua envoltória no tempo e superfície sonora para o primeiro segundo de som. Nota-se a grande possibilidade de observação dos detalhes macroscópicos e microscópicos som, como: pouco ataque, presença de muitos harmônicos, centralização do harmônico de maior magnitude (fundamental) sobre o ponto do eixo da frequência correspondente a 440 Hz, variação da envoltória do espectro ao longo do tempo e oscilações da intensidade do som ao longo do tempo, entre outros.

Aplicou-se sobre a matriz harmônica desse som uma operação timbral que expande o espectro do som, na razão de 1 para 2, ou seja, os harmônicos componentes do espectro são distanciados linearmente entre si e a sua envoltória passa a ocupar o dobro do intervalo em frequência que originalmente ocupava.

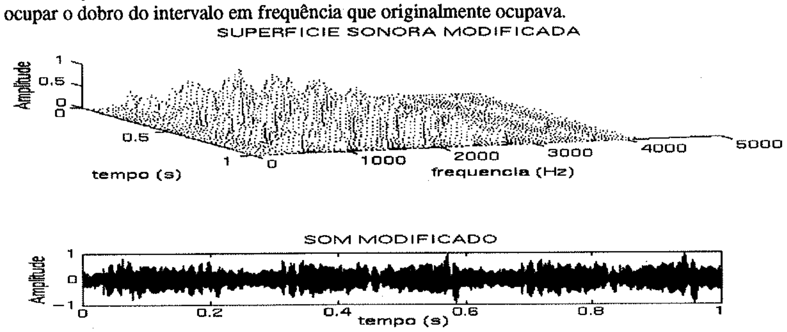


Fig.5 Violino. Transformação por expansão dos espectros (2:1). Envoltória temporal da amplitude e SS.

O resultado é mostrado tanto na forma de superfície sonora, quanto de envoltória do som no tempo. Observa-se o espalhamento do espectro, que antes ocupava uma região menor de intervalo de frequência. Passam a existir componentes em frequências mais graves e também mais agudas.

A seguir utilizou-se como arquivo de entrada o som de uma nota contínua emitida por uma flauta, por mais de 1 segundo.

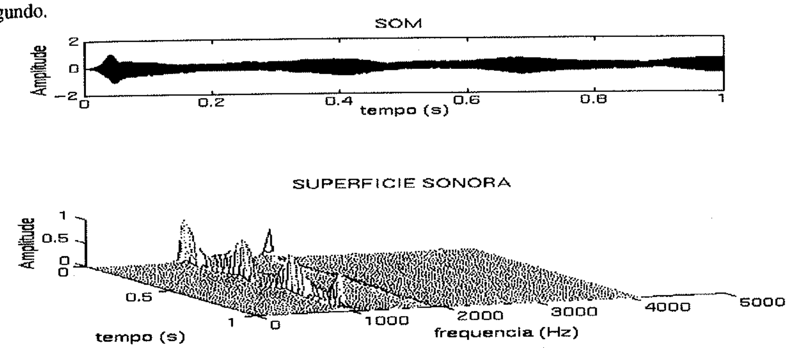


Fig. 6 Amostra de som de flauta. Envoltória temporal da amplitude e SS.

Pode-se observar uma maior quantidade de ataque e menor quantidade de harmônicos presentes do que na amostra de violino. Durante o ataque, ocorrem harmônicos que depois desapareceme que são relacionados ao som "assoprado" do instante inicial da emissão de uma nota deste instrumento.

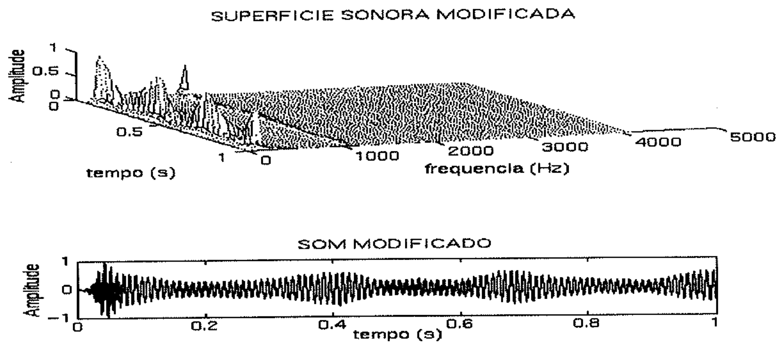


Fig.7 Flauta. Transformação por deslocamento da fundamental de 100 Hz para 200 Hz. Envoltória temporal da amplitude e SS

Aplicou-se um deslocamento linear dos harmônicos do espectro para frequências mais graves. O resultado que se observa na envoltória é que o tempo de ataque a das ondulações da intensidade do som permanecem inalterados. No entanto o som todo tornou-se bem mais grave, podendo até se observar na envoltória do tempo a periodicidade correspondente aos harmônicos de baixa frequência.

6. Resultados e possibilidades

A simulação do sistema foi feita através de programas que fazem o papel de operações timbrais, e dessa maneira, modificam os elementos contídos na matriz harmônica. Em tempo-real, as operações manipulam o espectro e armazenam o resultado na MH.

Também é importante salientar que as simulações foram feitas sobre uma superfície sonora, cujos eixos de amplitude, frequência e tempo são todos lineares. No entanto, a percepção auditiva, para a intensidade e a frequência do som, são aproximadamente logarítmicos. Portanto, as operações timbrais, programadas no sistema em tempo-real, devem levar em conta a natureza da audição humana para estas grandezas. A resolução e os limites máximo e

mínimo de cada eixo da superfície sonora deve corresponder a percepção auditiva para as variações da sua grandeza correspondente. Para o tempo, a escala é linear, com resolução menor que 50 ms. Para a amplitude, a escala é logaritmica (dB), com resolução melhor que 3dB. No caso do som amostrado, deve-se ter uma conversão A/D com resolução de 20 bits, afim de garantir SQNR = $20.\log_{10}2^{20}$ = 120 dB [Chamberlin]. Para a frequência, a escala deve ser modelada de acordo com a sensibilidade auditiva $\Delta f/f$, que equivale a 0,3% entre 500 a 4.000 Hz e aumenta (torna-se menos sensível) abaixo e acima dessa região.

Através da simulação do sistema proposto foi possível observar algumas possibilidades de transformações sonoras que as operações timbrais permitem. Deve-se levar em conta que o sistema é uma ferramenta sonora, e que as operações timbrais poderão ser programadas por software, e executadas em tempo-real. Isto mostra que as suas possibilidades de utilização são muito grandes. Pode-se citar algumas:

Composição timbral Através de um material sonoro inicial, o artísta pode criar novos timbres, tendo a SS como interface visual e pelos resultados sonoros.

Modelamento virtual do timbre O modelamento timbral pode também ser feito através de interfaces controladoras para realidade virtual, uma vez que existe a possibilidade de observação em tempo-real das transformações timbrais.

Performances artísticas Podem-se variar as transformações no timbre de um instrumento durante uma performance. Isto pode ser feito tanto por controladores que o próprio instrumentista opere, quanto por um outro artista, que as promova ou ainda por programas pre-estabelecidos em software..

Tratamentos e análises sonoras Este sistema pode ser útil para estudios de gravação e em laboratórios de análise sonora. A operação em tempo-real, dispensa o uso de grandes capacidades de memória, para o armazenamento do trecho sonoro que vai ser processado ou analisado. Todos os tratamentos sonoros, como: compressões, equalizações *inteligentes* , correções de *pitch*, detecções de padrões sonoros, podem ser feitos através da programação de operadores timbrais adequados. O som passa apenas uma vez através do sistema e na saída tem-se todas as transformações, ou detecções, executadas.

Bibliografia

Alles G. Harold. *Music Synthesis Using Real Digital Techniques*. Proceedings of IEEE, vol. 68, n.4, april. 1980.
Boldrini, José Luis. *Algebra Linear*. 3.ed. cap. 5. São Paulo : Harper & Row do Brasil. 1980.
Chamberlin, Hal. *Musical applications of microprocessors*, cap. 2. 2ed. Indianapolis, Hayden Books. 1985.
Culver, C. A. *Musical Acoustics*. 4°ed. cap. 5. McGraw-Hill. 1968.
Parker. Sybil P. *Acoustics source book*. McGraw-Hill. San Francisco. 1987.
Rosemberg, Martin E. *Sound and Hearing. Studies in Biology n,145*. cap. 2. Afd Sonologie. 1982.
Plomp, Reiner. *Aspects of the tone sensation, A psicophisiological study*. Academic Press. London, New York, San Francisco. A subsidiary of Hancourt Brace Favonovich Publisher. 1976.

PROCESSAMENTO DE SINAIS ACÚSTICOS UTILIZANDO TRANSFORMADA WAVELET

GLÓRIA MARIA FONSECA GALANTE
gloriam@omega.lncc.br
JULES GHISLAIN SLAMA
jules@serv.com.ufrj.br
COPPE - UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
PROGRAMA DE ENGENHARIA MECÂNICA
Cidade Universitária - Ilha do Fundão - Centro de Tecnologia,
Bloco G sl. 204 - Rio de Janeiro, RJ

ANGELO MAIA CISTER
cister@cos.ufrj.br
EMBRATEL
Rua da Assembléia, 10 sl. 2222 - Rio de Janeiro, RJ
&
COPPE - UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
PROGRAMA DE ENGENHARIA DE SISTEMAS
Cidade Universitária - Ilha do Fundão - Centro de Tecnologia,
Bloco H sl. 319 - Rio de Janeiro, RJ

Abstract

Our goal in this paper is to present a new mathematics tool called Wavelet Transform or Time-Frequency Analysis. We try to show both the pros and cons of this new tool against the secular Fourier Transform.
At the end of this paper the reader will be able to understand the fundamental ideas of Wavelet Transform and how this idea has spread among academics.
To motivate the reader, the first sections talks about a model that we have assumed as being easy to be implemented. In our model we have imagined any musician or any music lover that inputs the signal (music) in the computer and the computer runs the pre-processing by WT and a Neural Network recognises the pattern and the output will be the score. This kind of work already exists but it works with Fourier Transform.
We wrote a section that talk about the history of WT, its mainly mentor and other famous mathematicians who have been working with this modern tool.
Finally, we will show the necessary calculus to understand WT, but in a simple way with no hard equations. Further on we show, in a general sense, the Uncertainty Principle, which governs the size of the window. In particular, it will be observed in this article that the time-frequency, window of any Short-Time Fourier Transform is rigid, and is not very effective for detecting signals with high frequencies against WT, which presents a variable window and is capable of detecting any kind of frequency even in non-stationary signals or better, in non-continuous function.

I. INTRODUÇÃO

O objetivo deste trabalho é apresentar aspectos de uma análise tempo/tempo-freqüência de sinais através da utilização da Transformada Wavelet. É realizada, também, uma comparação com a análise tempo/freqüência obtida através da Transformada de Fourier.

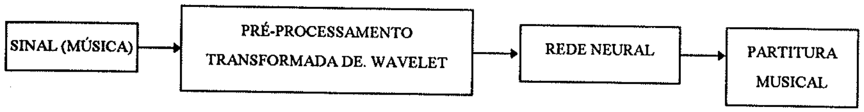
A motivação que originou este trabalho foi a observação de situações que acontecem no dia-a-dia do músico, ou de um melômano, que deseje levantar a partitura de uma peça musical; ou criar uma composição sem ter a preocupação de parar a cada instante para escrever as notas musicais.

Estas operações podem vir a ser viáveis do computador, que, ao receber como entrada a peça musical, de um único instrumento, procede o pré-processamento do sinal (a música), utilizando as transformadas citadas anteriormente, e em seguida, executa o reconhecimento de cada uma das notas musicais que integram a composição. Este reconhecimento é feito por uma rede neural previamente treinada.

O tipo de rede utilizado e como é feito seu treinamento ou como ela efetua o reconhecimento das notas não fazem parte do escopo deste trabalho, mas sim, o pré-processamento, como poderá ser efetuado e quais as vantagens e desvantagens em usar Transformada Wavelet ou Transformada de Fourier.

Como proceder?

A idéia, de modo geral, é bastante simples. Basta que tenhamos como entrada o sinal, a peça musical, feita via teclado musical ou tape-deck (aqui, mais uma vez, admitimos uma peça musical executada por um único instrumento), e que o computador possua um software residente que execute um pré-processamento do sinal e que sua saída seja a informação necessária para o reconhecimento das notas, através de uma rede neural. A figura abaixo ilustra o processo.



Por que fazer um pré-processamento?

A dificuldade principal em se trabalhar no domínio temporal é o chamado alinhamento temporal (time-warping), onde determinados sinais que contenham a mesma informação, possam ser analisados em tempos diferentes. Por isso a necessidade da troca do domínio temporal para o domínio frequencial, isto é, alguma espécie de transformada é necessária para analisar o sinal na freqüência para eliminação desta espécie de problema, ou seja, os padrões iguais (templates) possuem semelhanças entre si, com isso a probabilidade do reconhecimento é muito grande.

II. ANÁLISE DO SINAL

Histórico

A Transformada de Fourier foi desenvolvida ainda no século XIX. Desde então, a Transformada de Fourier tem sido usada como o principal instrumento para análise de sinais tendo se tornado o carro chefe para a moderna técnica de análise de sinais e tem se mostrado incrivelmente versátil. Além disso, qualquer sinal suficientemente bem comportado pode ser escrito como uma combinação infinita de funções senoidais e cosenoidais.

Todavia, tal transformada sofre certas limitações, por exemplo, sinais transientes ou que apresentem algum tipo de descontinuidade não se comportam de maneira adequada para esse tipo de análise.

Na tentativa de se suprir estes problemas, outras técnicas surgiram, entre elas a Transformada de Fourier de Janela Deslizante (Short-Time Fourier) e a Transformada Wavelet, sendo que esta última será o tema central deste trabalho.

A transformada Wavelet "TW" (WAVELET) é uma técnica recente de processamento de sinal introduzida por J. Morlet, geofísico francês, que em meados dos anos 80, desenvolveu esta técnica a partir dos estudos de Gabor, que introduziu pela primeira vez uma gaussiana como janela deslizante (função analisante).

O campo de aplicação da TW é muito extenso e se estende ao processamento de medidas em mecânica dos sólidos (choques, vibrações, transientes) ou em mecânica dos fluídos (turbulência, escoamentos multifásicos, acústica, etc); na área biomédica (eletroencefalogramas e eletrocardiogramas); análise de imagens; processamento de sinais e geofísica.

Outros matemáticos de renome mundial também vêm desenvolvendo trabalhos utilizando a TW como instrumento poderoso para suas pesquisas. Entre estes podemos citar Yves Meyer (com seus estudos em análise harmônica), Sthefane Mallat, Ingrid Daubechies e outros.

III. TÉCNICAS DE ANÁLISE

Transformada de Fourier

Sinais temporais são analisados frequentemente por meio da Transformada de Fourier que estabelece uma correspondência biunívoca entre o domínio temporal e o frequencial.

A Transformada de Fourier (TF) é dada pela expressão:

$$\hat{S}(f) = TF[s(t)] = \int_{-\infty}^{\infty} s(t)e^{-i2\pi ft}dt \tag{1}$$

onde s(t) é o sinal que está sendo analisado e TF é uma função da freqüência f.

Além da correspondência biunívoca entre os domínios temporal e frequencial, a TF pode ser considerada como uma medida de quanto um sinal s(t) é "semelhante" às funções teste $\cos(2\pi ft)$ (parte real) e $\text{sen}(2\pi ft)$ (parte imaginária) já que, a cada freqüência, a TF é igual ao valor médio do produto do sinal pela exponencial complexa.

Visto sob este ângulo, devido a média ser tomada sobre um intervalo de tempo $(-\infty,+\infty)$, a TF apresenta o inconveniente de não permitir identificar (localizar no tempo) uma semelhança do sinal com a função teste, se essa semelhança está limitada a um intervalo pequeno de tempo (Figura 1).

Concluindo, a TF não se adapta bem à análise de sinais não estacionários. Como exemplo, podemos citar a TF de um Delta de Dirac (função impulso) que apresenta um espectro de amplitude constante, se estendendo de $-\infty,+\infty$. A partir da TF do Delta de Dirac não é possível localizar diretamente a sua posição temporal.

Para suprir esta deficiência, surgiu a técnica da Transformada de Fourier de Janela Deslizante (TFJD), descrita a seguir.

Transformada de Fourier de Janela Deslizante

A TFJD é uma representação tempo/freqüência de um sinal temporal e apresenta as seguintes características:

- o sinal a ser analisado é ponderado por uma função janela de suporte finito (a função possui valores nulos fora de um determinado intervalo);
- o sinal ponderado é submetido à TF que fornece um espectro associado ao instante de aplicação da janela; e,
- a janela é deslocada sobre o sinal e o processo de análise é recomeçado.

Deste modo, a TFJD fornece uma sucessão de espectros que permitem acompanhar a evolução do conteúdo frequencial do sinal e, portanto, perceber dentro de uma certa medida, as não estacionaridades do sinal, quando presentes (Figura 2).

A TFJD é uma função do tempo e da freqüência e é expressa por:

$$S_j(t,f) = TFJD[s(t)] = \int_{-\infty}^{\infty} s(t')j(t'-t)e^{-i2\pi ft'}dt' \tag{2}$$

onde j(t'-t) é a função janela deslizante.

Podemos observar que a expressão (2) expressa uma convolução do sinal s(t) com a janela $j(t)e^{-i2\pi ft}$.

Ainda que útil, a TFJD apresenta certos incovenientes, como:

- não existe transformada inversa (senão por partes);
- as informações de fase entre os espectros são perdidas;
- conduz frequentemente a uma amostragem muito densa do sinal temporal para permitir perceber flutuações rápidas do conteúdo frequencial; e,
- ela não permite uma localização temporal equilibrada para todas as freqüências.

O efeito da média temporal sobre um intervalo de tempo considerado introduz um "bias" na localização temporal das "freqüências instantâneas", tanto maior quanto a freqüência de análise aumenta.

Durante a duração da janela, na medida de "semelhança" entre o sinal temporal e a função teste, a freqüência sobre um número de oscilações é diferente da função teste.

Transformada Wavelet

A Transformada Wavelet (TW) resulta da preocupação de se obter um equilíbrio entre a localização temporal e a localização frequencial do sinal. Ela é uma transformação tempo/freqüência e tem por objetivo identificar, no domínio temporal, o conteúdo frequencial de um sinal. Tal abordagem é muito útil na análise de sinais não estacionários ou transientes (Figura 3).

A preocupação de Morlet foi obter uma função teste (função analisante ou wavelet) que permitisse uma localização equilibrada, que não privilegiasse um domínio em detrimento do outro, o que significa (visto sob o ângulo da medida de "semelhança" do sinal com a função teste) que a função teste deve ser localizada no tempo e ter a mesma morfologia qualquer que seja a freqüência e que independente de sua forma, ela deve possuir sempre o mesmo número de oscilações no domínio temporal. Isto é obtido fazendo-se a janela, além de ser deslizante, ter sua largura "a" alterada de modo a manter o mesmo número de oscilações, independente de qual faixa de freqüência está sendo analisada.

A TW é expressa por:

$$S(a,b) = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} s(t)g\left(\frac{t-b}{a}\right)dt \tag{3}$$

onde a função g é chamada função analisante ou wavelet.

As wavelets são funções simples que permitem uma representação simples, eficaz e robusta dos mais diversos sinais através da TW, que pode ser interpretada como uma decomposição em série do sinal, sendo as wavelets a base para esta decomposição.

Morlet utilizou uma gaussiana multiplicada por uma função harmônica complexa como janela, dada pela fórmula a seguir:

$$g(t) = e^{-t^2/2}.e^{-i2\pi f_0 t}$$

Posteriormente, trabalhos de matemáticos (Y. Meyer e P. G. Lemaire, entre outros) permitiram mostrar que existem outras formas de wavelets que possuem propriedades de ortogonalidade que as de Morlet não possuíam.

Propriedades da TW

A TW apresenta algumas propriedades importantes descritas a seguir:

a) Linearidade:

$TW[\alpha s(t) + \beta y(t)] = \alpha TW[s(t)] + \beta TW[y(t)]$

b) Translação no tempo:

Se $TW[s(t)] = S(a,b)$, então:

$TW[s(t-t_0)] = S(a,b-t_0)$

c) TW em freqüência:

$$S(a,b) = a^{1/2}\int_{-\infty}^{\infty} e^{i2\pi bf}\,\hat{G}(af)\,\hat{S}(f)\,df$$

d) Relação de Energia da TW:

A energia do sinal na TW é dada por:

$$E = \frac{1}{c_g}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|S(a,b)|^2\frac{1}{a^2}\,dadb$$

onde c_g é uma constante positiva definida por:

$$c_g = \int_{-\infty}^{\infty}\frac{\hat{G}(af)\,\hat{G}^{*}(af)}{a}\,da$$

e) TW Inversa:

A TW admite uma transformada inversa, porém não de forma biunívoca, isto é, existem várias fórmulas de reconstrução do sinal. Abaixo é dada uma delas:

$$s(t) = \frac{1}{c_g}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{S(a,b)}{a^{3/2}}\,g\left(\frac{t-b}{a}\right)dadb$$

PRINCÍPIO DA INCERTEZA

Neste momento, vale a pena "abrirmos um parênteses" para a definição de alguns parâmetros importantes no tratamento de sinais não estacionários.

Consideremos uma função analisante g(t) simétrica, de suporte finito e que tenda rápidamente para zero em seus extremos.

Define-se por t_c o centro temporal da wavelet g(t) e por Δ_g seu comprimento radial, ambos dados por:

$$t_c = \frac{1}{|2\pi f|^2}\int_{-\infty}^{\infty} t\,|g(t)|^2\,dt$$

$$|\Delta_g|^2 = \frac{1}{|2\pi f|^2}\int_{-\infty}^{\infty}(t-t_c)^2\,|g(t)|^2\,dt$$

Desde que Δ_g corresponde ao raio da função analisante, temos que a largura da wavelet é dada por $2\Delta_g$.

Sendo assim, a transformada Wavelet, dada por (3), localiza o sinal s(t) com uma janela no tempo situada no intervalo: (4)

$[b+at_c-a\Delta_g,\ b+at_c+\Delta_g]$

e centrada em $b+at_c$ com largura de $2a\Delta_g$. A isto denominamos *localização no tempo* do sinal s(t).

Suponhamos, agora, que a transformada de Fourier da função analisante dada por $\hat{G}(f)$ tenha seu centro frequencial em f_c e raio $\Delta_{\hat{G}}$.

Consideremos a função analisante $\hat{G}_1(f)$ dada por:

$$\hat{G}_1(f) = \hat{G}(f-f_c)$$

Assim sendo, a transformada Wavelet de um sinal $\hat{S}(f)$ se torna:

$$S(a,b) = a^{1/2}\int_{-\infty}^{\infty} e^{i2\pi bf}\,\hat{S}(f)\,\hat{G}_1(af)\,df = a^{1/2}\int_{-\infty}^{\infty} e^{i2\pi bf}\,\hat{S}(f)\,\hat{G}\left(a\left(f-\frac{f_c}{a}\right)\right)df \qquad (5)$$

A transformada Wavelet dada por (5) localiza a informação do espectro $\hat{S}(f)$ do sinal s(t) com uma janela na freqüência situada em:

$$\left[\frac{f_c}{a}-\frac{1}{a}\Delta_{\hat{G}},\ \frac{f_c}{a}+\frac{1}{a}\Delta_{\hat{G}}\right] \qquad (6)$$

cujo centro está em f_c/a e com largura $2\Delta_{\hat{G}}/a$. Isto é chamado *localização na freqüência*.

Ao combinarmos as quantidades (4) e (6) para uma análise tempo-freqüência usando a transformada Wavelet, obtemos a janela:

$$[b+at_c-a\Delta_g,\ b+at_c+\Delta_g] \times \left[\frac{f_c}{a}-\frac{1}{a}\Delta_{\hat{G}},\ \frac{f_c}{a}+\frac{1}{a}\Delta_{\hat{G}}\right]$$

cuja largura é constante e igual a $4\Delta_g\,\Delta_{\hat{G}}$.

O resultado acima monstra a diferença principal entre a TFJD e a TW. Na primeira são usadas janelas de largura constante porém com freqüências variáveis. Deste modo, a localização temporal é constante para qualquer freqüência analisada. Assim, sinais com variações menores que a resolução temporal dada por esta localização não são detectados.

Já na TW, ao se variar a largura da função analisante no tempo, altera-se sua localização temporal, alterando-se também a localização frequencial, de modo que a área da janela permanece constante independente da freqüência analisada, sendo que para altas freqüências temos uma largura temporal estreita e vice-versa.

Conclusão

Foi apresentada uma proposta de trabalho envolvendo a transformada Wavelet como elemento de pré-processamento para a escrita automática da partitura. Esta transformada já foi utililzada em outra oportunidade em Acústica Musical para análise e modificação de sinais musicais nos laboratórios do LM, Marseille, e esperamos que a associação desta transformada com a utilização de redes neurais permita enriquecer ainda mais as ferramentas a disposição do músico.

Referências

Bonaldo, André Vinicius (1993). *Transformada Ondelete, Algoritmo Baseado na Transformada Rápida de Fourier*. Tese de Mestrado, COPPE/UFRJ - Programa de Engenharia Mecânica.

Daubechies, Ingrid (1990). The Wavelet Transform, Time-Frequency Localization and Signal Analysis. *IEEE Transactions on Information Theory*, 36(5), 961-1005.

Kroland-Martinet, R. (1984). *Analysis of Sound Pattern trough Wavelets Transforms*.

Schlegel, R. (1989). *Post-Traitement de Resultats d'Essais et de Calcul par la Methóde des Ondelettes*. 26º Colloque d'Aérodynamique Appliquée - Toulose.

Slama, Jules Ghislain. *Ondelettes - La Transformée on Ondelettes Continue*. Notas de aula.

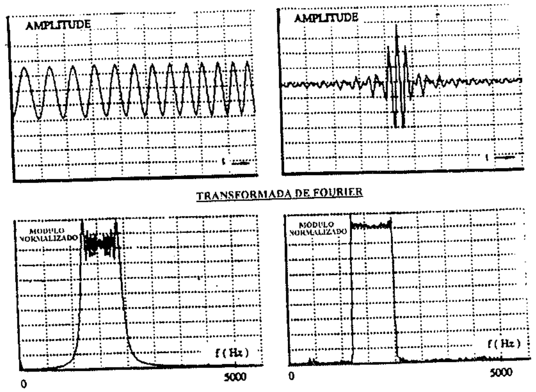


FIGURA 1

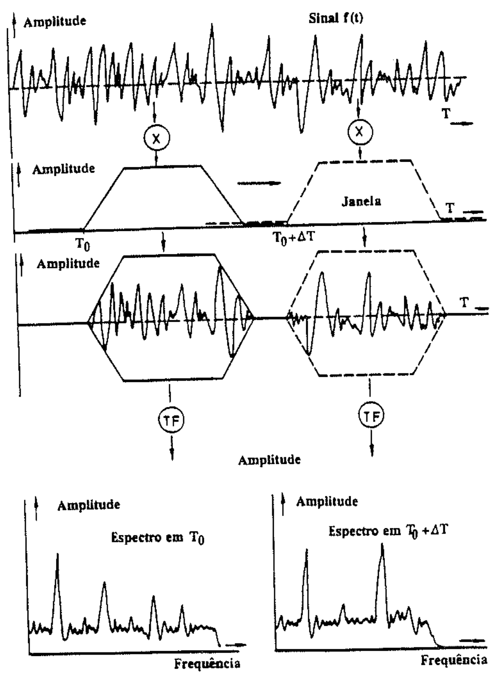
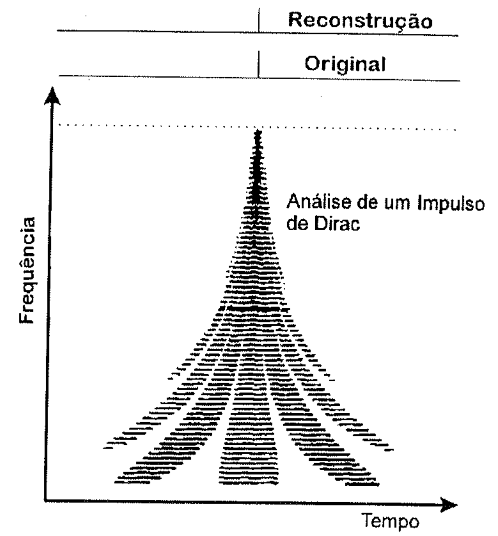
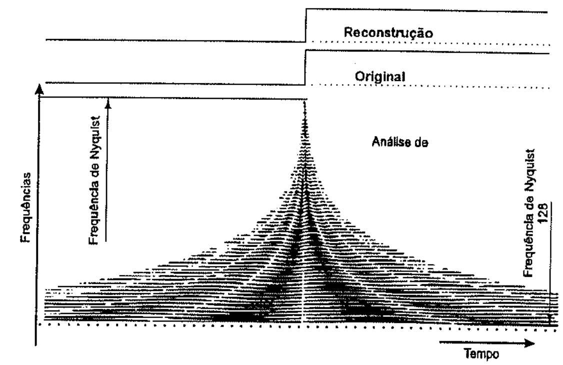


FIGURA 2



(a)



(b)

FIGURA 3

Wavelets as a Multiresolution Analysis and Synthesis Technique for Sound Timbres Edition

REGIS ROSSI ALVES FARIA
Computer Music Group
LSI - Laboratório de Sistemas Integráveis
Escola Politécnica da Universidade de São Paulo
(regis@lsi.usp.br)
JOÃO ANTÔNIO ZUFFO
LSI - Laboratório de Sistemas Integráveis
Escola Politécnica da Universidade de SãoPaulo
(jazuffo@lsi.usp.br)
Av. Professor Luciano Gualberto, 158, Tv.3
05508-900 São Paulo - SP Brasil

Abstract

Real acoustic instruments' sound pureness still stands an appreciable distance ahead of the artificial instruments' one, as synthesizers, in timbre quality as much as in the virtually infinite variability and modulation possibilities.
Analysis and synthesis techniques have been employed for years as a tool to analyse timbres' temporal-spectral characteristics, and re-synthesize them from these extracted parameters.
In this paper we show the utilization of Wavelets - a mathematical tool employed in signal processing with additional advantages over the classic Fourier transforms - as a technique for sound analysis, edition and synthesis under a multiresolution scheme. The central idea is to explore the capacities of this technique in altering fine timbre components in the time-frequency (scale) spaces, in order to produce improved timbres.

Introduction

Several analysis/synthesis techniques for signal processing have been employed through years now in the field of music. Successive development of computational resources -both in hardware and software- adding to a growing interdisciplinary interaction with other knowledge branches (as applied mathematics, biology and others) propitiate the emergency of efficient and sophisticated techniques to process several types of signal - including musical ones.We introduce wavelets as one these emergent tools, coming from the marriage of several mathematical branches and engineering and computational techniques.

So far the realism and sound pureness produced with real-world musical instruments have not been surpassed: the infinite number of physical parameters, and also the musicians' psychological impressions, turn the musical experience into a complex phenomenon, hard to be efficiently modeled in computer applications.

One way towards the enrichment of music representation on computers is the processing of the musical signals in both time and frequency domains in order to work out instruments' timbres - whether they are acoustical or not. Acoustic instruments have complex tones. The spectral evolution in an attack reveals different evolution in time of the various harmonics. Even at the "steady-state" the tone undergoes many micromodulations and other variations that are perceptible. Also at the end of a note the harmonics do not decrease uniformly: each frequency component reveals a different collapse in time. The musician in stage imposes several modulations to the sound. His or her style and mood are translated into complex modulations and different intensities during performance. Add to this the influence of local acoustics, giving birth to effects like reverberation, etc. The final sound is a signal full of details hidden in its time evolution.

One can easily sense large variations in intensity and recognize major pitches. One can also recognize timbres, sense some vibratos and other effects. However, a human listener often interprets these parameters in an emotional context, where feelings are translated into a great spectral variability in time, e.g, fine modulations and "micro-structures" mixed inside the signal.

It happens that these details are present inside the signal at several different scales. As in a map, one can see large areas at larger scales, and details at smaller scales. It is desirable an unbounded resolution signal edition system, where changes can be made over the signal at different scales.

The wavelet framework is adequate for this purpose. We show how wavelets can be used to implement a multiresolution signal edition system, dealing with the signal at different scales, and in an efficient and concise manner.

Wavelets

The application of wavelets to signal processing is only a few years old. The theory is based on representing generic functions in terms of basic building blocks. This is an "atomic decomposition" algorithm, as the Fourier transform.

Looking back over the history we find seven different origins of wavelets: the idea of focusing a signal over different scales came up independently in many fields of physics, mathematics and engineering. In 1984, Grossman and Morlet introduced an unified framework, giving birth to the first definition of a wavelet. Mallat and Meyer (1986) formulated the multiresolution analysis, a natural framework for understanding and constructing wavelet bases. From then on the number of contributions and applications of this theory has grown substantially.

Fourier techniques are an ideal tool for studying stationary signals, decomposing them into linear combinations of trigonometric waves (sines, cosines). Musical signals may be classified as nonstationary signals, where transients and other non predictable events might happen, and the wavelets techniques are an ideal tool for studying such signals, decomposing them into linear combinations of wavelets.

Wavelet Representation

Wavelets consist in a family of basis functions $\psi_{j,k}$, in $L^2(\mathbf{R})$, obtained from a single mother-wavelet ψ by dilations (scaling) and translations (shifts).

$$\psi_{j,k} = 2^{j/2}\psi(2^{j/2}t-k) \qquad \text{,where } j,k\in\mathbf{Z}.$$

There are infinite possible families like this, and the usefulness of them is linked to some desirable properties they must possess. The mother wavelet ψ should verify some important properties: be a localized pulse that decreases to zero and has integral zero. Also, it should verify the admissibility condition (C_ψ) below:

$$C_\psi = \int_0^\infty |\psi(\xi)|^2 . |\xi|^{-1}. d\xi < \infty \quad \text{and} \quad \int_{-\infty}^{\infty} \psi(t).dt = 0$$

where $\Psi(\xi)$ is the Fourier transform of ψ.

The family $\{ \psi_{j,k} \}$ above is an orthonormal basis of $L^2(\mathbf{R})$. That is $<\psi_{j,k} , \psi_{l,m}> = \delta_{j,l} .\delta_{k,m}$ $(j,k,l,m\in\mathbf{Z})$, and every function $f(t) \in L^2(\mathbf{R})$ can be written as

$$f(t) = \sum_j \sum_k d_{j,k}\psi_{j,k}(t) \qquad \text{(S)}$$

where the wavelet coefficients $d_{j,k}$ are given by

$$d_{j,k} = < f(t), \psi_{j,k}(t) > \qquad \text{(A)}$$

We are interested in wavelet functions whose binary dilations and dyadic translations are sufficient to represent all functions $f(t)$ in $L^2(\mathbf{R})$. Observe that the family $\{\psi_{j,k}\}$ covers infinite scales $a=2^{-j}$, and performs a frequency band analysis when generating the wavelet coefficients. We have frequencies separated in consecutive octaves (2^j), a natural scaling factor in music.

Wavelet Transform

Unlike Fourier analysis, the integral form of the wavelet transform is intimately related to the wavelet series representation: the coefficients $d_{j,k}$ of $f(t)$ are precisely the values evaluated by the integral wavelet transform (not shown) at the dyadic positions in the binary dilated scales. The wavelet representation simultaneously localizes f and its Fourier transform with the multiscale analysis capability. Since there are real-time algorithms for calculating the coefficient sequences and for recovering f from these sequences, we will center our attention in the discrete signal analysis and synthesis through discrete wavelet transforms.

The formulas in equations (S) and (A) are a simplified form of the synthesis and analysis processes, respectively.

Timbres Edition

An unbounded resolution signal edition system should support:

- an efficient representation of the signal at different scales. *It should be possible to "see" details at different scales.*
- operation on the signal at different scales. *It should be possible to process the signal sequence (at scale a) with known processing methods -as filtering, modulation, addition and subtraction of other signal sequences, applying envelopes, etc.*
- propagation of changes. *Changes to the signal on a level (scale) must propagate to other levels, in a non-redundant manner, and without loss of information. In other words, levels must be connected.*

A multiresolution analysis (and synthesis) approach is the choice for implementing the above processing structure. In the next session, we introduce the concepts of a multiresolution analysis framework, and its properties. It is presented the pyramidal algorithm, and its implementation with filter banks. Thus a natural connection between filter trees in discrete time and the multiresolution in continuous time is made, showing that filter trees can implement multiresolution analysis. Finally, it is shown how filter trees lead to wavelets.

Multiresolution Analysis

A multiresolution analysis consists of a sequence of successive approximation (closed) spaces V_j. Each subspace V_j is contained in the next subspace V_{j+1}. A function in one subspace is in all higher (finer) subspaces:

$$\ldots V_{-1} \subset V_0 \subset V_1 \subset \ldots \subset V_j \subset V_{j+1} \subset \ldots$$

A function $f(t)$ decomposed into these spaces has a piece in each subspace. The piece (projection of $f(t)$) in V_j is called $f_j(t)$. The union of all subspaces is $L^2(\mathbf{R})$, and the intersections between subspaces is a null space ($\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$). There are additional requirements:

- Completeness: $f_j(t) \to f(t)$ as $j \to \infty$, and Emptiness: $\| f_j(t) \| \to 0$ as $j \to -\infty$
- V_{j+1} consists of all rescaled functions in V_j: $f(t) \in V_j \Rightarrow f(2t) \in V_{j+1}$
- Shift invariance: $f(t) \in V_j \Rightarrow f(t - 2^{-j}.k) \in V_j$
- There exist a basis for each subspace V_j: $\{\phi_{j,k}, k\in\mathbf{Z}\}$ is an orthonormal basis **for** V_j, $j\in\mathbf{Z}$.

We call ϕ the "*scaling function*" of the multiresolution analysis.

The function $f_{j+1}(t)$ in V_{j+1} has a better resolution than f_j in V_j. The missing portion necessary to aproximate $f_{j+1}(t)$ from f_j is in a new subspace W_j: $\Delta f_j = f_{j+1}(t) - f_j$, where $\Delta f_j \in W_j$. From the subspace point of view, $V_j \oplus W_j = V_{j+1}$. The new family of subspaces are the orthogonal complement of V_{j+1} in V_j. It follows that

$$V_{j+1} = W_j \oplus W_{j-1} \oplus W_{j-2} \oplus \ldots \quad \text{or} \quad V_{j+1} = W_j \oplus W_{j-1} \oplus W_{j-2} \oplus \ldots \oplus V_0, \quad \text{which implies that}$$

$$f_{j+1}(t) = \Delta f_j + \Delta f_{j-1} + \ldots + \Delta f_1 + \Delta f_0 + f_0$$

Naturally, it follows that the union of all subspaces W_j is also the whole space $L^2(\mathbf{R})$, and the requirements above also apply to the family of (closed) subspaces W_j. The family of functions $\{\psi_{j,k}, k\in\mathbf{Z}\}$ constitutes an orthonormal basis for W_j. More: The whole collection $\{\psi_{j,k}, j,k\in\mathbf{Z}\}$ constitutes an **orthonormal basis** for $L^2(\mathbf{R})$, which is called **a wavelet basis** of $L^2(\mathbf{R})$, with $\psi_{j,k}(t) = 2^{j/2}\,\psi(2^j t - k)$ (to maintain the coherence with $\phi_{j,k}$).

The structure that connects the subspaces V_j to W_j is a pyramid, as below:

$$\begin{tikzcd}[column sep=small]
V_j \arrow[r] \arrow[dr] & V_{j-1} \arrow[r] \arrow[dr] & V_{j-2} \arrow[r] \arrow[dr] & V_{j-3} \arrow[r] \arrow[dr] & \ldots \\
& W_{j-1} & W_{j-2} & W_{j-3} & \ldots
\end{tikzcd}$$

We begin our calculations at some unit scale. The wavelets are a basis for the whole space $L^2(\mathbf{R})$, but the scaling function ϕ at $j=0$ and the wavelets with $j\geq 0$ are a more practical basis. We can recover a $f(t)$, decomposed into a set of subspaces V_j and W_j:

$$f(t) = f_0(t) + \sum_{j=0}^{+\infty} \Delta f_j = \sum_{-\infty}^{+\infty} \Delta f_j = \sum_{-\infty}^{+\infty} d_{j,k}\psi_{j,k}$$

where $d_{j,k}$ **are the wavelet coefficients** of $f(t)$. From $f_0(t)$ (level V_0) and we extract the other $f_j(t)$ from mathematical operations over the pyramid. We can stop at some scale, say 2^{-J} (level J, where lies $f_J(t)$) with enough high frequencies components (finer) to reproduce an exact signal. In the opposite direction (down the pyramid), decomposing f_0 into successive coarser aproximations, we obtain less resolved descriptions of $f(t)$, at larger scales -as in a map.

The Wavelet decomposition and reconstruction algorithms

We need to have our signal $f(t)$ described at different scales. It is desirable the ability to go from a coarse aproximation of $f(t)$ towards a finer one, where more details are available (better resolution), and vice-versa, and perform operations on the signal at chosen scales. The multiresolution framework offers the environment to accomplish these operations. In this scheme, projections of $f(t)$ into subspaces V_j and W_j are related by:

$$f_J = f_{J-1} + g_{J-1}, \quad \text{and by iteratation follows that} \quad f_J = g_{J-1} + g_{J-2} + \ldots + g_0 + f_0.$$

There is a intimate relation between $\phi(t)$ and $\phi(2t-k)$ and between ψ and $\phi(2t-k)$ known as the *two-scales relation*:

$$\phi = \sum_n h_n \phi_{1,n} \quad \text{and} \quad \psi = \sum_n g_n \psi_{1,n}$$

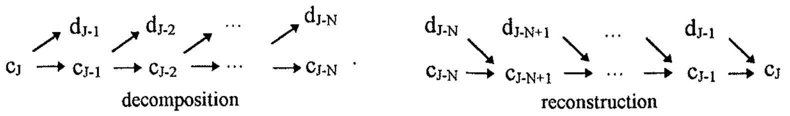
since $\phi \in V_0 \subset V_1$ and $\psi \in W_0 \subset V_1$. From these relations we derive the decomposition formulas:

$$c_{j-1,k} = \langle f, \phi_{j-1,k} \rangle = \sum_n h_{n-2k} c_{j,n} \quad \text{and} \quad d_{j-1,k} = \langle f, \psi_{j-1,k} \rangle = \sum_n g_{n-2k} d_{j,n}$$

We can define now f_j and g_j as $f_j = \sum_k c_{j,k}\phi_{j,k}$ and $g_j = \sum_k d_{j,k}\psi_{j,k}$. It is clear that $C_{j,k} \in V_j$ and $d_{j,k} \in \mathbf{W}_j$. Since $f_j = f_{j-1} + g_{j-1}$, the reconstruction algorithm is

$$c_{j+1,k} = \sum_n [\overline{h_{k-2n}}c_{j,n} + \overline{g_{k-2n}}d_{j,n}]$$

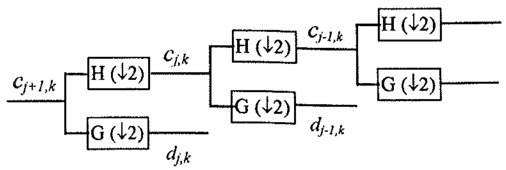
This is **wavelet decomposition** and **reconstruction**, which can be schematize in a analogous pyramid algorithm:



It is the recursive nature of wavelets algorithms that make them computationally fast and efficient.

Filter banks implementation

A multiresolution pyramid analysis can be implemented with filter banks, structured in a tree format, as below:



H is a low-pass filter, which computes averages. G is a high-pass filter, which computes differences. The downsampling steps (↓2) get even numbered components from the input sequence. The averages and downsampling go on indefinitely, each step taking us from a finer level to a coarser one (as in the multiresolution step from V_j to V_{j-1}). In real computations we can start from a fine scale 2^{-J} at level J and go down the tree towards level $j=0$, which, for example, can be normalized with $\Delta t=1$. If the input vector x(n) has lenght $N=2^J$, we reach level $j=1$ with 2 inputs, almost the coarsest level. In the filter tree, the input sequence x(n) corresponds to the coefficients of $c_{J,k}$.

In practice, we can assume the values of signal sequence $f(n)$ as the coefficients $c_{J,k}$ (actually f_J), and the analysis will provide the next level coefficients $c_{j,k}$ and $d_{j,k}$.. This last contains the details of $f(n)$ separated into band-frequencies. In the synthesis, we invert the processing direction, as in continuous multiresolution synthesis, the only difference being in executing an upsampling (↑2) instead of a downsampling from a level to another.

The operations in a filter tree, as above, can be conveniently implemented through matrices multiplications.To transform N-lenght input sequence to its N coefficients (generated in the decomposition) it is necessary a NxN matrix. It is like solving a linear system. The inverse transform operations (the synthesis) involves the inverse matrix:

<u>Synthesis</u>: $\mathbf{x} = \mathbf{W}.\mathbf{b}$ <u>Analysis</u>: $\mathbf{b} = \mathbf{W}^{-1}.\mathbf{x}$

where **b** is the vector of the N coefficients, **W** is the wavelets matrices obtained from the filter banks coefficients, and **x** is the input vector (signal sequence).

The choice of the high-pass and low-pass filters will exert a strong influence on the properties verified by these matrices. For example, if we have an orthogonal filter bank (with orthogonal H and G) the correspondent

(filter bank) matrix will be orthogonal. By means of a proper normalization, the orthogonality turns into orthonormality, and, thus, $\mathbf{W}^T.\mathbf{W} = \mathbf{I} \Rightarrow \mathbf{W}^{-1} = \mathbf{W}$. This turns the transform into a *fast transform*, because these matrices can be factorized into 2 or 3 matrices, with many zeros entries, and the number of numerical operations can be dramatically reduced. Actually, it can be proved that the number of multiplications for the *fast wavelet transform* is bounded to less than 2.T.N, where T is the number of the filter coefficients. In other words, the algorithmic complexity is O(N).

Now, let's show the bridge that leads filter trees to wavelets-based multiresolution schemes.

There are many parallels between a filter tree in discrete time and a pyramidal multiresolution in continuous time:

Filter banks (discrete time)	Wavelet multiresolution scheme (continuous time)
filter bank tree	multiresolution pyramidal structure
downsampling ($v(n)=y(2n)$, $\omega \to \omega/2$)	rescaling $t \to 2t$
lowpass filter	averaging with $\phi(t)$
highpass filter	detailing with $\psi(t)$
orthogonal matrices	orthogonal basis
analysis bank output	wavelet coefficients
synthesis bank output	sum of wavelet matrices
product of filter matrices	fast wavelet transform

The construction of a wavelet basis was connected previously to the existence of a scaling function ϕ. It is now appropriate to show the connection between the low-pass filter choice and the scaling function.

Dilation and wavelet equations

The low-pass filter coefficients $c_{j,k}$ (n) are the link that leads to wavelets. The operation {H(↓2)} in the pyramid algorithm might be, in theory, executed indefinitely. It consists of a recursive operation. Suppose that ϕj,k is one basis at level j in the filter banks. At level j-1 the basis is ϕj-1,k, as if it was computed by a filtering/downsampling operation. It is a two-scale relation, required for multiresolution analysis:

$$\phi^{(i+1)}(t) = \sum_k h(k)\phi^{(i)}(2t-k)$$

where $(i+1)$ and (i) indicates a recursive calculus. This is called the cascade algorithm. If those functions $\phi^{(i)}$ converge as $i \to \infty$ take the limit of the iteration, which is the *dilation equation*:

$$\phi(t) = \sum_k h(k)\phi(2t-k)$$

This, in multiresolution language, means that the space V_0 is contained in V_1. The wavelet subspace W_o is also in V_1, and there exist a similar relation connecting $\psi(t)$ and $\phi(2t-k)$, but this time through the high-pass filter portion, i.e. the second channel in the filter bank. This is called the *wavelet equation*:

$$\psi(t) = \sum_k g(k)\phi(2t-k)$$

This is how filter banks leads to wavelets representation. The trick in constructing a wavelet-basis is in the choice of the filters. Not all filters leads to wavelets. The filter must verify some important properties in order to be useful. The orthogonality theorem says that if the cascade algorithm converges, and if the coefficients c(k) and k(k) come from an orthogonal filter bank, then they lead to an orthonormal basis $\phi_{j,k}$ and an orthonormal wavelets basis $\psi_{j,k}(t)$. Generally, if H($\omega=\pi$) = 0, there is convergence of $\phi^{(i)}(t)$ to $\phi(t)$, and when $|H|^2$ is halfband the $\phi(t)$ is orthogonal to its translates.

There are numerous other conditions and special properties of some filters that lead to specific wavelets representations, but it is not our aim to go further in mathematical and filter banks issues for the moment.

One only has to keep in mind that we can construct wavelets using filter banks, and as we can conceive infinite types of (qualified) filters, it is possible to obtain infinite wavelets basis. This is one advantage over Fourier techniques, which only take into account trigonometric sines and cosines.

Conclusions

We think that a wavelet-based multiresolution analysis and synthesis environment is an efficient framework to examine musical signals, inherently non-stationary, and possessing finite energy. One attractive is the possibility of extrapolating the higher resolution limit, creating a "detail" at the top-most level (in W_J) and expanding to a new higher level. A modification in the signal at level j should affect only the elements in W_j, since through the wavelet basis this change will be propagated onwards.

A detail in W_j can be created by processing the sequence that represents the signal in this level with known signal processing techniques. Wavelets operations are feasible in real-time. Real-time sound signals edition, however, is dependent on the efficiency of the editions operations itself.

One important point to stand out is on the choice of the filter bank/wavelet basis. Some wavelets are better than others to treat a specific signal type. For example, an algorithm that is excellent for data compression can be a disaster when applied for analysis. A large portion of the work lies in the research of an optimal basis for the type of signal that will be processed. Performances of different algorithms should be compared, as in benchmark tests.

In the case of speech and music, since quality judgments are greatly influenced by "human-factors", it is also advisable to take into account the opinion of musicians.

Rather than proposing a final technique, this paper has shown the multiplicity of research routes and alternatives in music synthesis utilizing wavelets techniques.

References

Daubechies, I. *Ten Lectures on Wavelets.* CBMS-NSF Regional Conference Series on Applied Mathematics. Philadelphia, SIAM, 1992.

Strang, G.; Nguyen, T. *Wavelets and Filter Banks.* Workshop Notes: San Jose State University, jan (13-16) 1995. Wellesley-Cambridge Press, 1995. /to be published/

Chui, C.K. *An Introduction to Wavelets.* San Diego, Academic Press, 1992.

Meyer, Y. *Wavelets: Algorithms and Applications.* Philadelphia, SIAM, 1993.

Risset, J.C. *The computer, music and sound models.* In: Combes, J.M. et al. Wavelets: time-frequency methods and phase space: proceedings of the International Conference, Marseille, dec.1987. 2ed., Springer-Verlag, p.102-123, 1989.

Jaffe, D.A. *An overview of criteria for evaluating synthesis and processing techniques.* In: XIV Congresso da Sociedade Brasileira de Computação: I Simpósio de Computação e Música: Anais, Caxambu, aug. 1994, p.53-61.

Oppenheim, A.V. *Discrete-time signal processing.* New Jersey, Prentice-Hall, 1989.

Acknowledgments

This paper has been produced in connection to MSc program research works, under support of CAPES.

Chaosynth

Um sistema que utiliza um autômato celular para sintetizar partículas sônicas

Eduardo Reck Miranda
Compositor, Mestre em Tecnologia Musical, Doutor em Música & Inteligência Artificial
Rua da República, 541 Apt. 317, Porto Alegre - RS, Brasil
E-mail: miranda@music.ed.ac.uk

Resumo em inglês: In this paper I introduce *Chaosynth*, a new sound synthesis system which uses a cellular automaton to produce sounds. *Chaosynth* functions by generating a large amount of short sonic events, or particles, in order to form larger, complex sound events. The synthesis technique of *Chaosynth* is inspired by granular synthesis. Most granular synthesis techniques uses stochastic methods to control the formation of the sound events. *Chaosynth*, however, uses a cellular automaton. I begin the paper by introducing the basics of granular synthesis and explain the functioning of the *Chaosynth* technique. I then introduce the basics of cellular automata and present ChaOs: the cellular automata used in *Chaosynth*. I also explain how ChaOs controls the synthesis parameters and how I used parallel computing to accelerate its performance. I conclude the paper with some final remarks and suggestions for further developments. An early version in English of this paper can be found in *Leonardo* Vol. 28, No. 4 (Journal of the International Society for the Arts, Science and Technology, MIT Press). A project report is available in the World Wide Web site of Edinburgh University: http://www.music.ed.ac.uk/pgrecs/eduardo/chaosynth_report/epcc_project.html. **Palavras chaves**: síntese sonora, autômatos celulares e música, modelagem simbólica de circuitos neuronais, computação paralela.

A Síntese Sonora Granular e o sistema *Chaosynth*

A Síntese Sonora Granular (SSG) é uma técnica para síntese de eventos sonoros complexos. O funcionamento da técnica SSG tem como princípio a produção de milhares de minúsculos eventos sonoros simples, ou *partículas sônicas* (por exemplo, partículas de 30 milisegundos cada), que ao todo formam eventos sonoros complexos.

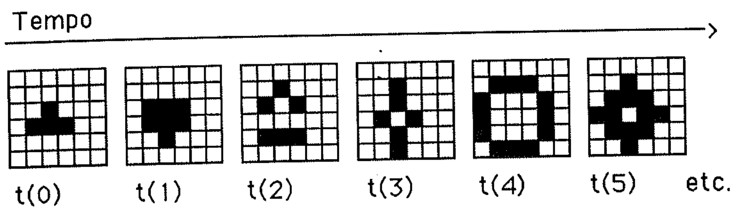
Esta técnica de síntese tem como base a *teoria granular de representação sônica* proposta na década de 40 pelo físico Dennis Garbor (1947). A teoria propõe que os sons de morfologia complexa são compostos por sequências de partículas sônicas menores e mais simples (Figura 6). A teoria granular de representação sônica teve muita repercussão no meio científico; por exemplo, Nobert Wiener, uma das maiores autoridades da teoria da informação, inspirou-se nas idéias de Dennis Garbor para medir o grau de informação de uma mensagem sonora (Wiener, 1964). O compositor Iannis Xenakis foi o primeiro a utilizar, na década de 60, uma teoria de representação granular para fins musicais (Xenakis, 1971). Entretanto, foi somente na década de 80, com a popularização dos computadores de alto desempenho, que as teorias de Dennis Gabor e Iannis Xenakis tiveram a oportunidade de serem postas em prática por compositores de um modo geral. Desde então, várias variantes da técnica SSG têm sido propostas e utilizadas; veja por exemplo (Truax, 1988; Roads, 1991).

O ponto crucial para o bom desempenho de um sistema de SSG é o método utilizado para controlar a produção das partículas sônicas; exemplos: o controle da quatidade de partículas por segundo e o controle da duração de cada partícula. A grande maioria dos sistemas de SSG utilizam métodos estocásticos (isto é, probabilísticos) para esse fim. Em *Chaosynth* eu proponho um método diferente: o método proposto utiliza um autômato celular chamado ChaOs (Miranda, Nelson & Smaill, 1992; Westhead, 1993).

Introdução aos autômatos celulares e seus princípios "musicais"

Os autômatos celulares são modelos matemáticos de sistemas dinâmicos e não-lineares, onde espaço e tempo são expressos por valores discretos e finitos. Um autômato celular (AC) é geralmente representado por um arranjo matemático (de 2 ou 3 dimensões) de variáveis discretas chamadas *células*. Os valores destas células definem o *estado* do AC. Estes valores mudam constantemente, em sincronia com o pulso de um relógio imaginário. A mudança dos valores das células é controlada por uma *função de transição global* (FTG), que determina o valor de uma célula em função dos valores de suas células vizinhas. Para que possamos visualizar o comportamento de uma FTG, o AC necessita ser programado em um computador, de tal maneira que as células sejam dispostas no monitor de vídeo como uma grade de minúsculos retângulos coloridos (ou cubos, se for um AC tridimensinal); neste caso, cada cor representa um valor numérico (Wolfram, 1983; Wilson 1988; Dewdney, 1989) (Figura 1).

***Figura 1**: Uma função de transição global controla os valores das células de um autômato celular. Neste caso as células podem assumir apenas dois valores: 0 (branco) ou 1 (preto).*



O primeiro AC foi inventado em meados da década de 60 pelo matemático John von Neumann, com a intenção de simular fenômenos biológicos, como por exemplo a *auto-reprodução.* (Cood, 1968). Ele estava interessado em construir uma máquina (mesmo que abstrata) que pudesse auto-reproduzir-se. O AC que o John von Neumann inventou consistia de uma grade bidimensional de células, cada qual podendo assumir um número finito de valores; cada valor representava um dos componentes da máquina auto-reprodutora. Ele definiu uma FTG cujo funcionamento fazia com que a "máquina" (uma figura na grade, composta de células de vários valores) se extendesse, ocupando porções vazias da grade (isto é, regiões compostas de células de valor neutro), e pouco-a-pouco fosse produzindo cópias dela mesma (isto é, fazendo com que as células neutras assumissem valores iguais aos da "máquina").

Desde então, uma grande variedade de ACs e FTGs têm sido inventadas e adaptadas para os mais diversos tipos de modelagem, nas áreas da física, biologia e computação. Recentemente, os princípios organizacionais dos ACs têm atraído a atenção dos musicólogos. Alguns compositores e pesquisadores estão investigando a possibilidade de utilização dos ACs para controlar, tanto as estruturas formais de alto nível de suas composições (forma musical), como os constituentes de baixo nível do espectro dos sons (espectro sonoro) (Millen, 1990; Hamman, 1991; Orton, Hunt & Kirk, 1991; Miranda, 1993; 1994). Em *Chaosynth* eu utilizo um AC para controlar a síntese dos contituentes espectrais dos eventos sonoros que eu tenho utilizado em várias composições, incluindo a peça *Olivine Trees* (que se encontra no CD do II SBC&M).

O autômato celular ChaOs

Um breve comentário sobre a metáfora do ChaOs

ChaOs (uma espécie de acrônimo para a expressão "Chemical Oscillator", ou Oscilador Químico, em português) é um AC bidimensional que pode ser utilizado para modelar alguns fenômenos químicos e biológicos, como por exemplo um fenômeno neurofisiológico conhecido como *circuito neuronal reverberante* (Eccles, 1958; Carpenter, 1990). Neste caso, o ChaOs pode ser imaginado como um circuito composto de milhares de componentes eletrônicos idêndicos, chamados de *células nervosas* (como se fosse um tecido nervoso constituído de neurônios interconectados). Em um dado momento, as células nervosas podem assumir uma variedade de valores, representados por cores diferentes. O ChaOs começa com as células nervosas nos mais variados estados, livremente distribuídas na grade. Na medida em que ele avança no tempo, a sua FTG faz com que a distribuição dos valores das células formem ciclos oscilatórios, que no monitor do computador são representados por padrões cíclicos de células coloridas (veja a Figura 5).

Eu associo o comportamento do ChaOs com o ciclo evolutivo dos sons produzidos pela maioria dos instrumentos musicais acústicos: eles geralmente começam com uma distribuição aparentemente desorganizada dos seus parciais (ruído), mas o espectro organiza-se rapidamente em uma certa configuração de parciais que oscilam com regularidade durante a emissão do som. Por exemplo, num som produzido em um violino, o ataque do arco na corda primeiramente produz um ruído; somente quando a corda finalmente começa a vibrar com regularidade é que o som da nota é produzido.

O algoritmo da Função de Transformação Global (FTG)

As células nervosas do ChaOs interagem com as suas células vizinhas através do fluxo de corrente elétrica que circula entre elas. As células nervosas podem estar em um dos dos três estados: *polarizada*, *despolarizada* ou *queimada*. Para definir o estado de uma célula nervosa, a FTG utiliza dois valores de referência: um valor de limite mínimo (**Vmin**) e um valor de limite máximo (**Vmax**). Se a voltage interna da célula nervosa (**Vi**) for menor que o **Vmin**, então esta célula está *polarizada*. Se o valor da **Vi** é maior ou igual que o **Vmin**, então a célula está *despolarizada*. Cada célula nervosa possui um *divisor de potência* (ou potenciômetro) cuja função é manter a sua **Vi** menor que o **Vmin**. Entretanto, quando o divisor de potência falha, a célula se despolariza. As células nervosas também possuem um *capacitor elétrico* que regula o seu grau de despolarização; a tendência, no entanto, é de tornar-se cada vez mais despolarizada. Quando a **Vi** atinge o **Vmax**, a célula nervosa "detona" e torna-se *queimada*. Uma célula nervosa queimada é regenerada no próximo pulso do AC (isto é, a FTG automaticamente torna uma célula nervosa queimada (no pulso **t**) em uma célula polarizada (no pulso **t+1**)).

De um modo geral, o plano de ação da FTG é especificado através dos seguintes parâmetros:

a) número **n** de possíveis valores para as células nervosas
b) as resistências **r1** e **r2** do divisor de frequência das células nervosas
c) a capacitância **k** do grau de despolarização das células nervosas
d) a quantidade de pulsos **t** do relógio imaginário da FGT
e) o tamanho da grade (isto é, o número de células nervosas)

Os estados da células nervosas são definidos por um número entre **0** e **n-1**, sendo que **n** não deve ser menor do que **3** (isto porque o ChaOs trabalha com células nervosas que podem assumir 3 estados; logo devemos prever pelo menos um valor para cada estado). O valor **0** corresponde ao estado *polarizado*, que é geralmente representado pela cor *branca*. Por outro lado, o valor **n-1** corresponde ao estado *queimado*, que é geralmente representado pela cor *preta*. Todos os valores entre **0** e **n-1** correspondem ao estado *despolarizado*. Conseqüentemente, o estado despolarizado pode possuir (quero dizer, quase sempre possui) vários graus de despolarização; quanto maior o valor de **n**, maior é o grau de despolarização. Várias nuances de cores entre o branco e o preto podem ser utilizadas para representar o estado despolarizado (isso fica a critério do programador).

A FTG do ChaOs é definido por 3 regras, que são aplicadas a todas as células da grade ao mesmo tempo; mas somente uma regra é selecionada para cada célula, dependendo do seu estado (**n** = uma certa célula da grade e **t** = pulsos do relógio imaginário):

(a) <u>se polarizada</u>: a célula pode ou não tornar-se despolarizada, dependendo do número de células vizinhas (neste caso, as 8 células que circulam a célula) que estão polarizadas (**Pcels**), do número de células vizinhas queimadas (**Qcels**) e das resistências do divisor de potência (**r1** e **r2**);

SE $cel(n)_t = 0$
ENTÃO $cel(n)_{t+1} = int(Pcels(n)/r1)_t + int(Qcels(n)/r2)_t$

(b) <u>se despolarizada</u>: uma célula despolarizada tende a tornar-se mais despolarizada ainda, dependendo da capacitância **k** e do grau de polarização de sua vizinhança (isto é, das 8 células vizinhas). O grau de despolarização é calculado pela soma dos valores das células vizinhas (**SVal**), dividida pelo número de células vizinhas que ainda se encontram polarizadas (**Pcels**);

SE $0 < cel(n)_t < n\text{-}1$
ENTÃO $cel(n)_{t+1} = k + int(SVal(n)/Pcels(n))_t$

(c) <u>se quimada</u>: uma célula nervosa queimada é regenerada no próximo pulso do relógio, isto é, torna-se polarizada;

SE $cel(n)_t = n\text{ -}1$
ENTÃO $cel(n)_{t+1} = 0$

O controle da síntese de partículas sônicas

Embora os princípios funcionais e metafóricos do ChaOs intuitivamente sugerem que ele pode ser utilizado para controlar a produção de partículas sônicas, eu tenho que confessar que eu encontrei sérias dificuldades para elaborar um método de controle que produzisse eventos sonoros interessantes;

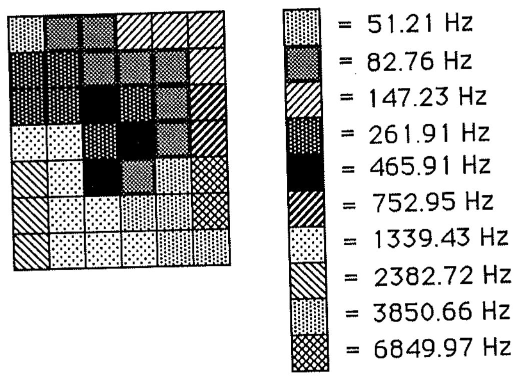
relacionar os valores numéricos produzidos pelo ChaOs com os parâmetros de um instrumento de "síntese granular" foi uma tarefa que demandou muita experimentação. Eu apresento a seguir o método que me pareceu mais razoável e que produziu, em minha opinião, os sons mais interessantes.

O método de contrôle

O espectro de cada patícula sônica produzida pelo *Chaosynth* é composto de vários parciais, sendo que cada parcial é uma onda senóide produzida por um oscilador (Figura 4). Para produzir uma onda senóide, um oscilador necessita de três parâmetros; frequência (em Hz), amplitude (em dB) e duração (em milisegundos) da onda. O ChaOs se encarrega de controlar os valores das frequências dos parciais (isto é, o conteúdo espectral) e a duração de cada partícula, enquanto que os valores das amplitudes dos parciais (isto é, o envelope espectral) são defenidos *a priori* pelo musicista. O systema *Chaosynth* relaciona os diferentes estados do ChaOs com os valores de frequências pré-estabelecidos pelo musicista (Figura 2).

A grade de células nervosas é dividida em várias sub-grades de mesmo tamanho, sendo que cada sub-grade é responsável pelo funcionamento de um oscilador. A Figura 3 ilustra uma grade de 693 células nervosas, distribuídas entre 9 sub-grades. A cada pulso **t** do relógio imaginário do ChaOs, o systema *Chaosynth* produz uma partícula sônica, cujas frequências dos seus parciais são calculadas pela média aritmética das frequências relacionadas com os valores das células nervosas das sub-grades correspondentes aos osciladores.

Figura 2: *O sistema relaciona os diversos estados das células nervosas com valores de frequência especificados pelo musicista.*



Além das amplitudes dos parciais, o musicista também estabelece *a priori*: (a) o tamanho da grade, (b) a quantidade de sub-grades (ou melhor, a quantidade de osciladores), (c) a distribuição das células nervosas entre os osciladores, (d) o número de possíveis valores para as células nervosas e as frequências relacionadas com estes valores, (e) as resistências **r1** e **r2** do divisor de potência do ChaOs, (f) a capacitância **k** do capacitor que regula o grau de despolarização e (g) o número de pulsos **t** do relógio da FTG.

Em outras palavras, cada partícula sônica é o resultado de uma síntese aditiva (Dodge & Jerse, 1985) de ondas senóides (Figura 4); na medida em que o ChaOs se desenvolve, os valores de frequência associados aos valores das células nervosas das sub-grades são utilizados para computar os valores das frequências dos osciladores que produzem cada senóide da partícula. A duração de um evento sonoro é determinada pelo número de pulsos do relógio do ChaOs; por exemplo, 100 pulsos produzindo partículas de 30 milisegundos cada, resulta em um evento sonoro de 3 segundos.

Este método de controle é interessante porque ele explora o comportamento do ChaOs para produzir eventos sonoros; esse comportamento pode ser associado ao funcionamento de alguns instrumentos musicais acústicos, como por exemplo o violino e a voz humana. Os sons destes instrumentos começam com uma distribuição livre e (aparentemente) desorganizada de seus parciais. Entretanto, os parciais organizam-se rapidamente em uma configuração espectral "pseudo-regular". O ChaOs começa com as células nervosas, dos mais diversos valores, livremente distribuídas na grade, mas na medida em que ele se desenvolve, as células tendem a organizar-se em formações ondulantes e

oscilatórias (Figura 5); a velocidade do processo de organização é controlada pelas resitências **r1** e **r2** e pelo capacitor **k**.

Figura 3: *Uma grade de 693 células nervosas, distribuídas entre 9 sub-grades, sendo que cada sub-grade se relaciona com um oscilador.*

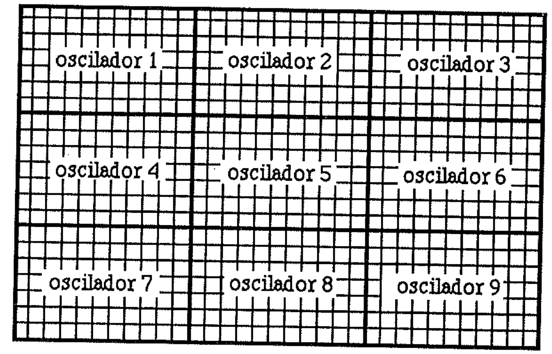
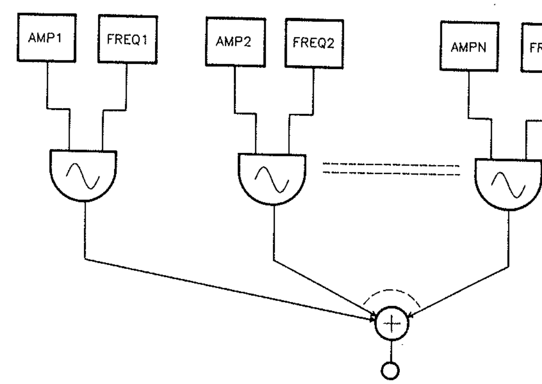


Figura 4: *Cada partícula sônica é o resultado de uma síntese aditiva de ondas senóides.*



Os sons do *Chaosynth*

Eu tenho sintetizado sons utilizando até 40 valores diferentes para as células nervosas (isto é, 40 valores de frequência diferentes) e uma grade de 25.000.000 (5.000 x 5.000) células nervosas, distribuídas entre 25 osciladores (isto é, cada partícula pode possuir até 25 parciais) (Figura 6).

Os sons resultantes são na sua maioria muito parecidos com os sons de água em movimento; eu costumo apelidar os sons do *Chaosynth* de "sons líquidos e borbulhantes". O *Chaosynth* produz sons "líquidos" e "borbulhantes" nos mais variados graus de densidade, de acordo com a duração individual de cada partícula (por exemplo, 100 a 600 milisegundos cada). Variações na *coloração* do timbre são controlados pela variação dos valores das frequências associadas aos valores das células nervosas, enquanto que o grau de transição entre *ruído* e *padrões oscilatórios* é controlado pelos valores do capacitor **k** e dos resitores **r1** e **r2** do ChaOs.

Maiores informações sobre a relação entre os controles do sistema e os sons que ele sintetiza podem ser obtidas no relatório que está disponível na World Wide Web (veja o endereço no final do resumo em inglês, na primeira página deste artigo).

Figura 5: Na medida em que o ChaOs se desenvolve, as células tendem a se organizar em formações ondulantes e oscilatórias.

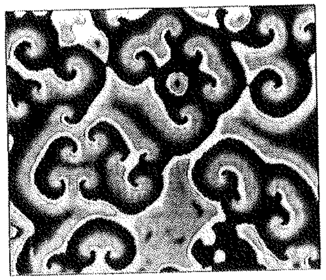
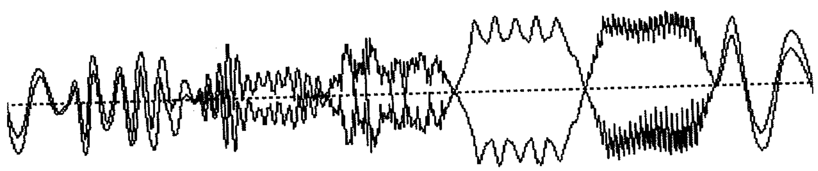


Figura 6: A representação de uma sequência de partículas sônicas produzida por Chaosynth.



A arquitetura do sistema e o uso de computação paralela

A arquitetura do *Chaosynth* está ilustrada na Figura 7. Os sons são sintetizados utilizando o compilador Csound (Vercoe, 1991). O Csound é uma linguagem para síntese sonora no qual o musicista especifica um programa de síntese (isto é, um instrumento digital) em um *arquivo-orquestra* e uma lista numérica de parâmetros de síntese (para o *arquivo-orquestra*) em um *arquivo-partitura*. Quando o compilador do Csound é ativado, estes dois arquivos são processados e o "som" é gerado e arquivado em um *arquivo-sonoro*.

A *interface* gráfica do *Chaosynth* fornece vários "knobs" e "sliders" para a especificação dos diversos parâmetros do sistema (por exemplo, as frequências associadas aos valores das células nervosas e as amplitudes dos osciladores). Uma vez especificados os parâmetros, o musicista então ativa o ChaOs, que por sua vez produz um *arquivo-partitura* (compatível com o *arquivo-orquestra* do sistema). Finalmente o compilador Csound é ativado e o som então é sintetizado.

A versão atual do sistema (1.0) utiliza computação paralela para tornar a FTG do ChaOs mais rápida. Para paralelizar a FTG eu utilizei a ferramenta PUL-RD, desenvolvida no Centro de Computação Paralela de Edimburgo (EPCC) (PUL-RD significa Parallel Utilities Library - Regular Domain). A ferramenta PUL-RD utiliza uma técnica chamada *decomposição de domínios regulares* (ou "regular domain decomposition", em inglês) para paralelizar programas baseados em autômatos celulares cuja grade pode ser decomposta regularmente (Brown, 1994).

A técnica de decomposição de domínios regulares foi desenvolvida para computar problemas cujos dados podem ser organizados em grades de considerável tamanho. Ela divide a grade em sub-grades regulares e as distribui entre os diversos processadores do computador paralelo para o processamento concorrente. Eu utilizo a PUL-RD para dividir a grade de células nervosas em sub-grades, de tal forma que cada sub-grade seja relacionada com um oscilador (Figura 8). Neste caso, todos os parciais de cada partícula sônica são computados em paralelo.

Figura 7: A arquitetura do sistema.

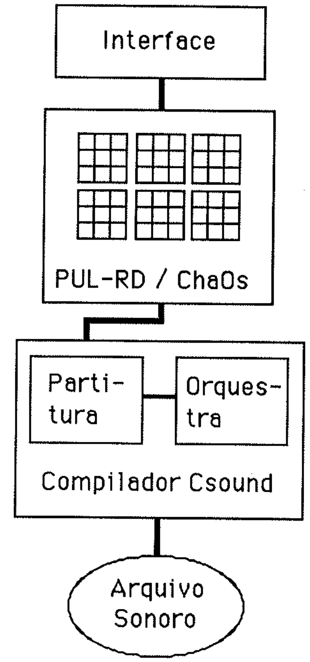
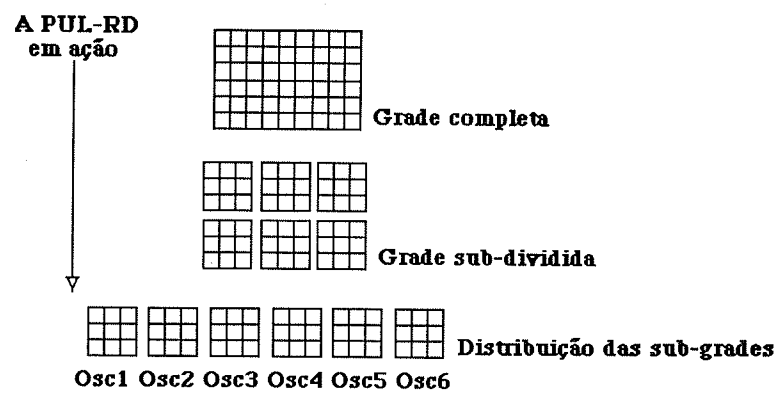


Figura 8: A ferramenta PUL-RD divide a grade em sub-grades regulares e as distribui entre os diversos processadores do computador paralelo.



Conclusão

O *Chaosynth* é um sintetizador cujos parâmetros de síntese são controlados por um autômato celular (AC) chamado ChaOs. A técnica de síntese do *Chaosynth* se assemelha muito com a técnica de síntese granular, pois ela produz uma grande quantidade de partículas sônicas simples, que no total formam eventos sonoros mais complexos. Pelo menos duas características diferem o *Chaosynth* dos sistemas tradicionais de síntese granular: (a) o fato de que eu utilizo o ChaOs para controlar a produção de partículas sônicas e (b) cada patícula é o produto de uma síntese aditiva de várias ondas senóides, enquanto que nos sistemas tradicionais cada partícula é composta por apenas uma senóide.

A versão atual do sistema utiliza apenas um método para mapear o comportamento do ChaOs com os parâmetros de síntese. Em futuras versões eu pretendo deixar esse mapeamento em aberto, de tal forma que o musicista possa inventar e experimentar outros métodos.

No momento eu estou investigando a possibilidade de paralelisar o módulo de síntese sonora, com o objetivo de minimizar ao máximo o tempo necessário para computar as amostragens digitais (ou *samples*) para produção do som. Assim o *Chaosynth* também poderá funcionar como uma espécie de instrumento musical, tocado por intermédio de controladores apropriados; como por exemplo a *luva interativa* que está sendo desenvolvida na Unicamp, pela equipe do Jônatas Manzolli, cujos resultados de pesquisa também estão sendo apresentados no II SBC&M. A longo prazo, eu pretendo unir o *Chaosynth* ao sistema CAMUS (também de minha autoria). O sistema CAMUS utiliza os autômatos celulares para controlar níveis mais altos de composição; por exemplo, o nível formal da composição de fragmentos musicais (Miranda, 1993; 1994).

O *Chaosynth* foi implementado em um supercomputador CM-200, no Centro de Computação Paralela de Edimburgo (EPCC), na Escócia, com o auxílio técnico de Martin Westhead and Robert Fletcher.

Referências Bibliográficas

Carpenter, R.H.S. **(1990)**, *Neurophysiology*, Edward Arnold, GB.
Cood, E.F. (1968), *Cellular Automata*, Academic Press, GB.
Brown, S. (1994), *Parallelising Grid-Based Applications with PUL-RD*, relatório técnico, Edinburgh Parallel Computing Centre, GB.
Dewdney, A.K. **(1989)**, "A cellular universe of debris, droplets, defects and demons", *Scientific American*, August:88-91.
Dodge, C. and **Jerse**, T. **(1985)**, *Computer Music - Synthesis, Composition and Performance*, Schirmer Books, UK.
Eccles, J.C. **(1958)**, *The Physiology of Imagination*, WH Freeman & Co (reimpressão de artigo publicado na revista Scientific American), EUA.
Garbor, D. **(1947)**, "Acoustical Quanta and the Theory of Hearing", *Nature*, 159(1044):591-594.
Hamman, M. **(1991)**, "Mapping Complex Systems using Granular Synthesis", *Montreal ICMC 1991 Proceedings*, 475-478.
Millen, D. **(1990)**, "Cellular Automata Music", *Glasgow ICMC Proceedings*, 314-316.
Miranda, E.R., **Nelson**, P. and **Smaill**, A. **(1992)**, "ChaOs: a Model for Granular Synthesis using Cellular Automata", *Annual Report & Project Directory 1991-92*, Edinburgh Parallel Computing Centre, 153-156, GB..
Miranda, E.R. **(1993)**, "Cellular Automata Music: An Interdisciplinary Project", *Interface*, 22(1):3-21.
Miranda, E.R. **(1994)**, "Cellular Automata Music Composition", *Languages of Design*, 1994(2):105-107.
Orton, R., **Hunt**, A., and **Kirk**, R. **(1991)**, "Graphical Control of Granular Synthesis using a Cellular Automata and the Freehand Program", *Montreal ICMC 1991 Proceedings*, 416-418.
Roads, C. **(1991)**, "Asynchronous Granular Synthesis", *Representation of Music Signals*, De Poly, G. et al (editores), MIT Press.
Truax, B. **(1988)**, "Real Time Granular Synthesis with a DSP Computer", *Computer Music Journal*, 12(2):14-26.
Vercoe, B. **(1991)**, *Csound Manual*, disponível via Internet ftp no diretório *music* da máquina media-lab.mit.edu.
Westhead, M.D. *Chaosynth: a System for Granular Synthesis using a Cellular Automaton*, EPCC Summer Course Project Report, Edinburgh University, GB
Wiener, N. **(1964)**, "Spatial-Temporal Continuity, Quantum Theory and Music", *The Concepts of Space and Time*, Capek, M. (editor), Reidel.
Wilson, G. **(1988)**, "The life and times of cellular automata", *News Scientist*, October(8):44-47.
Wolfran, S. **(1983)**, "Statistical mechanics of cellular automata", *Reviews of Modern Physics*, 55(3):601-623.

One Tool, Two Programs and Several Ideas for Composition with Spectral Modelling Synthesis

CELSO AGUIAR

Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305-8180 USA
aguiar@ccrma.stanford.edu

Abstract

Sound can be approached from complementary perspectives. This paper describes aspects of one of my works, **'Piece of Mind'** for tape, where time domain techniques derived from *'musique concréte'* are blended to powerful spectral modelling (**SMS**) in the processing of natural sounds. Programs written in collaboration with Xavier Serra have made SMS available in **CLM**'s synthesis environment. Deterministic components are synthesized by a highly efficient IFFT algorithm that relieves the delay of pleasure composers have historically accumulated with additive synthesis. Bell sounds and singing voice are hybridized or cross-faded to provoke ambiguity of identity. On the other hand, the way SMS handles its stochastic part exposes the fact that dsp guys and composers may not always speak the same language. Being aware of what synthesis techniques are geared to and the ability to present proper solutions are first requirements. A graphical interface to display SMS information on a single view is introduced.

SOME PREMISES

A prediction was once made by researcher Julius O. Smith III where he affirmed that time domain techniques for sound synthesis and transformation would migrate, or be absorbed in the future, into frequency domain ones (**Smith**, 1991). By time domain techniques he meant all synthesis techniques derived from *'musique concréte'* like sampling and granular synthesis. Under frequency domain he listed several techniques including spectral modelling synthesis (SMS) and inverse Fourrier synthesis.

His prediction was made about four years ago and considered the fact that sampling and granular techniques can add very interesting new colors to the sonic palette, but they are very difficulto control. To obtain control over these techniques, more general sound transformations would be required. As these transformations are to be understood in terms of what we hear, and the best way to understand what we hear is through the short-time spectrum it seems logical to assume that time domain techniques point towards spectral modelling.

The word 'control' and its meaning in his text may stir polemic if directly transposed to a compositional context. Composers, including myself, may want to deal with the uncontrollable, and that stands very far from avoiding control over materials. However, I still think his idea is interesting enough to deserve some discussion and consideration. In one side we have time domain techniques that are straight-forward to implement but which impose barriers to effectively manipulate sounds. On the other hand we have very powerful analysis-synthesis techniques which allow us to modify sound in very effective ways but whose implementation isnot as immediate as in the previous case and which involve much more computation than the mere reading oftables or files of recorded samples.

For the composer the amount of delayed pleasure associated with additive synthesis, for example, is such that few of us will risk to write a piece depending entirely upon it. It's a very well known facthat the computational cost of sum-of-sinusoides directly increases with the number of partials in a sound. This limitation in part explains what has been going on recently with composers who rely in some way ondigital synthesis to produce their musical works. Tired of the predictability of sounds obtained through abstract methods of synthesis they feel quite comfortable and relieved when turning on to time domain methods in the hope that the sonic aspect of their piece will retain the complexity of the original sound sources. These methods have derived mostly from the renewed computer practice of concrete procedures but the point is: who can blame them?

On the other hand physical modelling still seems to be quite in its infancy and hasn't yet been able to clearly fulfill the promise of providing controllable instruments with a small number of simple and intuitive parameters. Besides that fact, physical modelling seems also a lot geared towards performance, an aspect of computer music which shouldn't be discarded but whose full potential is only obtained through the use of controller devices interfacing a particular algorithm to human performers. Just like playing an oldnatural instrument again, Science seems to be trying to reproduce nature in a box.

<cl> (EQUALP FANCIER_TOOLS BETTER_MUSIC)
NIL

My basic drive when composing 'Piece of Mind' was to write a piece where a strong sense of movement and direction would be present. One who is not used to composing tape music on non real-time systems may still easily perceive how hard it is to keep up that urge.

Besides the technical difficulties and constraints that computer music in general normally imposes on the composer, tape music in particular is a medium which also inflicts its own hardships upon the act of composition. The necessity of maintaining attention focused on the large scale aspects of a composition is severely damaged by the repeated hearing of sounds and by the rate at which these sounds are being created by a certain algorithm. On the one hand they naturally have to be examined in detail to verify if our initial setting of parameters corresponds to what we actually hear, but on the other, this repeated hearing quickly wears out the impact this sound would have to suggest new associations on our imagination. The most precious gift in tape music, being able to previously hear while planning what a piece will actually sound like during the concert, can easily become our most dangerous weakness.

This process of give and take, if not economically carried out, will make it very easy to loose track of our initial intentions and will make us zoom into details which don't really matter for the overall picture and will easily be missed by the audience (**Smalley**, 1986). A minimally clear plan has to be previously set up. The nature of this plan may differ from composer to composer in the idiosyncrasies their creative process may have inherited or acquired in the use of the medium but it still has to be there if a minimum of success is desiredfrom the result.

From another perspective, complexity can exert a great deal of appeal over all of us. It's quite easy to associate fancier tools with better results in our minds. Composers in particular tend to think that the more advanced tools will automatically generate better music. I've been finding this piece of thought to be one the most insidious lies propagated nowadays in computer music composition. Despite the fact that we have to be open to the new, it's easy to find this overabundance of systems and tools to be hiding a deficiency in creativity if not a resistance to the true demands for the new in art, exceptions made.

The idea of approaching sound from two different and complementary angles should allow for a balance in the demands to keep the focus of attention and a minimum degree of complexity in the sonic result. Particularly for this piece I wanted to be able to either avoid or fulfill expectations by the simple interplay of textures juxtaposed in time. Movement of sound in space was not really as important as being able to transmit an impression of imminent instability, an impression that things are about to happen. Higher portionsof the available frequencies should be explored as well, and they are easy to forget if one is not conscious of the real difference they can make in the overall result of a piece.

The use of voice and natural sound sources would provide for a better commmunication with audience but above all I wanted to explore the consequences of making clearly audible transformations in or between them. What we may be avid to find is a musical discourse where it would be possible to incorporate complexity and yet not be complicated.

AN ALGORITHM

Xavier Serra was telling me of all these new spectral modelling researches that had been appearing aftehis famous SMS thesis. Of how spectral modelling (based on sound as it arrives our ears) will in the near future look much like the physical modelling approach (based on how the sound is actually produced at its very source) in trying to recreate sounds from natural instruments and in general. One of this improvements was an IFFT algorithm for additive synthesis developed by Xavier Rodet. A student of Serra in Spain had been working on it without positive results yet. I decided that I should set out to reconstruct it even if from very scarce bibliography (**Rodet**, 1992).

The main idea behind the IFFT algorithm is that, instead of performing the old time-consuming method of sum-of-sinusoides, we use the Fast Fourrier Transform for the conversion of domains involved in additive synthesis. In order to do that some assumptions have to be previously made. We start by constructing aframe of our spectra in the frequency domain. For this frame of spectra we have to assume that our partials have been imposed some kind of windowing in order to reduce the spectral splatter along the bins of our FFT. The Blackman window seems as a natural candidate to achieve this. It concentrates most of the energy of apartial in a total of nine bins if we consider a FFT size of 256 points. In the Blackman window all other bins remain at least at a comfortable 200 Db below the peak amplitude of our partial (Fig 1).

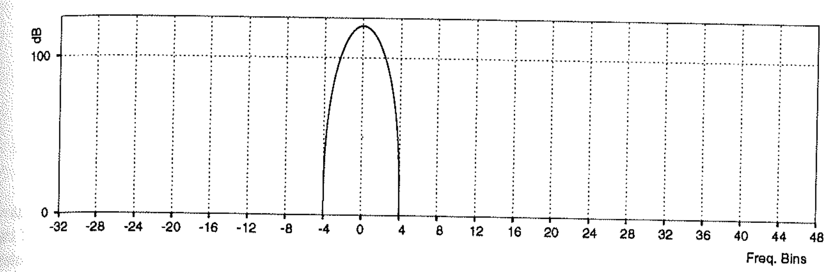


Fig. 1 - The 9 most significant bins in a frequency domain Blackman window (256 FFT size).

In our implementation of the program we decided that a resolution of 288 points for in-between bins frequencies would be enough. Higher resolutions can be easily implemented. The function displayed in Fig2 is used as a table for sampling the partials when constructing our frequency domain frame of spectra.

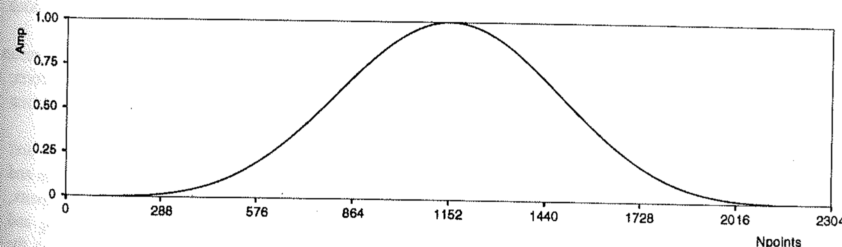


Fig. 2 - Frequency domain Blackman (undB'ed version of above) and resolution in our IFFT CLM program.

The resulting frequency of a partial obtained after we perform the inverse FFT depends on the position we start sampling the 9 bins within the table. It's also easy to notice that in reality we have only 8 binsand not 9 as one of them will always falls off the limits of our table look-up process. For smaller partial amplitudes this number can be further reduced due to the loss of significance in the least central bins.

When deciding the size of the FFT we have to consider the fact that it has to be small enough to avoid perceptual problems. A 256 points FFT at 44.1 kHz seems like a good choice. It is long enough to permit a fine frequency resolution and is short enough to avoid our perception of the overlap-add frame rate. Wealso have to keep track of the phase information for each of our partials in the spectra when performing the overlap-add portion of the algorithm since they're tied to the frequency information.

We then perform the inverse Fourrier transform. Once in the time domain, we now can divide our resulting waveform by a time domain Blackman window in order to compensate for our initial assumption that our spectra was windowed. To perform a 50% overlap-add in the short-time spectra we also multiply our waveform by a triangular window precisely splicing the waveforms together. The two steps described above canbe collapsed into an one multiplication operation. The necessary window is displayed in Fig. 3.

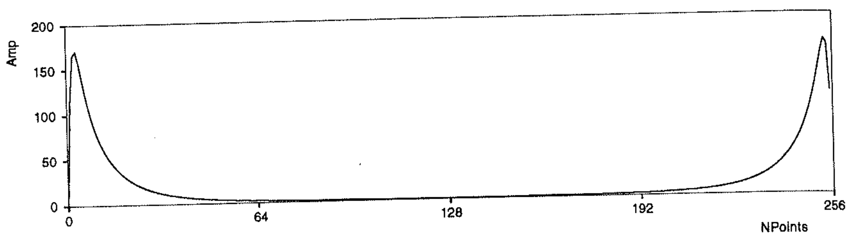


Fig. 3 - Time domàin Triangular/Blackman windowing.

One problem that we had to face when writing the algorithm regards to what happens when we have partials sounding below 344.53 Hz, or the 4th bin in our 256 point FFT. The lower bins will soon fall off the limit of our DC bin component and the balance in our Blackman windowed spectra will be lost. At this pointwe just assumed that the negative ones would reflect into the positive domain with a change of sign, or suffer an inversion of phase, just like in the case of FM synthesis (**Chowning**, 1973). The same can be demonstrated for the Nyquist frequency case.

Another interesting point was that Xavier Serra believed that we would be able to add the SMS stochastic data into the spectra of partials (deterministic data) before performing the inverse FFT in order to resynthesize them altogether. We found this not to be possible. As we are assuming a Blackman windowed spectra in the input to the IFFT it's easy to notice what will happen when we divide by a Blackman window once in the time domain. SMS' stochastic data are made out of piece-wise linear segments that approximate noise bands of the spectra in the frequency domain, lending themselves difficult to window compared to what could easilybe done in the case of partials.

The operation of the CLM instrument built from this algorithm to resynthesize deterministic components requires the memory storage of the following look-up tables: **wave**, for our time domain output waveform; **window**, the function in Fig. 3; **blackman**, the function in Fig. 2; **sine**, **cosine**, for phase interpolation calculations.

The parameters which can be controlled in this basic deterministic resynthesis instrumen(IFFTdet) are: **beg**, **dur**, **amp** and **freq**, common parameters for time, duration, amplitude and pitch of a note; **file-amp** and **file-frq**, two SMS format files containing the information from the partials of analyzed sounds; **ampenv**, **freqenv** and **sclenv**, envelopes for amplitude, frequency and scaling of the spectra of a sound; **recFrq** and **recAmp**, envelopes for record positioning inside an SMS file; **spectEnv**, for imposing an envelope on the magnitudes of a spectra; **partialAmps** and **partialFrqs**, lists with partial numbers for recombining their structure; **coord** and **rev-amount**, for reverberation and spatial localization.

Our implementation of the IFFT algorithm has achieved a very good level of performance when resynthesizing SMS deterministic data. In CLM, by nature a non real-time environment, our specs for a22 kHz, 30 partials sound using a complex IFFT have been 3:1 on a NeXT slab and .7:1 on a Pentium 60 MHz.

EXPLOSION OF HYBRIDIZED SOUNDS

Hybridization and cross-fade of sounds are only two of the new possibilities opened by approaching sounds in their inner spectral content (**Serra**, 1994). They provide us with objective ways to play with the identity of natural sounds. The power they have in suggesting new ambiguous associations is at the root of electroacoustic music thought.

Hybridization and cross-fade of sounds have already been tried with other synthesis methods and systems. I believe our implementation for sound hybridization and cross-fade in CLM to be able to render very clear and precise results, especially regarding intelligibility of articulated voice sounds (**Harvey**, 1986).

Fig. 4 below displays the spectrogram of a cross-fade from a complex type of bell sound to a segment of singing voice. It's possible to notice how the many partials in the complex bell sound are actually merging together and transforming into singing voice with its more transparent nature.

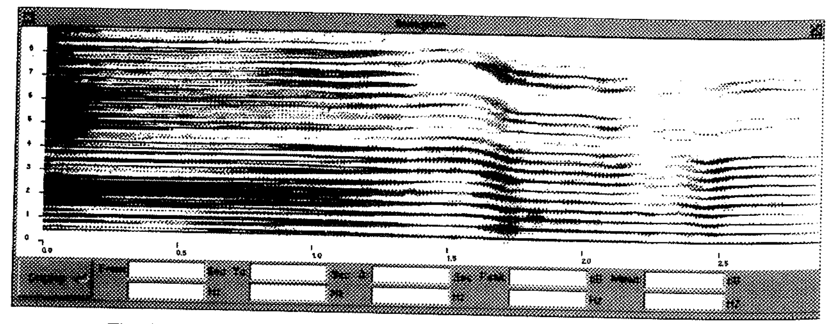


Fig. 4 - Cross-faded sound (bell -> voice) showing interpolation of partials.

The following example (Fig. 5) explores the opposite case where departing from a more simple and transparent bell sound we arrive at the same singing voice segment in the previous example. In this casethe partials have to bifurcate in order to merge the partials of the subsequent sound.

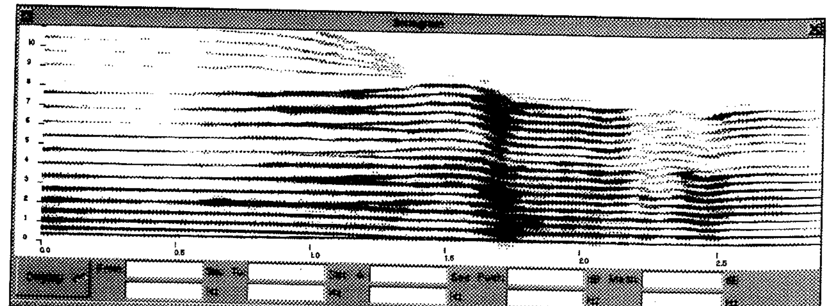


Fig. 5 - Cross-faded sound (another bell -> voice) showing contrary merging and interpolation of partials.

A program was written to perform a match in the partials of the two sound identities to be cross-faded or hybridized. These sound identities have to be previously analyzed in their short-time spectral content with the tools in SMS. Their resulting files are then used as input to that routine which by its turn performs the match or comparison in the proximity of the two groups of partials depending on the points between which we wanthem to be cross-faded. Output lists inform which partials in the departure sound should then be interpolated to what other partials in the target sound. The process above should be able to work for any natural sound whose spectral content can be translated into deterministic components.

Besides the common standard parameters for duration, amplitude and frequency, we have this additional set of parameters to control in our IFFTfade instrument: **file-amp1, file-frq1, file-amp2, file-frq2**, four SMS format files containing frequency and magnitude of the sounds to be cross-faded; **ampenv, freqenv1** and **freqenv2**, envelopes for amplitude and fundamental frequencies in each sound; **recFrq1** and **recFrq2**, envelopes for frequency record positioning inside an SMS file; **recAmp1** and **recAmp2**, envelopes for amplitude record positioning inside an SMS file; **partials1 amps1, partials2, amps2**, output lists from the matching program to control partial interpolation; **spectEnv1, spectEnv2**, for imposing envelopes on the magnitudes of each spectra; **scl1** and **scl2** scaling parameters to be imposed on the spectra of the sounds; **iFrqenv, iFrq, iFrbase, iAmpenv, iAmp** and **iAmbase**, parameters to control the sections to be cross-faded in the sounds; **coord** and **rev-amount**, for reverberation and spatial localization.

Improvements can be made and new possibilities can be open within SMS yet. Some of them point in the direction of being able to analyze and transform larger portions of sound files. There's a clear necessity for more intelligence from the part of the analysis programs provided by SMS. The time consumed in the choice of the right parameters to perform the analysis is still big compared with the time spent in the analysis itself. This things can and should be made automatic.

The idea of sounds decomposed in deterministic and stochastic parts is very interesting but the rendering of the stochastic part seems still unsatisfactory, at least to me. In the process of composing the piece I realized that Xavier had meant it to be geared towards data compression. A new residual sound file in the size of the original sound is generated during the decomposition process of SMS. I could achieve more precise results just by phase-vocoding this residual sounds and then mixing them to the deterministic part instead of applying the stochastic model.

One idea that can be implemented is to apply graphically based processes to SMS files considering its data to be a huge matrix that can be transformed in different ways. This idea has been suggested while building a new interface for displaying SMS data with the *RenderMan* package of graphical rendering routines. The nice thing about this interface is that all SMS data can be displayed on a single view. The picture can then be moved around via mouse and be viewed from different angles during the process of analyzing and transforming sound. This same picture could then be viewed as a surface object that could suffer whatever modifications graphic surfaces can undergo.

REFERENCES

Chowning, J. (1973). "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation." *J. Audio Eng. Soc.*, vol. 21, no. 7, pp. 526-34.

Harvey, J. (1986). "The Mirror of Ambiguity." In *The Language of Electroacoustic Music*, edited by Simon Emmerson. Macmillan: London, pp. 175-190.

Smalley, D. (1986). "Spectro-Morphology and Structuring Processes." In *The Language of Electroacoustic Music*, edited by Simon Emmerson. Macmillan: London, pp. 61-93.

Smith, J. O. (1991). "ViewPoints on the History of Digital Synthesis." Keynote Paper, *Proceedings of the International Computer Music Conference*.

Rodet, X. and P. Depalle. (1992). "Spectral Envelopes and Inverse FFT Synthesis." 93rd Convention of the Audio Engineering Society, San Francisco.

Serra, X. (1994). "Sound Hybridization Based on a Deterministic plus Stochastic Decomposition Model." *Proceedings of the International Computer Music Conference*.

ACKNOWLEDGMENTS

My thanks go first of all to composer David Soley from Stanford University who allowed me to have his ears as my third and fourth ears, in my opinion a very healthy practice when composing music for tape. I also thank Xavier Serra for his detached collaboration and teaching during CCRMA's DSP Summer Workshop of 94. The paper was written while under a DMA fellowship from brazilian research support agency CAPES.

O zig-zag conceitual no estúdio de composição.

RODOLFO CAESAR
email: caesar@vms1.nce.ufrj.br
Escola de Música, Centro de Letras e Artes, Universidade Federal do Rio de Janeiro.
Rua do Passeio 98, Centro, Rio de Janeiro. R.J. Brasil.

Resumo: O texto discorre sobre a situação do compositor em estúdio eletroacústico e o que resulta da peculiaridade desta condição, comparada com a da escrita notacional. Retomando antigas discussões (musique concrète vs. o mundo) aborda questões pertinentes à tecnologia para fins musicais, tocando em conceitos e preconceitos de inspiração talvez estética, eventualmente mercadológica, e certamente ideológica. Palavras-chave: matéria, material, forma, objeto sonoro, 'écriture', intradutibilidade, resposta não-linear, MusicV, Csound e outras ferramentas. Extraída, traduzida e re-escrita de minha tese de doutoramento 'The Composition of Electroacoustic Music', (UEA - Inglaterra, 1992), esta comunicação foi lida no painel "Oficina de Música e Composição", realizado no Centro de Pós-Graduação, Pesquisa e Extensão', Conservatório Brasileiro de Música, em novembro de 94.

Introdução.

A exposição a seguir resultou de tentativas de reduzir a palavras - a princípio apenas para efeito de elaboração de uma tese de doutorado - um pouco da minha vivência em estúdio de composição eletroacústica, e o que eu bem pudesse objetivar de minhas estratégias de trabalho. Trazendo à tona, de passagem, o pormenor específico da música concreta que - no meu entender - readquire importância, de certa maneira este texto poderia confirmar aquela impressão generalizada de que os compositores só sabem falar de si próprios, e que sendo assim a teoria deveria ser deixada aos teóricos. O trabalho de composição em estúdio não é um assunto do interesse direto de todos. Entretanto a exploração desta prática, ou melhor, sua redução a palavras, pode tocar em territórios condizentes com esta conferência, especialmente no que diz respeito ao cruzamento da experiência da composição com a verbalização, esta porta principal da teoria musical. Sendo este depoimento um produto híbrido - são reflexões originadas em uma prática e pesquisadas com apoio bibliográfico - acredito que chegará a um campo de interesse comum, sobretudo quando o percurso resvalar em dúvidas acima de qualquer manifesto de cunho pessoal, deixando-se impregnar por inquietações de ordem mais geral. Nestas incluem-se questões acesas como a relação da tecnologia com a atividade humana, especialmente a música.

Como preliminar para iniciar o tema, trago - antecipadamente - em meu auxílio a afirmação de um colega:

> 'Para a música de sons fixados [ou eletroacústica (*as chaves são minhas*)], o material sonoro coloca uma questão mais crucial, porque a realização acústica de uma peça deste gênero está inteiramente entregue às mãos do compositor: os dois níveis - de material e de organização - estão implicados em ações comuns.' (Michel Chion, 1991), p.29.

De fato, quando quer que eu tente apresentar verbalmente minhas peças, dou-me conta de que expressões tais como 'som', 'unidade', 'objeto', e principalmente 'material' e 'estrutura' só podem ser aceitas enquanto tentativas rudimentares, como invólucros emprestados de diversas áreas de nossa experiência, cuja inadequação esbarra no estofo sutil da percepção musical. Devido à natureza do próprio 'material' eletroacústico em questão, até os fundamentos do vocabulário musical revelam sua inadequação diante da tarefa de discorrer sôbre a ME. Sentidos flutuam de uma palavra a outra e entram em colisão pela posse da mesma palavra, estabelecendo assim uma relação crítica entre verbalização e percepção. A interminável necessidade de empregar metáforas frouxas para a descrição do 'musical' só é sobrepujada pelo derramamento mútuo entre elas. Conceitos básicos como 'material', 'matéria' e 'estrutura' - com que sempre nos defrontamos ao tentarmos descrever nosso trabalho composicional - não oferecem firmeza para quem trabalha com a música eletroacústica, apenas podem ser usados quando em uma constante relativização e fluidez (cf. o esforço no capítulo 2, CANTO, de minha tese).

Debruçando-nos sobre estes termos encontraremos diferenças transparentes entre a assim chamada música eletroacústica e a assim chamada música 'instrumental', e até tentaremos compreender uma confusão que foi instaurada entre 'invenção' e 'descoberta'. Concluiremos tmbém que a estampa de nossa relação com a tecnologia tem sido marcada pela inadequacão.

Material e Estrutural.

Conceitos flutuantes permeiam quase todas as teorias da música eletroacústica, algumas das quais conscientes desta condição. É importante lembrar aqui o trabalho de Pierre Schaeffer não só por sua apresentação de um novo universo sonoro e musical, mas também pelo vasto âmbito das indagações que vieram a provocar inquietações sempre experimentadas por muitos compositores, entre os quais me incluo.

Em seus 'Traité des Objets Musicaux' (1966) e 'Solfège de l'Objet Sonore' (1967), Schaeffer lida com a percepção sonoro/musical, apresentando um método de dividir a experiência musical em diversos quadros e camadas de conceitos bipolares. Queixam-se compositores, hoje como há trinta anos, que aqueles só podem ser assimilados quando equilibrados sobre algum gráu de impermanência. Um 'objeto musical' é na verdade sempre sonoro, e um 'objeto sonoro' pode ser por vezes musical, porém um e outro não devem ser confundidos. São modos de se descrever um evento acústico: no primeiro caso quando se encontra desligado de contexto musical. Os conceitos schaefferianos até hoje são ferramentas analíticas poderosas se, e apenas se, ao invés de isolados em opostos rígidos, deixarem-se ler como vetores apontando para pólos. Não pode haver algo como um 'objeto musical' puro, mas sim algo que se dirige nesta direção, vindo do pólo 'objeto sonoro', e vice-versa.

Indo mais fundo, mergulhando perceptualmente 'objeto sonoro' adentro, nos deparamos com o critério de percepção chamado 'allure', uma qualidade mais ou menos correspondente ao vibrato ou ao tremolo, que pertence ao que Schaeffer chama de 'critérios de matéria', em oposição aos de forma. Aqui perguntamos unânimemente: quando é que - por desaceleração - esta figura deixa de ser uma 'allure' e penetra no domínio dos 'critérios de forma', tornando-se, por exemplo, um perfil melódico ou um perfil dinâmico? Protegido por uma aproximação deliberadamente fenomenológica, Schaeffer jamais proporá traçar uma fronteira entre matéria e forma, isolando-se deste modo de todos os esforços por uma objetividade explícita e científica. Vejamos como esta linha embaçada se apaga de vez no trabalho em estúdio.

Diversas estratégias composicionais são comuns tanto às músicas eletroacústicas quanto às instrumentais. Compositores de ambas 'espécies' geralmente identificam dois estágios de operação que, à medida em que progride o trabalho de composição, eventualmente confundem-se um no outro. O primeiro corresponde à produção e seleção dos elementos primários: os 'objetos', ou 'unidades menores' (sons, motivos, células, etc.), os quais, no segundo estágio, adquirem uma existência de acordo com sua posição em uma 'estrutura' dada. Muitas vezes, já neste estágio secundário, os compositores sentem a urgência de retornarem ao primeiro: se um dos elementos primários não desempenha o que era esperado no nível estrutural, uma substituição se faz necessária, ou pelo menos uma correção.

Fazer uns sons.

O estúdio eletroacústico propicia uma diferença de grau significativa, requisitando dos compositores uma atitude muito mais flexível diante da polarização. Os recursos sonoros (mais do que isso: soantes) gravados em suporte analógico ou digital, ou produzidos em tempo real, por sua natureza 'animada' podem mesclar os dois estágios indissolùvelmente.

A frase, banalmente dita, 'fazer uns sons' - para denominar os primeiros passos no trabalho de composição - é tão usada por compositores de ME quanto é vaga e imprecisa. Porém não é nada banal. 'Fazer uns sons' engloba não apenas a produção de elementos primários (que posteriormente serão conjugados em idéias musicais), mas fala também, neste estágio inicial, da possibilidade de se permitir a descoberta ou a confecção de algo maior, pertencente a uma escala mais estrutural. Isto é, 'fazer uns sons' pode remeter tanto à prospecção de 'objetos', personagens ágeis ou unidades inquebráveis, quanto à detecção de 'estruturas', 'cenas', 'cenários', texturas, grupos, etc. Seja como for, o pressuposto é o de que não haja uma linha definida separando o que pertenceria ao estágio material do que pertenceria ao estágio estrutural ou formal.

A dificuldade de estabelecer um vocabulário próprio para acompanhar a produção de uma música eletroacústica poderá ser fatal se confiarmos rìgidamente em termos como 'material', 'matéria', 'forma', 'objeto', e tantos outros. Seu risco cresce na medida em que induzem a que se acredite neles enquanto termos de uma polaridade definida. Para ouvintes, e para muitos compositores, tais termos não podem representar conceitos estáticos. Ouçamos o que dizem:

> '...não se deve reificar nem o material, nem sua organização, durante um momento ou em um detalhe específico durante a realização de uma peça. Precisamos aceitar que a definição de seus limites deve ser deixada aberta e flutuante.' Chion (1991) p.38.

> 'Na música espectro-morfológica [eletroacústica] não há unidade menor consistente, e conseqüentemente não há um equivalente simples à estrutura da música tonal com sua estratificação hierarquizada da nota para cima, passando por motivos, frases, períodos, até abranger a obra inteira.'...'Uma 'unidade menor', ou objeto isolável, é muitas vezes difícil ou impossível de se perceber, particularmente quando em contextos musicais que operam com morfologias e deslocamentos íntimamente interligados.' Denis Smalley, no ensaio "Spectro-morphology and Structuring Processes", (Simon Emmerson, ed., 1986), p.80.

Resposta não-linear e o derrame.

O que é que acontece, no estúdio, que faz nosso entendimento vacilar? Esta discussão já tinha sido exaustivamente exposta por Pierre Schaeffer, quando afirmou que o objeto sonoro por si não pode garantir consistência após manipulação eletroacústica. Em

outras palavras: sons transformados em estúdio podem ser alterados ao ponto de erradicarem qualquer semelhança com o original. Isto porque, segundo Schaeffer:

> 'Um aparelho eletroacústico não é em si mesmo um instrumento musical.' (Pierre Schaeffer,1967), Face IV, n°71.

Hesitações conceituais de diversas ordens começam já no nível mais baixo da dimensão tecnológica. Na base da afirmação de Schaeffer está o que eu chamaria de 'resposta não-linear' de programas e aparelhos de transformação para composição eletroacústica, conotando o quanto a tecnologia está distante de coincidir com a percepção humana.

'Resposta não-linear' é a capacidade aberta da aparelhagem de estúdio diante dos processos e das estratégias de trabalho dos compositores. Por exemplo: quando uma só operação pode determinar transformações tanto no 'nível primário' quanto em eventos estruturados. Ou quando alterações em apenas um dos parâmetros acústicos podem abrir possibilidades em qualquer nível. Isto quer dizer que já o estúdio em si impede a concepção de um processo linear de composição que começa com pequenas células e desenvolve-se progressivamente em algo mais elaborado, porque o estúdio é a materialização da acústica e de sua incapacidade de ouvir musicalmente.

Experimentando com alguns programas de alongamento de duração (descritos no capítulo 2 de minha tese) revivemos imediatamente o capítulo 'Les Seuils Temporels de l'Oreille', (Schaeffer, 1967), que descreve transições perceptuais, como a que vai de iterações a grão, de grão a altura, e de altura a timbre, todas produzidas pela alteração de um único parâmetro da acústica: a frequência. E então fica bom lembrar que a intenção explícita do capítulo era a de denunciar a falta de uma correspondência direta entre os parâmetros da acústica e a percepção, criticando de passagem a pretensão da Elektronische Musik de identificar os parâmetros seriais com os da acústica.

Justamente por causa dessa identificação entre o equipamento eletroacústico e a ciência acústica, o preconceito que identifica estúdio com instrumento musical pode permanecer o maior obstáculo para a composição. Isto se expressa, por exemplo, em programas derivados do MusicV, como o Csound e outros, com os quais nos contorcemos em busca de resultados 'não convencionais', por prevalecer em suas formulações estruturais a dualidade orquestra/partitura. Confiantes numa 'linearidade' da resposta dos aparelhos, isto é, em uma suposta 'instrumentalidade' das máquinas, compositores correm o risco de cometerem os mesmos equívocos da Elektronische Musik e passarem ao largo do que poderia ser, justamente devido a sua 'não-linearidade', ferramentas para trabalhos originais. No entanto isto requer - como condição básica para administração de seus trabalhos - que compositores em estúdio não apenas reconheçam o derrame entre os estágios e os vários termos, como os já mencionados 'material' e 'estrutura', mas também se preparem para uma compreensão mais dinâmica - e talvez modesta - da dupla 'invenção' e 'descoberta', como veremos a seguir.

Por enquanto preciso lembrar a todos como seria fácil de ilustrar este texto com exemplos de peças onde compositores falharam nesta compreensão por confiarem na ciência da época, mas minha preocupação básica volta-se para problemas de outra magnitude. Até porque também foram muitas as falhas na produção dos compositores que se perfilavam do outro lado. Na verdade, o tempo decorrido nos privilegia, oferecendo-nos a medida de aprendermos com os enganos passados. Talvez o destino da música poderá deixar de ser conduzido pelo ricochete entre duas muralhas de intransigências. Neste exato momento posso me referir àquela encorajada por discursos que, veladamente, promovem uma discriminação institucional entre 'invenção' e 'descoberta'.

Esta separação tinha sido aberta como pano de fundo para a histórica disputa entre compositores da música concreta [de estúdio] e os que menosprezavam esta música por sua falta de 'determinação' composicional, isto é, por uma dependência da boa sorte de achados ocasionais. Uma obra da sorte jamais poderia disputar o mesmo status autoral de uma outra produzida pela 'écriture', na qual todos os eventos sonoros são atribuídos a decisões previamente escritas pelo compositor. Compositores da 'écriture', como Pierre Boulez, estarão mais próximos da invenção do que os meros descobridores favorecidos pela sorte. Adiante tentaremos compreender melhor o que possa ser a 'écriture'.

É inegável que muitas das queixas contra a música eletroacústica, especialmente a endereçada à atitude da 'musique concrète', não estão totalmente deslocadas, se considerarmos todos os abusos de poder em estúdio. É farto o número de peças onde propostas composicionais de maior fôlego mostram-se implausíveis, ainda mais se ouvidas hoje - passado o frenesi da novidade. Muitas não passam de ondas episódicas de sons tratados com muito efeito e agregados a eventos anedóticos, confiantes em uma ilusória expressão por força da novidade. A falta de vontade composicional ou de densidade de uma 'escrita', nestas peças, disparou reações até certo ponto justificadas, mas que se generalizaram contra um modo de compor que, na verdade, nunca se propôs, como um todo, a desabilitar a invenção. Entretanto, é notório que grande parte do repertório eletroacústico vem sendo preguiçosamente composto. Mesmo que apresentem por vezes belos esforços, são desligados e digressivos.

Contra esta fragilidade, localizada com precisão na ausência de uma determinação sempre oferecida pela notação, Pierre Boulez deposita fé - cheia de idealismo - em um estúdio habitado por máquinas qualificadas como 'instrumentos' para a 'invenção' musical, as quais serão enfim capazes de reunir as 'idéias' dos compositores ao 'material'. Sua recente aquiescência com respeito à abertura de espaço para a eclosão de 'acidentes' e 'descobertas' ainda mantém dois pólos distintamente separados, onde 'descoberta' continua sendo apenas a contrapartida da mais honorável 'invenção':

> 'A invenção musical deve propiciar a criação do equipamento musical de que necessita; por seus esforços, ela proverá o impulso tecnológico necessário que responda funcionalmente a seus desejos e imaginações. Este processo deverá ser suficientemente flexível para evitar a rigidez extrema e o empobrecimento de um determinismo excessivo, e para que possa abranger o acidental e o imprevisto, o qual deverá estar pronto para ser integrado posteriormente em uma concepção maior e mais rica. A longa preparação da pesquisa e a descoberta imediata não devem ser mùtuamente excludentes, mas devem afirmar a reciprocidade de suas respectivas esferas de ação.' Pierre Boulez, na introdução "Technology and the Composer", (Emmerson, 1986), p.11.

Com relação à questão terminológica, Boulez enfatiza o poder da racionalidade:

> 'O novo material sonoro esbarrou em possibilidades insuspeitadas - não por pura sorte, mas por extrapolação dirigida - e tende a proliferar por si mesmo; ele é tão rico de possibilidades que por vezes deve-se criar novas categorias mentais para podermos fazer uso delas.' Ibid, p.9.

Conforme já vimos, categorias mentais e descrições verbais encontram limitações diretamente proporcionais à complexidade do trabalho com recursos sonoros não-anotados. Mas não é verdade que, para explorarmos positivamente a riqueza do sonoro, nos seja imperiosamente necessária a criação de novas categorias. Estamos é diante de uma situação diferente, em que o que se apresenta como causa é também efeito. Quero dizer: a fortuna da música eletroacústica é encontrar seu campo de mineração na situação de trabalho em estúdio, na posse do som. O vazamento entre invenção e descoberta é conseqüência direta da comunicação de compositores com este som. Suas composições avançam enquanto os ouvidos se mantêm abertos aos sinais de escrita inerentes ao 'material' disponível. A

condição para a invenção em estúdio é uma prospecção de ingredientes que depende inteiramente da capacidade do compositor de operar com conceitos em flutuação e com máquinas imperfeitas, portanto de lidar positivamente com a 'descoberta'.

Ao aceitar, por fim, o 'acidental', Boulez parece ter-se tornado menos empenhado em disputas passadas, adotando posições mais próximas às que antes criticou. Deve-se no entanto lembrar que seu texto afirma que a música deve promover invenções tecnológicas que: '...respondam funcionalmente aos desejos e imaginações [da invenção musical]' Ibid., p.9. Preocupado com o uso de '...meios materiais que podem ou não estar de acordo com um pensamento musical genuíno,' Ibid., p.9., Boulez assevera que para estarmos de acordo '[com este pensamento genuíno]:

> 'Devemos notar que muito antes da tecnologia contemporânea, a história dos instrumentos musicais foi coberta de cadáveres: invenções supérfluas ou ultra-complicadas, incapazes de integração com o contexto de idéias musicais requerido pela era que os produziu; caíram em desuso porque não havia equilíbrio entre a originalidade [deles] e a necessidade [musical].' Ibid., p.9.

Para Boulez, o que melhor pode testemunhar a favor da 'música do pensamento genuíno' condiciona-se a se ela é ou não um produto da 'écriture'. Esta expressão, que tem sido amplamente utilizada no cenário da música contemporânea da França, não apenas se refere à composição como sendo um ato de escrita, como também tem servido como testemunho reivindicatório das decisões e determinações exatas do compositor. O termo às vezes é usado com implicações críticas à suposta 'indeterminação' da música eletroacústica. Pierre Boulez e seu tradutor para o idioma inglês avisam que:

> 'O termo 'musical composition' que aparece diversas vezes neste texto (para traduzir écriture) não se presta muito bem para esta palavra francesa, porque esta implica em um raciocínio simbólico.' (Pierre Boulez, 1987), p.171.

Adiante, o compositor Antoine Bonnet esclarece:

> 'A palavra francesa "écriture", usada como tal por todo este texto [a seguir], não é fácilmente traduzível para o inglês. Seu sentido inclui tanto o ato e o produto da anotação do pensamento, quanto uma espécie de raciocínio simbólico - o de pensar uma composição começando da manipulação de símbolos abstratos e discretos'. (Bonnet, 1987), p.209.

Qualquer compositor que, em estúdio, sentir-se desafiado a compor sem usar uma única vez algum raciocínio simbólico entenderá que esta não é de fato uma falta do idioma inglês - nem de qualquer outro idioma - e tampouco do estúdio. Quem, de nós, consegue fazer alguma coisa sequer que não esteja passando por alguma forma de raciocínio? E quem seria capaz de apresentar algum raciocínio não-simbólico?

Mas então o que faz aqui o Boulez, o que é que ele tem que haver com nossos problemas, nossas oficinas, composições e pesquisas instrumentais? Ele está aqui justamente por sua tentativa de propor uma orientação geral para múltiplos problemas, hierarquizando as músicas com a 'écriture' ditada do tôpo de uma máquina perfeita. Nosso consolo é que ainda não esquecemos que não apenas o fraco, mas principalmente o rico repertório da música eletroacústica tem contado com a ajuda do desprezado - mas ainda inesgotado - gravador, já lá se vão mais de quarenta anos. E quem se lembra da 4x, máquina inventada por Giuseppe di Giugno no IRCAM, então dirigido por Boulez, e que deveria funcionar como uma extensão dos instrumentos tradicionais?

O suporte simbólico.

Se os estúdios devem substituir 'máquinas' por 'super-instrumentos', então a música resultante será forçosamente comprometida com a 'instrumentalidade', e conseqüentemente com noções mais rígidas de 'matéria' e 'estrutura', etc, tudo outra vez. E se a tecnologia é quem deve ser a responsável por tudo, então ela terá que produzir um tipo de máquina/instrumento que facilite o modo de composição onde descoberta e invenção estarão mais funcionalmente integrados. Mas isso é esperar a salvação pela tecnologia. Tomando por outro ângulo, acho que as perguntas mais oportunas agora são: seria possível - e necessário - esperarmos por um desenvolvimento tecnológico que prestasse alguma atenção às peculiaridades sutis do ouvido musical? Não seria mais viável se simplesmente aceitássemos que o lugar do limite é entre o ouvido e a máquina, e não entre matéria e forma?

Ao invés de banirmos rígidamente o suposto 'oportunismo' do estúdio eletroacústico, teremos claro que tem sido de seus 'defeitos' que se conseguiu tirar algum proveito para a elaboração de um repertório. Não permaneceremos paralisados por escrúpulos exagerados, nem re-direcionaremos o problema para a tecnologia, e muito menos pretenderemos escrever a continuação da história da música. Pelo contrário, o que nos interessa é apenas estarmos em uma situação que desafia todas as nossas categorias, um privilégio a ser celebrado com música.

Esta pode ser nossa proposta em trabalhos, cursos e oficinas onde a experiência eletroacústica seja possível. Derrames conceituais serão para nós semente e fruto de uma só lavoura, onde nenhuma palavra do vocabulário é abandonada, nem mesmo as molas-mestras da metafísica: tudo é relativizado e reciclado.

A condição de intradutibilidade explicava por que, em outros capítulos de minha tese, as descrições e outras tentativas de apreensão do empenho musical são por vezes interrompidas lacônicamente, levando-os a um beco sem saída. Ser inteiramente 'objetivo' não foi o meu objetivo. Apesar do esforço gasto em busca de uma clareza de modo a expor minhas peças enquanto trabalhos cuidadosos de composição, o lugar para o imponderável sobrepuja a tudo. A dificuldade das músicas, e da eletroacústica em particular, nada tem que haver com a suposta ausência de um suposto 'raciocínio simbólico', testemunho da determinação do compositor no ato de compor, mas sim com a fundação de um suporte simbólico comum que possibilite a análise e a avaliação de peças do seu repertório. Eis aí moeda mais segura para quem estiver interessado na aquisição de um 'mercado' acadêmico. Enquanto persistir esta condição, as músicas intraduzíveis sempre apresentarão dificuldades para o exercício teórico; mas quem sabe não seria extamente este desafio a maior marca de seu interesse?

Conclusão.

É impróprio chamar de conclusão a este parágrafo porque nele nada se conclui. Antes pretende erguer pelo menos uma das pálpebras de nossa fé na continuidade histórica, a qual se daria por meio da extensão instrumental em seus múltiplos desdobramentos tecnológicos. A função do desfecho, pensando na presente situação deste encontro, limita-se a chamar a atenção para as propostas onde se oferece - mais uma vez e como em tantas ocasiões - a ilusão de um instrumento adequado para a arte, associada à uma suposta neutralidade do computador. Pode ser - e parece tão certo - que o trágico da arte esteja em seu destino inelutavelmente entrelaçado com o da técnica, mas ainda assim existe algo que deveríamos resguardar. Em nossa pequena passagem pelo mundo como fazedores e apreciadores de uma arte, talvez seja cauteloso confiarmos menos na técnica, nossa co-existente, como modo de assegurarmos, se não o contrôle - improvável - pelo menos uma leve impressão de poder de decisão. Sem apontar para vias regressivas, não custará muito investigarmos até que ponto se articulam mutuamente a estética do progresso e a

obsolescência tecnológica. Assim talvez saberemos por quanto tempo ainda nos será facultado o belo direito de escrever por linhas tortas. Confiamos então no ouvido, que não tem pálpebras.

Referências.

Bonnet, A. (1987). Music and Psychology: A Mutual Regard. *Volume 2, Part 1, Contemporary Music Review.*

Boulez, P. (1987). Music and Psychology: A Mutual Regard. *Volume 2, Part 1, Contemporary Music Review.*

Boulez, P. (1987). Technology and the Composer. *The Language of Electroacoustic Music*, ed. Emmerson, S. London: The Macmillan Press.

Caesar, R. (1992). *The Composition of Electroacoustic Music.* Tese de doutoramento. University of East Anglia, Norwich, Inglaterra.

Chion, M. (1991). *L'Art des Sons Fixés.* Éditions Métamkine/Nota Bene/Sono-Concept, Fontaine (France).

Schaeffer, P. (1967). *Solfège de l'Objet Sonore.* Ed. du Seuil, Paris.

Smalley, D. (1986). Spectro-morphology and Structuring Processes. *The Language of Electroacoustic Music*, ed. Emmerson, S. London: The Macmillan Press.

Agradecimentos.

A versão original deste artigo foi elaborada (em inglês) enquanto fui bolsista do CNPq, na Universidade de East Anglia. Agradeço ao compositor Patrick Ozzard-Low pelo interesse paciente e pela exaustiva orientação ortográfica.

DO SOM DO TEMPO AO TEMPO DO SOM

FLO MENEZES

Diretor Artístico do
Studio PANaroma de Música Eletroacústica da UNESP/FASM
Av. Lavandisca, 52 - apto. 74
04515-902 São Paulo - SP
Tel./FAX: +55 11 2412401
email: fmenezes@bra000.canal-vip.onsp.br

ABSTRACT:

This article intends to demonstrate that there was in the history of the ten early years of the electroacoustic music a transference of importance from the attack to the sustain of the sounds. It was only after this transference that the compositional thinking were in absolute convergence with the electroacoustic means. On the other hand, one can see that the instrumental music is more propense to the organization of the proportions between the very beginning of the sounds (their attacks) than to the extension of the sounds in time, so that a more accurate perception of the spectral qualities is more realizable in the electroacoustic music than in the instrumental one. This factor doesn't mean that one will abolish the another. Just on the contrary, it seams that the confrontation between both worlds can generate very interesting results regarding the perception of the inner textures of sounds. However, both the instrumental composition and the electroacoustic music can only produce good and permanent works if they are the result of an arduous elaboration of time and spectral structures.

Um dos mais evidentes exemplos da influência direta do pensamento weberniano sobre os primórdios da 'música eletrônica' de cunho serial, tal como esta fora desenvolvida no início dos anos 50 principalmente por Eimert, Stockhausen, Pousseur e Goeyvaerts, é a presença manifesta na obra de Webern, constituindo um de seus traços estilísticos, de texturas musicais onde a linha se contrapõe ao ponto (ou bloco). Já presente em Schoenberg, particularmente explicitada na quarta peça das *Sechs kleine Klavierstücke* Op 19 (1911), a textura da linha X ponto solidifica-se em Webern já mesmo desde suas *Sechs Bagatellen* Op 9 (1911-1913) para quarteto de cordas enquanto característica proeminente de seu estilo, perpetuando-se por diversos momentos em suas obras posteriores.

A bem da verdade, nesta dicotomia entre o som que permanece como uma linha e o bloco interferente, que se intromete, ainda que de modo esporádico e quase efêmero, em meio à existência perdurável do primeiro, têm-se de modo patente a oposição, presente mesmo no interior de quase todo fenômeno sonoro, entre a *ressonância* e o *grão* sonoro. Mas se a ressonância era praticamente circunscrita às possibilidades corporais dos instrumentos e dos espaços arquitetônicos antes do advento da música eletroacústica, tendo podido sofrer interferência compositiva propriamente dita somente com a utilização de aparelhos eletrônicos, através dos quais transforma-se comumente em *reverberação*, tal dicotomia deve ser vista, a rigor, como uma ainda mais lapidar e manifesta da vida dos espectros: como oposição entre o *ataque* e o regime de *sustentação* do som.

O curioso é percebermos que a problemática em torno desta crucial e básica oposição binária da constituição mesma dos espectros, amenizada somente em parte pela *primeira queda* (ou *decay*, transitório entre o ataque e a sustentação do som) e a *extinção* (ou *queda final*, *release*, transitório entre o regime de sustentação do som e sua (des)integração no silêncio), constitui o cerne da própria evolução estilística na ainda curta história da música eletroacústica. E isto quase que de forma cronológica.

Assim como poderíamos metaforicamente afirmar – como já o fizeram – que a evolução da história da música percorre, do Canto Gregoriano à emancipação do ruído preconizada no início de nosso século pelo futurismo italiano e, posteriormente, por Varèse, a própria constituição da série harmônica natural de um som de altura definida (som *composto* na terminologia de Eimert; som *tônico*, na de Schaeffer), da oitava à quinta (do Canto Gregoriano ao Conductus), desta à tríade (Renascença), da tríade à 7ª menor que tornará instável a tríade e consolidará a função de Dominante (Barroco), desta ao diatonismo por grau conjunto (Período Clássico), depois ao cromatismo (Romantismo), e deste último aos intervalos microtonais (Música Contemporânea), etc., não seria falso dizer que, em aproximadamente apenas uma década (mais precisamente de 1948 a 1959-1960), a música eletroacústica enveredou-se pela percepção, quase passo a passo, da vida dinâmica dos espectros no tempo.

Desvendado o inesgotável potencial dos parciais inarmônicos, dos ruídos, e escancaradas as possibilidades infindáveis de transição entre as regiões harmônicas mais seletivas (alturas) e aquelas mais estatísticas (timbres), deslocou-se o foco de atenção, antes voltada para a questão harmônica, para a questão do *tempo da escuta*, ou vice-versa da *escuta do tempo*. E é nesse sentido que Webern fora mesmo o grande precursor da música eletroacústica, reivindicado tanto pelos 'concretos' quanto pelos 'eletrônicos': contrastando permanência do som com as efêmeras interferências às quais este se submete nos contextos instrumentais, interferências estas que, por sua brevidade, aproximam-se do próprio silêncio, Webern acabara por despertar a atenção para as distintas dimensões do tempo sonoro tal como este existe na própria vida dinâmico-temporal dos espectros.

Assim é que, tanto teórica quanto praticamente (composicionalmente), defrontamo-nos no decorrer destes poucos anos com um caminho que vai, a grosso modo, da importância do ataque à percepção prioritária da sustentação do som. No nível teórico, duas abordagens do fenômeno merecem aqui ser pontuadas: de um lado, na vertente da música concreta, temos a importância do ataque tal como nos fora revelada e relevada por Pierre Schaeffer em seu 'Traité des Objets Musicaux' (Éditions du Seuil, Paris; e que muito embora date de 1966, na realidade remonta aos desenvolvimentos teóricos de Schaeffer desde o início da música concreta); de outro, pela corrente da música eletrônica com seu pretendido *continuum* musical, deparamo-nos com a teoria da 'Unidade do Tempo Musical' de Stockhausen (em: 'Texte zur elektronischen und instrumentalen Musik', Band 1, DuMont Verlag, Colônia, 1963, pp. 211-221), elaborada em 1961 a partir da realização de sua obra *Kontakte* (1959-1960).

No caso de Schaeffer, tanto na parte referente à análise das anamorfoses temporais (e em especial nas considerações acerca dos sons do piano, pp. 216-243), quanto na relativa aos critérios de ataque e de perfil (pp. 532-546), têm-se o incontestável mérito de se apontar pela primeira vez para a importância inestimável que o ataque dos sons exerce na percepção de seus timbres. Seu *pendant* prático é constituído pelo *Etude Violette* e pelo *Etude Noire*, baseados na manipulação de sons do piano, e que constituem os terceiro e quarto movimentos dos *Cinq Etudes de Bruits* (1948).

No que diz respeito à teoria da unidade do tempo musical, Stockhausen procura demonstrar, com muita pertinência, que percepção rítmica e percepção freqüencial nada mais são que faces de uma mesma moeda: elas constituem, na verdade, regiões ou categorias perceptivas diferenciadas de um mesmo universo sonoro, de um *continuum* musical que, aliás, faz com que esta moeda adquira outras tantas faces: para aquém da categoria rítmica, incorporando-se os sons nas macro-estruturas, têm-se a região das formas; para além da categoria perceptiva das freqüências, adentrando-se na constituição harmônica dos espectros no nível de suas micro-estruturas, têm-se a percepção timbrística. Neste sentido, as freqüências podem ser vistas como ritmos extremamente acelerados, em que a percepção funde os pulsos numa sensação de periodicidade interiorizada, enquanto que os ritmos serão vistos como articulações de baixa freqüência, em que cada período se individualiza como pulso particular. A inquestionável contribuição desta teoria reside no fato de que, nas regiões 'transitórias' entre as dimensões rítmica e freqüencial, o ouvido pode, com certo esforço e concentração, adentrar-se na íntima percepção dos espectros, quebrando as barreiras impostas pela tradição e pela prática da música instrumental às categorias perceptivas do ritmo e da freqüência, e degustando, assim, os sons pelas suas próprias *faturas*, tanto – para utilizarmos os termos de Schaeffer – no nível dos grãos (*grains*) quanto no nível da percepção do dinamismo interno, das sutis flutuações dos objetos sonoros (*allures*).

Sob este ponto de vista, entretanto, a postura de Stockhausen representa, a rigor, uma conseqüência radical do pioneirismo de Schaeffer, consolidando uma importante transição estética no seio da música eletroacústica: da importância conferida ao ataque e de sua conscientização pelo compositor e pelo ouvinte, transita-se agora em direção à percepção do som em seu regime de permanência, em sua *duração*.

Esse passo decisivo, efetuado mormente por Stockhausen como conseqüência de uma prática composicional cujos reflexos são claramente observáveis também em obras de outros compositores deste período (Berio, Maderna, Pousseur, Eimert, Koenig, Ligeti, Kagel, e tantos outros), na verdade apontava para o caminho natural da estética eletroacústica, demonstrando uma real convergência dos anseios estéticos com os meios técnicos de realização das obras: ao contrário da música instrumental, circunscrita às formas possíveis de emissão sonora e, portanto, limitada no que se refira à sustentação dos sons, as manipulações eletroacústicas favoreciam o mergulho interno do compositor na estrutura mesma dos espectros, interferindo-se de forma essencial (não simplesmente contingencial ou metafórica) não somente no ataque dos sons, mas também e principalmente em seus regimes quase-estacionários ou quase-periódicos, ou seja, em sua *sustentação*.

Aquilo que havia nascido há tão pouco tempo experimentava, assim, um primeiro renascimento. E, desta forma, constituía-se uma bipolaridade que, a bem da verdade, traduz as duas principais 'linhas de força' de toda e qualquer composição musical no confronto de suas concepções com a questão do tempo: ou centra-se questão no tempo enquanto estruturação rítmica dos sons, em que a distância e a proporção temporal entre o ataque dos sons adquire papel preponderante em detrimento da sustentação dos mesmos, ou se opta pelo contrário, onde o ritmo cede lugar à duração dos sons e o tempo passa a ser percebido enquanto condição essencial e dinâmica da própria constituição dos espectros. Neste último caso, a percepção do pulso cede passo à da fatura, a percepção rítmica dá lugar à textural, fazendo com que cada grão sonoro, cada *ponto*, como preconizara Webern, se insira na *linha*, no tempo musical, enquanto condição de textura. Em suma: a importância do *ataque* transfere-se para a *sustentação*, o *ritmo* de outrora dá lugar à *duração*, condição de modernidade, e o som que era do tempo cede passo ao tempo de cada som.

Óbvio está que a cada maneira de abordar o tempo corresponde, em princípio, um 'gênero' musical, com suas especificidades e suas limitações. Sob este aspecto, a escritura instrumental está para a organização rítmica assim como a latente 'escritura' eletroacústica para a categoria das durações. Possíveis incursões criativas no domínio autônomo desses gêneros musicais podem resultar em singulares obras. Mas não teria sido este o principal pivô das transformações estilísticas no decorrer de toda a história musical: o permanente conflito entre a intenção compositiva e os meios através dos quais a idéia se cristaliza? Não seria no seio deste conflito que residiria o teor propulsor do virtuosismo instrumental, tanto no nível da escritura, da concepção em si da obra, quanto no nível de sua mediação pelo intérprete? Não seria no bojo deste confronto entre intenção e resultado que, instrumental ou eletroacusticamente, se alojaria a invenção?

Dizer, pois, que ao emancipar o tempo de suas conotações rítmicas, deixando os pulsos decisivamente para as abordagens mais mercadológicas e temporalmente imediatas, a música eletroacústica aboliria a música instrumental, seria, no mínimo, presunçoso. Se o universo eletroacústico emancipa o que em Webern fora apenas preconizado, Webern preconizava aquilo que, em Brahms, era apenas longinquamente acenado, e que, em Gluck, certamente nem existia. Uma certa *evolução* indubitavelmente se perfaz, e a noção mesma de *progresso* não é de todo impertinente. O totalitarismo da *exclusão* é que nos parece, contudo, deslocado. O confronto é justamente propulsor pela existência das partes conflitadas, não pela exclusão de uma pela outra. A uma determinada condição acresce-se uma certa intenção que, mais ou menos condizente com os meios aos quais se propõe, produzirá ou não um maior ou menor conflito.

Este não seria o primeiro exemplo, desde o advento da música eletroacústica, em que o universo eletroacústico empresta suas preocupações e suas posturas estéticas ao terreno da música instrumental. Configurando-se como gênero autônomo mas não necessariamente excludente, a música eletroacústica, ao mesmo tempo em que emancipa aspectos da música instrumental e consolida outros tantos específicos de seu próprio universo, acaba por transferir suas irrevogáveis aquisições para o âmbito da escritura instrumental, mais velha mas não por isso menos susceptível de influência, dimensão instrumental esta que, por sua vez, pode se demonstrar o suficientemente flexível para que se submeta a complexas evoluções de seus próprios elementos composicionais, inclusive no que se refira à questão *rítmica*. Daí a importância, por exemplo, de um Messiaen... Se a referência rítmica se demonstra quase que inevitável no âmbito instrumental, que ela resulte em contínuas

evoluções de seus dados de linguagem, fazendo com que a proporcionalidade entre o tempo dos ataques constitua, por fim, um dado de textura.

Assim é que o confronto entre a dimensão instrumental, em cujo universo a categoria da percepção rítmica encontra-se de certo modo embutida, e a eletroacústica, propensa à percepção do som enquanto duração, enquanto fatura, pode trazer e traz, via de regra, intrigantes resultados. O compositor se depara aí com uma *potencialização* dos instrumentos, baseada principalmente nos fatores concernentes ao tempo e ao espaço, e mediados pelas incursões nos domínios da harmonia e do timbre. O ouvinte se vê envolvido por uma sala de inúmeros espelhos, projetando sua imagem acústica, sua identificação corpórea, no próprio instrumento, susceptível a infinitas angulações.

Nada, entretanto, que possa ser resolvido por si só. Tanto a profícua interação entre o instrumento e o universo eletroacústico quanto a opção puramente instrumental ou puramente eletroacústica enquanto gêneros autônomos da composição terão uma mínima chance de resistirem à rigorosa peneira da história apenas e tão somente se se submeterem a um ardoroso trabalho de *composição*: em suma, a uma complexa elaboração.

São Paulo, em maio de 1995

Laboratorio de Música Electroacústica del Departamento de Música de la Universidad Nacional de Córdoba

Martin Alejandro Fumarola
Estafeta 56
RA-5001 Córdoba
Argentina
Phone: +54 51 242388
Faxes: +54 51 251957 and 235370
E-mails: maralefu@famaf.uncor.edu and maralefo@turing.fis.uncor.edu

Abstract

The *Laboratorio de Música Electroacústica* is the main open studio of electroacoustic and computer music among all those belonging to Argentinian universities and the second in importance in the country after the LIPM. The activities at the lab can be broadly categorised into three different areas: 1) composition, 2) teaching, and 3) general promotion of electroacoustic music. In this paper, the current profile of activities and projects is outlined, emphasizing on those related to composition. Moreover, new and recent acquisitions are mentioned.

Introduction

The lab is above all an open working place for composers of electroacoustic and computer music. It is open for both members and non-members of the Music Department of the National University of Córdoba so any composer with enough background in electroacoustics may work in it. The lab was opened under the direction of the Uruguayan composer Ariel Martínez, who was also in charge of teaching electroacoustics in the Music Department. The lab is placed in the same studio room that belonged in the 1970s to the famous "Centro de Música Experimental", where the composers Horacio Vaggione and Oscar Bazán worked. The lab is member of the FArME (the Argentinian Federation of Electroacoustic Music) network of electroacoustic music centres.

Facilities and Equipment

Until the new studio room is opened, the lab occupies only one room. Space is available for both work and recording studios based on MIDI commercial equipment. Final mixes of material generated with MIDI equipment can be produced. During the last year several important additions have been made to the lab equipment. Its heart is a 16 channels TASCAM M-1516 console, with an attached 8 track TASCAM TSR-8 tape recorder. There is also a TASCAM DA-30 DAT player and recorder. Besides a YAMAHA TG77 tone generator was bought, in addition to the already existing YAMAHA DX7 synthesizer. Most commercial musical software for MS-DOS and Windows 3.1 environments is available now, such as VOYETRA Sequencer Plus Gold, Cakewalk Professional both for DOS and Windows, X-Or, Encore and Finale. All those programs run in a 286 PC computer but the purchase of a Pentium is planned for the near future. Finally, a Multisound TURTLE BEACH DSP card will be added as well.

Activities and Projects

The lab is mainly dedicated to the composition of electroacoustic and computer music (tape alone and mixte works). Several pieces of international trascendence have been composed by Oscar Bazán, Martín Alejandro Fumarola and Jorge Naparsteck. A few seminars under the direction of Diego Loza are given at the lab in cooperation with the LIPM. These seminars are mainly directed towards beginners and people with limited experience. It is expected that the lab will be fully operational by the beginning of 1996, when the new studio room is opened. In this sense, the incorporation of non-commercial musical software, such as the PC-MUSIC (CMUSIC for MS-DOS) developed by Pietro Fischetti (DIST, University of Genova, Italy) and an extension of the language Csound for DOS platforms allowing the musical utilization of the granular synthesis, developed by Franco Degrassi (University of Bari, Italy), which is planned by the end of this year, will be a significant contribution to the lab. A joint venture with the FaMAF (Faculty of Mathemathics and Physics) is also foreseen because in that Faculty there are NeXT workstations running version 3.2 of NeXTStep operating system including the version 4.1 of the Music Kit and with full access to the Internet.

CAC - MIDI MUSIC
The Computer without algorithm

NELSON CAVALCANTI DE ALMEIDA
UNIVERSIDADE FEDERAL DE PERNAMBUCO
DEPARTAMENTO DE MUSICA
Recife - Pernambuco - Brasil
Post code: 50 607-901
Phone number: (05581) 2718300
Fax number:
(05581) 2718300 - 2718308 - 2718142
E-Mail: 16almeida@npd1.ufpe.br
Department of Music
Keele University - Staffordshire
ST5 5BG - England
E-Mail: Mud00@cc.keele.ac.uk

Abstract

There exist a great diversity of uses of the computer to make music. Nevertheless, this diversity falls into two broad categories. One of them is concerned with processing an input a sequence of instructions, expecting an output, by means of a programming language. This type of music making is known as algorithmic. Another way is to use the computer in networked devices able to understand each other by means of a common language. This common language is MIDI which stand for Musical Instrument Digital Interface. MIDI opened a new and soft way to make computer assisted music

This Essay describes three aspects of computer music making: selecting sound, composing and performing, using a MIDI platform. A brief incursion into algorithmic environments is carried out in order to clarify some convergent and divergent points between non-algoritimic and algorithmic composition.

Introduction

Montague (in: Pope, 1994) has stated that in the 1960s many of the people interested in computer music were technicians or engineers and the music produced sounded like music made by technicians and engineers. I believe it was, among other reason, because dealing with that technology was a hard task. However, nowadays, the technology applied to music has improved. Electronic devices, computer hardware and software are now commercially available, price has decreased becoming attractive to many musicians. Nevertheless, this facility in handling brings with it a danger for new music. It is still concerned with technical quality, since the digital music world has attracted anyone more beyond technicians and engineers. This is made worse if one considers that many of the new users are not experts in either music or technology.

CAC - Computer Assisted Composition

According to Dodge & Jerse (Dodge & Jerse, 1985) the main fields of the computer music are three: Sound synthesis, Composition and Performance. There exist three classes of software as well: algorithms for sound generation, programs to assist composer and program which allow performance of the compositions. The above quoted types can work both in algorithmic or non-algorithmic environments The composer is free to choose between using the processes either in part or the whole of his composition.

From this point of view the composition may be done by the computer, which Cope has called machine composition (Cope, 1991); with the computer, generally known as computer assisted composition; and for the computer, that is music to be performed by the computer.

These three related aspects of computer music have a relationship with both algorithmic and non-algorithmic techniques. When we use, for example, random processes or deterministic operations we are working in an algorithmic environment. If we use MIDI or Digital Sound Processing we are in non-algorithmic environment. Once again the composer is free to use one or both systems isolated or gathered together, either as part or as a whole of a composition.

Computer Assisted Composition in an algorithmic environment, albeit its diversity, falls into two broad categories: aleatory and deterministic music.

In aleatory music the computer generates events based on statistics using random processes. In deterministic music the computer receives a set of predetermined compositional elements, such as rhythm, melodic or harmony patterns, pitch etc. It then performs determined calculation and yields a result.

From a strict point of view the computer can perform nothing without following instructions, so in theory, there is no computer without an algorithm. In fact what exist is a set of translators between what the user wants to perform and the final response of the machine. These translators range from the nearest machine language level up to 'clic-on-icon' programs. The user gives a sequence of instructions using an appropriated computer language or just clicks on an icon.

The difficulty of the task of giving a sequence of instructions to the computer depends on the level of that language. Nowadays applications tend to be less demanding from the point of programming and more user friendly.

For example, one can simulate a procedure to get a musical event using an imaginary computer language:

```
[Note1]
Play  p = 440    d= 120  v= 64
Play  p= 0       d= 60
Play  p = 880    d= 240  v= 127
End
>Note1
```

Where:
p= pitch
d= duration
v= velocity

A very simple algorithm to obtain, as an output, A4 with a duration of 2 seconds and a dynamic level of *piano*. It is followed by a rest of one second and by an A5 with a duration of 4 seconds and a dynamic level of *fortissimo*. It seems very easy to do, but the task becomes more difficult when we try to get something slightly more complex, even when it is as simple as "Brother John".

In order to enhance the previous example, on can change the data in that sequence by typing:

```
[Note2]
Sort    p = (170-1001)   d = (15-240)    v = (30-127)
repeat 10
End
>Note2
```

As a result the new sound will consist of different pitches, durations and dynamics.

These are very simple examples of algorithm composition.

I have been involved in the past in the development of an application for children (of any age) to compose using the computer. The program is called "Sound Letters". In programming it, a sequence of algorithms was associated with the first seven letters or the alphabet. Children just need to type a sequence of the first seven letters of the alphabet, in any order, instead of typing an instruction like play p= 440, d= 120, v= 64, and so on. In response they get their music. Of course other features, such as different octaves, tempo, etc. are available.

The program was developed using Pascal, in an algorithmic way, but the outcoming music is a clear example of non-algorithmic computer assisted composition.

MIDI Music

The first thing to bear in mind when working with MIDI and, in general with computer, is that the latter is not a wizard, it has no power to turn one into a great composer. This fact can be seen as the dilemma of this century. The quality of the music which has been done with so much technology. Nevertheless computers in general, and MIDI in particular, have no doubt brought new directions and easy to music making. MIDI must actually be considered as a powerful tool to enable musicians to experiment and realise sound generation, composing and performance.

Computer MIDI systems can perform a full orchestral score, or any combination of instruments. Synchronisation with images, in video and cinema, is now a easy task. Likewise acceleration or slowing of the tempo in order to adjust music to image, is now simple to achieve. Since MIDI music is recorded track by track and each track is recorded event by event it enables complete control over the music, ranging from the shortest of notes to the whole composition, including any individual parameter. Obviously this music still lacks human touch; however we must realise that it is not the same thing that listening to a live orchestra performing. In fact no recorded performance should be compared with a live performance.

MIDI music can be perfectly applied in a professional and commercial context. But it is not the only way: MIDI is a powerful tool for academic purposes to help teaching in matters such as perception, counterpoint, classic and contemporary harmony, aesthetics, analysis, music education and, of course, composition.

A MIDI workstation is not cheap, in fact it is very expensive considering the economical reality of many enthusiastic musicians. However, the price of the electronic devices often decreases barely one or two years, after they have been launched. Nevertheless one must decide what one really wants to do before buying MIDI system.

For players, a simple but good MIDI system would be a rack of expander and effect units under a keyboard, MIDI Guitar or wind controller. For education purposes a computer running appropriate software connected to a MIDI keyboard may be good enough. For composers, the above platform may be enlarged. The computer must run a Notation type program, Sequencer software, and digital sound processing software. A multi-timbral controller keyboard, at least one expander and a printer are also necessary.

Sound: Generation and Manipulation

Computers are actually able to generate and manipulate a large variety of sounds to be performed in musical context. Generation and manipulation of sound events can be done by a digital sound processing program or using a sound synthesis generator program.

Digital sound processing programs record and edit sounds, from any source, sampled via microphone or line inputs. Most pieces of software of this kind show the waveform of the recorded sound which can be altered by dubbing, mixing, cutting, pasting, adding effects, changing pitch, time, and so on.

Sound synthesis is the generation of a signal with acoustic quality.

Sound can also be generated and manipulated by voice editor program or using a MIDI expander, which may be a synthesizer or sampler unit.

A sampler is a MIDI device able to recorded sound as MIDI event. A sample is closest to a natural sound but a large amount of data is required when sampling. The sampler is required to have a large memory and editing is a challenging task if one is to obtain convincing results.

Sounds generated by MIDI samplers are saved in MID format, so that they are suitable for use with MIDI sequencing or notation programs as well in live performance using a MIDI controller.

Synthesizer units provide resources to generate and to manipulate sounds by adjusting parameters using increasing and decreasing push bottons supplied. The adjustments are shown in LCD displays. The generated sounds are stored into the internal memory of the synthesizer, into an external cartridge or into a computer memory via specific software and MIDI connexion. As in the case of samplers the new sounds are saved in MID format suitable for use in the same circumstances. To synthesize sounds may sometimes be an unwanted task, especially without editing units now available for some instruments. In any case, synthesizer usually to provide a large quantity of factory preset sounds, both in the internal memory or as external data storage, to aid the user's task.

Voice editors are computer programs which allow the musician to program sounds quicker and easier than with synthesizer modules. They mimic controllers over all synthesized sound parameters.

Besides the advantage of having more memory available in the computer, all the control knobs and switches are present on a large screen to be handled both by the computer keyboard or via the mouse. Another useful feature is the capacity of changes to envelope shapes using the mouse.

Voice editor pieces of software are able to sent any new generated sound directly to the synthesizer to be heard on request.

As stated above, voice editors save the file in MIDI format.

Composition

I will divide this writing into three broad parts: digital music composition (DMC), acoustic music composition (AMC) and digital-acoustic music composition.

Digital music composition (DMC) consists of the use MIDI instruments instead acoustic ones. Instrument, in this context, means exactly a timbre you choose from a MIDI device (in an algorithmic context, the term instrument means an algorithm that realizes a musical event). MIDI instruments are about 32 notes polyphonic and up to 16 parts multi-timbral. In others words they are, in fact, many instruments built into one unit. sixteen different instruments sound as a big orchestra. Each one of this 16 instruments is able to play up to 32 notes at once. It is like having 32 first violins or 32 trumpets playing (32 voices in a *divisi)*, each voice has independent control of parameters such as duration, pitch and dynamics. The voice or voices of an instrument are assigned a track in the software which often provides more than two hundred tracks. The 16 MIDI independent channels receive one or more of these tracks and play them back using a timbre which the composer choses. All 16 MIDI channels work via one port available in the software. Since the up-to-dated programs has at least two ports it means that we can get at least 32 independent channels or 32 different instruments playing together. One can get two or more different timbres for the voices within a track by connecting an expander via a THRU connector. Another simple way to do this is to copy one track to another and set a different channel for the latter.

Having finished the set-up of the instruments and effect processing devices, the next step is to chose the appropriate software.

The composer can choose between a sequencer, or notation software, or both, depending on what he intends to do. Let us assume that the best choice is a sequencer. This device works as if it was a multi-track tape recorder. Working with a sequencer, the composer sets up the MIDI orchestra by selecting the number of track he needs, choosing channel, timbres, etc. Secondly he input music by playing on the master MIDI instrument or using any step time process. Even using a real time method he can record in slow tempo in order to avoid mistakes. Changes in tempo do not alter the real pitch. Even in polyphonic passages, piano-like, he can play line by line having total parameter control by setting one track per music line. In the end, or at any time, he can access any editing mode.

The sequencer provides two basic kinds of editing facilities: edit event and take note. Both give the composer the ability to change data, at event level, such as pitch, duration and velocity. Event editing is provided in form of text or text/graphics screens that exhibit the data of a track respectively in text only (alpha-numeric) or in text graphic (horizontal bars) display. The value of each event can be increased or decreased both by the computer keyboard or by mouse.

'Take note' is another type of editing. It is in the form of traditional music notation. In it, events can be changed by using the computer keyboard or by mouse-dragging from a pallet provided. 'Take note' works as notation programs do. However there exist quite a few differences with the latter, since take note is just a tool within the whole software while notation is itself the complete program. Similarly one can find sequencing capabilities in notation packages. The differences between notation and sequencing is becoming blurred as most software comes now in one integrated package. Anyway, most sequencers can save files in MIDI format which can be loaded into the majority of notation packages and converted into traditional music sheet, if desired.

At this point the composition is quite ready to be performed. A sequencer itself can do this by acting as a multi track tape player. Again all the facilities described above, and other I have not mentioned, make this task easy for the composer. The main screen of a sequencer has bottons, just as the front panel of a tape deck, which allows change of time, fast forwarding, rewind, pause, accurate identification of song point, etc.

In order to write acoustic music composition (AMC) the composer may choose a notation program rather than a sequencer or he will use the former after using a sequencer, since the score will need to be printed out. Unlike Digital Sound Composition the choice of the instruments, timbre and effects occurs outside of the computer or electronic environment. However composing, editing, printing out and even producing a demo tape can be completely assisted by a computer and MIDI devices.

The workstation necessary for AMC may be simpler than that in DMC: a computer controller, a multi-timbral MIDI keyboard, a printer and an audio system is just what one needs. Of course, improvement this platform is welcome. The computer must be equipped with MIDI interface and software as well. Since AMC is conceived to be performed outside of the computer network I will just describe what is necessary while the composition is being built.

The keyboard does not need specifications as touch sensitive or after touch since the notation program may ignore its results. The simplest way of changing velocity is writing down dynamic marks along the score. The notation program will recognise them. We can change the parameter from a dynamic marks pallet by changing the default value provided. The MIDI keyboard represents nowadays more than the acoustic piano has represented along the 300 years for the composer, as it does not work alone but directly in connection with a multi track tape recorder and an animated score sheet. Both, recorded sound as well the score can be stored in disk data to be revised, modified or just to be continued after a break. The multi-timbral unit should be provided with a good library of acoustic sampled sounds. For example, while writing for real flute, a sample of this instrument may give a good idea of how it will sound in reality, mainly in relation with the timbre of the orchestra. However the composer must to keep in mind that it is just a virtual orchestra. The phrasing and articulation in real acoustic instruments are completely different, even In the case of the best sampled sounds.

In addition the composer's workstation provides facilities to record the finished composition on a tape, using MIDI sounds, to be used for purposes such as teaching, symposiums, commercial demos or just for pleasure.

Turning back to the notation programs, there are two kinds. One is just a note processor, similar to the word processor. It does not provides sound facilities but is rather useful in order to get a good publishing presentation.

The common kind of notation program handles multi-staves display, has all the ordinary music symbols, defines key and time signature, plays scores via MIDI, has automatic beaming and spacing, has a simple word processor, can transpose, copy, print, save into MIDI format and so on. Although all these advantages, notation still present many problems. One of them is its tendency to want to do everything automatically. Another common problem concerns the word processing, which is incompatible with the graphic symbols when the document is reformatted, repaginated or changes the size of the music fonts, for example. There is a notable lack of contemporary symbols; however some software increases this features by allowing the composer to draw his own notation. Flexibility can be achieved by allowing the user to chose between automatization for some features. Another good idea, I believe, is the capability to copy to a clip-board allowing the user the possibility of using an integrated desktop publishing package. I am hoping for a notation program which supports at least the two last features quoted. But it should not be only a single note/symbols processor. It should allows drawing symbols created or edited in a pallet provided or imported from a real graphics editor to be associated with a sound file. These features may be improved by allowing imported symbols from graphic editors, such Tiff format, as well as by the use of any true type font. The new program may provide a interface in order to create, edit and import sound files both in MIDI and DSP (digital sound processing) format from a Voice Editing program. This sound file would be associated with the symbol created/edited and the Notation would recognise it while in play back mode.

In order to produce digital-acoustic music (DAMC) composers may proceed by using some features present in both, acoustic and digital composition. I have spoken about the two processes separately. I have quoted the sequencer kind of software as the best option for making digital music. I have pointed out the notation kind software as the best choice for making acoustic music.

There are three steps in digital and acoustic music composition. The first consists of setting up the instruments, the timbres and the parameters. The second consist of choosing an appropriated program. The third consists of providing the performance. In digital-acoustic composition the process is not different. In fact, all we need is to gather together the two former processes, or least part of them. As the name indicates digital-acoustic music means music with both digital and acoustic sounds. The initial process of setting up sounds and parameters was explained above. The choice of software was also covered. In this composition the best option is a sequencer-notation integrated program, or using both simultaneously. The digital-acoustic composition takes the acoustic instruments in studio for a tape recording or takes the digital sound modules to the stage, for live performance.

Performance

Performance is the last phase of the composer's goal. Dealing with computer music, or electronic music as a whole, means to work with electronic devices in addition to people; often, more devices than people. Besides finished tape recordings for video, cinema or home users, computer music targets live performance on stage. This performance can use just recorded sounds or a mixing of recorded and live sounds.

Mari Kimura, who is a violin player working with both acoustic and MIDI violin, has stated (Kimura, 1995) that, as in old media, computer/electronic music performance has common problems concerning the sound. 'Performing electronic music involves working with sound sources other that the performer's own instrument. In live music for instruments and tape, and with live electronic such as integrative computer systems, instrumental performers have very little control over the synthesized or recorded sounds'. She argues that the environment affects these sounds considerably and differently than it affects traditional ones. The performers need to make adjustments in computer music rather than in acoustical performance. 'One must understand the difference, and compensate accordingly', she says.

Other particular characteristics of performing computing music have been discussed by composers and performers. No inclusion of computer music in the annual program of orchestras and chamber ensembles has been pointed as a generic problem applied to contemporary music performance. Lack of performers' interest, unsuitable concert halls and lack of audiences are quoted as main problems.

However the number of musicians and listeners involved in computer music has increased. New people are fascinated mainly by the novelty and, of course, by the publicity.

Conclusion

I was led to write this paper by a wish of organize a set of ideas concerning computer assisted composition and its relationship with algorithmic or non-algorithmic techniques

I must confess you that it was not a easy task for me. I am still worried about what it is that people related to computer music would like to listen, or to read.

To the experts my apologies if I said no more than the obvious. However it seems to me that, sometimes, experts hold information which could be useful to other people, just by considering it extremely evident.

My intention was to write for people wishing to join us whether it is a composer, performer or just a listener trying understand a small bit of the world of computer music, which despite its almost half a century of existence, still remains a new branch of musical science. It is frequently up-to-date and has improved the means of music making, therefore I consider that the first word was spoken, but the last one has not been uttered yet.

References

Anderton, Craig. (1986). *MIDI for Musicians*. London, Amsco Publications

Ballista, A et al. (1992). A MIDI/DSP Sound Processing Environment for a Computer Music Workstation *Computer Music Journal* 16:3, pp 57-72

Brummer, Ludger. (1994). Using a Digital Synthesis Language in Composition *Computer Music Journal* 18:4, pp 35-46.

Byrd, Donald. (1994). Music Notation Software and Intelligence *Computer Music Journal* 18:1, pp 17-20.

Chimera, Marie. (1995). Performance Practice in Computer Music *Computer Music Journal* 19:1, pp 64-75

Cope, David. (1991) *Computer and Musical Style*. Oxford. Oxford University Press

Dodge, Charles & Jerse T.A. (1985). *Computer Music: Synthesis, Composition, and Performance*. London: Collier Macmillan Publishers

Moore, F.R. (1988). The Dysfunction of MIDI *Computer Music Journal* 12:1, pp 19-28.

Pope, Stephen Travis. (1994). Music and Computer: NEMO Composers' Panel Discussion *Computer Music Journal* 18:3, pp 33-44.

Penfold, RA. (1989) *Computer and Music*. Sussex: PC Publishing

Penfold, RA. (1995). *Practical MIDI Handbook*: Glasgow. PC Publishing

Penfold, RA. (19.91. *Advanced MIDI*: Users Guide. Exeter: PC Publishing

Wright, Matthew. (1994). A Comparison of MIDI and ZIPI *Computer Music Journal* 18:4, pp 86-91.

Acknowledgements

I would like to thank Universidade Federal de Pernambuco, Brasil and Conselho Nacional de Pesquisa-CNPq for support my research at Keeel University, England.

Music without body: composition, computers and instrumental practice

CARLOS CERANA
Laboratorio de Investigación y Producción Musical (LIPM) - Buenos Aires
email: rqcerana@arcriba.edu.ar

GABRIEL PERSICO
Conservatorio Nacional - Buenos Aires

Abstract

This paper poses a theoretical discussion upon some aspects of computers' usage as a tool for musical composition, by analyzing its consequences to the education of composers, to the musical ideation process, and to the communication of musical work to the audience. Computer music brought --as one of its possibilities-- the performing of music without a player. This gave the composer the responsibility of making performing decisions in the very act of composing. The authors question the possibility of an exclusively intellectual creation, generated without experiences that relate sound and body. Extremely, the actual possibility of developing a training in composition and of operating on complex musical structures, without having ever played an instrument. As a conclusion, the paper stresses the relevance of instrumental practice as a significant tool for structuring musical thought.

"...music-making -especially children's music-making- is essentially a social activity. When new technology is introduced, this social dimension can be lost -the interaction is happening between the player and the computer screen rather than between people..." (Swingler, 1994).

Introduction

This paper poses a personal view on some implications of computers' usage in education and training of composers. The authors reflect from a pedagogic point of view. One of them is an electroacoustic composer and worked on children's music initiation, the other is a music educator and a flutist, specialized in performing early and contemporary music.

The growing interest towards musical creation with digital media has originated plenty of courses and careers on this particular field. The explicit social request is towards *music-making*, but it is also possible to recognize in it the fascination of modernity and of the usage of computers, an iconic tool of our times. Curiously, such courses and careers do not pose any requirement of instrumental skills for their students. The idea of an "easy access" to music production seems to have completely discarded any training or experience on playing instruments.

The authors wonder about the following questions. Is the present trend a healthy one? Could composers' education without physical experience with instruments be a complete one? Would not be something lost? Could music ideation be a purely mental fact, without the intervention of human body, even as a memory of corporal processes implied in performing?

Or is this a retrograde position? Is this the point of view of a musician that feels his place of knowledge and power invaded by new barbarians, not knowing or valuing the traditions of his art?

Composers without body

Computers' applications in the music field have enormously facilitated the access to different stages of musical production. Nowadays, the user of a digital system, amateur or expert, can interactively operate with it for realizing musical structures. This possibility has been welcomed as a "democratization of the access to the whole musical process -for a long time reserved to a kind of specialists, the musicians" (Iazzetta, 1994).

It is far from our intentions to condemn the dissemination of music knowledge and practice. What we want to point out is that this dissemination from the standpoint of computers' usage, brought with it many implications. We feel there is not an adequate reflection on some of these implications. The tool is not innocent.

Evidently, this is an answer to the social request that we mentioned above. New users seem to be much more interested in *making music* than in *playing an instrument.* The long time needed to develop fine motor faculties and muscular skills for playing an instrument, seems absolutely inefficient to the fascination of multimedia production and to the eagerness for music manipulation without effort.

This trend has also influenced the use of computers in children's music education. Many of these experiences have reached encouraging results, but others arise doubts about the pedagogic ideology that inspire them. The need of using computers seems to be --in some of these cases-- a compulsive decision of educators or institutions. They feel that computers could be a possibility of having quick achievements, without an "inefficient" contact with musical instruments.

In the past, the realization of any musical structure implied the participation of performers. Thus, development of performing capabilities --even to virtuosity-- was a major goal of musical education.

The incontrovertible fact is that present technology has made the operation on complex musical structures without a human performer possible. Musical production can be today a purely mental fact, without the intervention of a human body. Precisely, sequencers' appeal among amateur musicians is related to the possibility of achieving musical results that largely exceed the performing skills of their authors.

Extremely, this implies the actual possibility of training composers that play no instrument, and that operate on musical structures exclusively through a computer keyboard. This situation -even when we can recognize precedents- is new in music history. Traditionally, studies on composition were carried out *a posteriori* of certain basics that included instrumental training, once students have had the experience of producing music themselves, and after having interacted with other performers.

Instruments with power supply

Electronic instruments have posed the need of developing new and specific playing techniques. By depressing the key of a synthesizer, complex processes start -for example, rhythmic phenomena--, that are only controlled in the instant of triggering them. Actions within the sound take place, without the need of any action of the performer to keep them going. This is a new fact in music history: a pianist, flutist or violinist always needed to *continue playing* if they wanted to prolong sound actions. This requirement of motor faculty of the performer was necessary, in most of the cases, to supply energy to the instrument, for sound production. At the same time, it gave the player a precise and empathic control over sound processes that were being generated.

None of this is necessary in electronic instruments. Their power supply for sound generation has a source different from their players. Complex sounds, of the kind that are common in the electroacoustic exploration, are musical processes of their own. Their development is independent of the player, because it has been programed in the very act of composing them.

We will be told that MIDI continuous controllers can give the player the control that we are lacking. Nevertheless, many of the player's physical actions over these devices have a distant relationship with the sound processes that they control. A modulation wheel, for instance, doesn't seem to be the best interface to control the amplitude or speed of a tremolo, if we compare it with the actions needed to do it in an acoustic instrument. Many times, the action modes of this continuous controllers lack an organic correlation with the sound effect that they produce.

Nonetheless, it is undeniable that synthesizers **are** musical instruments. The situation described above could be stated as an interface problem, a case of remoteness between sound processes and player actions, in a way similar to the case of the pipe organ among acoustic instruments. In any case, we can deal with the problem in terms of adaptation of the interface. We can see the growing interest towards research on control devices --even virtual instruments-- that could use the totality of human gestures (Mulder, 1994). Physical modeling synthesizers also poses an interesting approach to a more direct and organic relationship between the performers' bodies and the sounds they produce.

But what happens when there is not even a synthesizer under consideration? When the electroacoustic composer or the student thinks of the computer as his instrument --as **the** instrument--? Is it possible for those who do not know any other way of interacting with sound than a computer keyboard or a mouse, to develop music ideation processes?

The importance of the body

Psychoanalysis have stressed the importance of the connection with sensitive world as fundamental to psychological development. Consciousness cannot develop if there is no connection with the external world. André Lapierre also pointed the close relation between the psychical and the motor faculty developments. "Only through experience... with an active participation of motor faculty, is how the fundamental structures of abstract thought can be produced". (Lapierre- Acouturier, 1977).

Music is related to body from the beginning. "Rhythm in the body is very primary, and comprises our earliest musical selves. Between four and eight weeks' development, rhythmic contractions along the spine of the human embryo begin propelling it around the mother's womb... Our oldest neurophysiology resides in our lower back, and it is musical. Rhythm is corporeal - it is bodily." (Spaulding, 1994)

Nattiez says that a minimal condition for music is sound (Nattiez, 1987). When there is no connection with the world of sound --even as a representation of past experiences-- composition is no longer an operation with musical structures, and becomes a part of mathematics. Such a case would show the aim of constructing a world of ideas, whose internal coherence do not need to be validated with anything else.

In the case of a composer trained exclusively through a screen, there is a loss of something essential, a psychical impoverishment due of the lack of experience.

Division of work

The origin of the trend that we are describing in this paper could be tracked in the past. Let's propose the following framework.

Applying Molino's ideas of semiological tripartition (Nattiez, 1987) to musical creation, we can see that any given symbolic form can be studied in connection to three different dimensions.

a) The *poietic dimension*, which refers to the process of creation of the symbolic form. This process supposes techniques, rules and strategies of production, and it is describable and can be reconstructed.

b) The *esthesic dimension*, that implies the construction of meaning in the course of an active perceptual process. Suppose perceptive strategies. "The enjoyment, the contemplation or the reading of a work, the musical performance, as well as the scientific and analytic approaches to music are *de facto* placed in the esthesic field" (Nattiez,1987).

c) The *trace* (also named "neutral level" or "material level") refers to the physical and material aspects, accessible to senses, that embody the symbolic form. It is the work in its material level, its immanent configuration. The poietic process "...cannot immediately be read within its lineaments, since the esthesic process (if it is in part determined by the trace) is heavily dependent upon the lived experience of the 'receiver'." (Nattiez, 1987).

Western classical conception of the generation and functioning of a musical work considers three functions: the composer, who generates the work, the performer, who actualize it, and the listener, who decodes it. A simplistic approach --as shown in figure 1-- could assign the framework of tripartition to this three categories, like:

1) The composer originates the poietic processes to configure the symbolic form.
2) The performer manipulates the trace generated by the composer.
3) The listener puts esthesical processes in action, to read and comprehend the symbolic form.

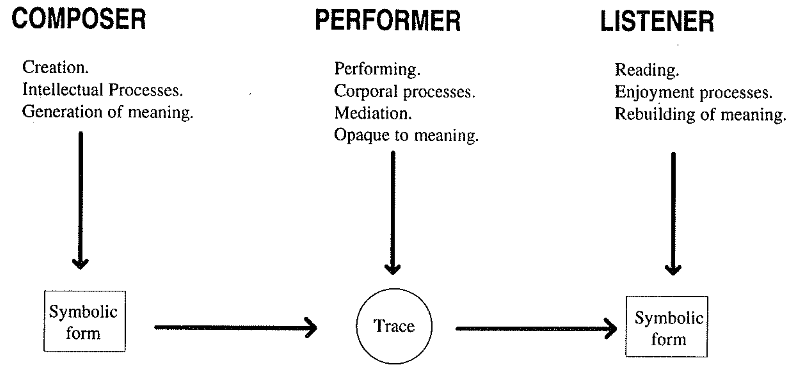


Fig. 1

Nevertheless, reality doesn't lend itself easily to this kind of simplifications. Although the functions of the composer, the performer and the listener are always present, it does not mean that these roles are necessary taken by different people. Let's take the example of a jazz improviser, that could be thought of as an almost "pure" performer, generating the work as he plays.

Without trying a morphology of music history or of the different aesthetics, we can see that the relationship among these roles varies according to musical periods and styles. To limit our analysis to the West, we see that historically, the roles of the composer and of the performer were intimately linked. In Bach's times, nobody could have imagined a composer who did not play one or several instruments, nor a good performer without compositional skills: music ideation was strongly related to its practice.

Other particular moments favored the separation between roles. For example, the medieval distinction between *musica theorica* --that was taught at universities along with mathematics--, and *musica practica* --made by singers and minstrles, generally of low social level and without a higher education--.

Closer to us, other ages have witnessed this separation. The situation of the baroque composer-performer begins to change with the development of the symphonic orchestra. In the romantic conception, the role assigned to the orchestra player was of a mere translator of the composer's intentions.

Berlioz, for example, imagined the orchestra as a big instrument. "The players of all kinds that constitute it, seem to be its strings, pipes, drums... machines become intelligent, but submitted to the action of an immense keyboard played by the conductor, under the direction of the composer." (Berlioz).

In symphonic music of late XIX and early XX centuries, the conductor, not the instrumentalist, was the responsible of musical decisions. Players lacked formation for the global comprehension of compositional processes. They only contributed doing an specific task, like workers in an assembly line, under the supervision of a foreman/conductor. We can see this phenomenon comparing well known treatises used for the formation of a professional flutist. In XVIII century, Quantz gave great importance, judging by the amount of chapters and pages, to the poietic aspects: handling of compositional structures, embellishment, aesthetic valuing criteria, etc. (Quantz, 1966). On the other hand, the Tafanel treatise --used till our days-- stresses the corporal skills, like scales, arpeggii, etc., without a need of comprehension of compositional issues (Tafanel, 1958) .

We find characteristics of this kind of practices in some of the studio recording techniques, when the instruments are recorded separately, in an artificial acoustic environment, without a real interaction among players.

This ideology of separation of tasks has led to a depreciation and underestimation of the role of the performer. Composers often look at the player as a mere executor of instructions, a necessary evil to translate --and many times to betray-- their intentions. Total serialism happily welcomed the possibility of electronic music, regarded as an overcoming of the performer. The wish of "total control" of the whole sound phenomenon, makes the player be a pain in the neck because of the degree of deviation produced by the act of performing. The composer, by means of the tape recording, could finally shape his work once and for all.

This attitude of devaluation and ignorance of the performer's role by the composers is in the background of the trend that is the subject of this paper. Computer appears as an efficient and well-disposed slave, --much more than humans--, that liberates composers from the painful jobs related to music production, allowing them to focus in the "intellectual" aspects of their job.

The role of the performer

Nevertheless, we know that the process is much more complex. The performer also assigns meaning, impoverishes or enriches the immanent structures, puts the piece in context again. That is, he operates within the poietic, and thus he has responsibilities in creation.

For example, research on early music requires an active role of the performer, where esthesic processes intensify to the extent that they incorporate elements of the poietic field. A language is recreated, from and for another cultural reality, enriching the original symbolic form. "...every rediscovery is also an increase... ...my reading... does not base itself only in the codes and the ideologies that have been recovered..., but also in codes and ideological perspectives specific to our days (enrichment codes), that allow us to insert an antiquarian's object into another context, enjoy it because of what it already meant, but also use it for the connotations that we could attribute to it with our present lexicon. We deal with a succession of surprises, of adventures, when discovering in a form their original contexts and when creating new ones." (Eco, 1978).

In this case we see that performers make important decisions related to the configuration of the work. Dealing with context, also, they devoted themselves to communicational processes. Of course, not every re-interpretation implies an enrichment. Let's remember the polemic on the coloring of classical films, on pretext of adapting them to a contemporary audience. That films were conceived in black and white, and this is an essential characteristic of the work. In this case, interpretation impoverishes the symbolic form.

In the case of the performers of early music, more a piece is far from present cultural patterns, less information is received from the "neutral level" or "trace". The performers should apply readings of ever increasing complexity, projecting on the "trace" all the information brought by historical research, and taking care of not operating exclusively from their own codes.

This analysis reveals the fallacious of the conception of the performer as a machine. Related to communicational issues, we should consider performer's gestures as part of communicational processes, as empathic gestures. The body of the artist thinks and feels, and makes its production something specific. "The sense of touch and especially the propioceptive sense, informing the player of the muscular tensions and of the positions and movements of the joints, colour the sequence of sounds events with significations that emanate from the concurrent sequences of playing actions. The essential aspect in the present context is the dual potential of these movements to support and promote momentary musical meaning and to give a rich and complex inner representation of musical processes, thus contributing to the identity of the music work... " (Edlung, 1994).

Nattiez also says that "... there is an entire series of nonsonorous phenomena that are quite rightly considered musical, by musicians themselves", and the he quotes a pianist´s assessment: "certain pianists have the impression that they give 'depth' to a chord by allowing the fingers to slide toward the interior of the piano after they have depressed the keys". Nattiez relates that with the poietic side, and continuing saying: "this... indicates that a kinesthesic and tactile sensation can intervene in the interpretants that the performer associates with the music produced". Alfred Brendel, quoted by Nattiez, says that "the sound of sustained notes on the piano can be modified... with the help of certain movements which make the pianist's *conception* of *cantabile* actually visible to the audience" Nattiez concludes that "separating the musical from the visual and the kinesthetic is indeed difficult"(Nattiez, 1987).

The idea of Varèse, of a machine that could faithfully transmit to the listener the music written in a score, supposes that in the score should be represented all the richness brought by the performer, without which the work would lack its interest to the audience.

We must remember that in the case of aesthetics like electroacoustic music, when there is no apparent performer, it is the composer who must assume this role.

And here we come back to our subject. How could it be possible to represent something that one has no experience of? Can a composer formed without any instrumental practice --whatever it is-- imagine processes on which there is no register in his psychological apparatus?

Music for no-body

Nattiez poses the problem of musical meaning by means of the following definition: "An object of any kind takes on meaning for an individual apprehending that object, as soon as that individual places the object in relation to areas of his lived experience --that is, in relation to a collection of other objects that belong to his or her experience of the world." (Nattiez,1987). The meaning is thus "the constellation of interpretants drawn from the lived experience of the sign's user, --the 'producer' or 'receiver'--, in a given situation". "Meaning exists when an object is situated in relation to a horizon." (Nattiez,1987).

It is interesting to point out two implications of this definition, regarding musical work. First, when we relate the problem of signification with a subject, that send us back to the communication processes mentioned. Second, when we talk about "lived experience", "experience of the world" or "horizon" we are referring not only to intellectual experience but also to sensory, kinesthesic and propioceptive experience. Intention of communication and sensitive experience are the two elements that we stress as basics for the generation of a musical work.

These considerations became particularly important at the present stage of computers' applications to composition. As Gareth Loy says: "What has become of composing where formal practices are used is simply the relocation of the compositional decision-making process to a higher position. (...) Some low-level elements of the compositional decision-making process may be taken over by an automatic process, but the composer must still both choose the process and accept the results." (Loy,1989).

Nevertheless, we have gone through the repeated experience of listening certain kind of works, algorithmically generated, whose authors seems to ignore that they should have an audience, and do not give an adequate flow of information to keep the attention of the listener. This idea, that could be considered a subjective impression, is confirmed when listening what the authors of such pieces say about them. These kind of works seems to have been composed to illustrate a thesis, as a theorem demonstration, as a "correct" solution of an equation. They ignore the communication process.

Independently of the interest that some results of musical exploring through algorithmic composition have, we believe that leaving the control of certain kind of musical processes to machines implies an abdication attitude of the composer. Particularly we refer to the control of temporal processes, by means of machines that do not perceive time. As an example, for a computer that could generate fugues in Bach's style, it would be the same thing to play them at M.M.= 72 or at M.M.= 1/1.000.000. The structures are "correct" in both cases, but in the first one we humans could recognize musical processes, and in the second they are not perceptible. The fact of configuring a symbolic form does not guarantee that it will be inserted into a communication process.

Communication is a particular case within symbolic processes (Nattier, 1987). To make communication possible, other requirements involving the "horizons" of performer and listener should be accomplished, besides the composer's universe. Figure 2 intends to represent the complex processes involved.

For this reason, it is essential that the composer should be the first performer of his work, and its first listener. That is why we insist in the importance of forming composers on the basis of experiences that involve body gestures. The ability of mental representation of real-time processes must be constructed. It is not replaced with the use of a machine that solves the problems "correctly".

Quod erat demonstrandum?

At this point, we would like our objective to be clearly understood, and not to be interpreted as a reactionary position. We are not posing an opposition --absurd at this stage-- between man and machine. We are not questioning the use of computers as musical instruments, not even consider heretical the developing of AI systems that allow machines to compose like humans.

The capital problem to pose is not *"should we teach computers to compose?"*, but *"how can we teach **composers** to compose?"* Computers are a powerful medium for expanding the cognitive and expressive universe of already skilled composers, and an invaluable tool for their education. What we are posing is the nonsense of their use as an *exclusive* tool for music education, and even the possibility of self-defeating effects.

We would be told that nobody is supporting such a position. Nevertheless, the facts we mentioned at the beginning of this paper are pointing out that social processes related to music practice and teaching, are generating situations towards this direction.

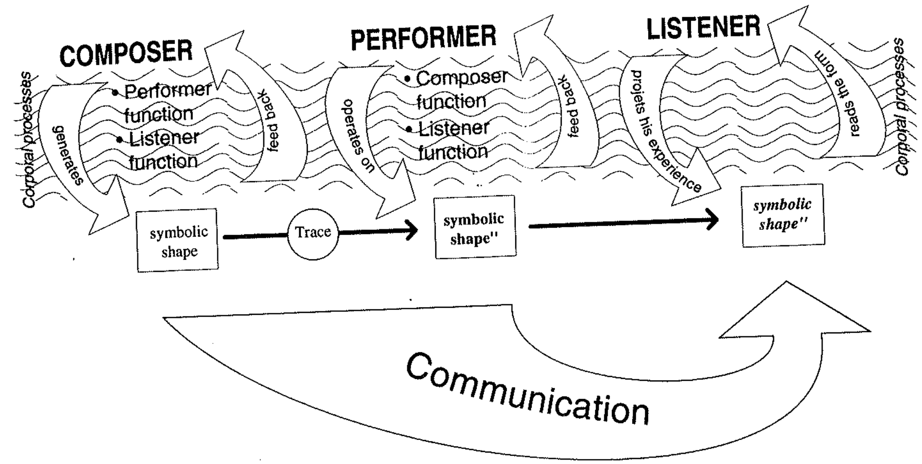


Fig. 2

The considerations above do not imply that the new ways of making music can't achieve artistic goals. Anyway, we do believe that these "users", trained without a connection with the body, will suffer an impoverishment of their musical thinking and an unawareness of certain processes of artistic communication. We do believe that composers' education should generate in them the capability to deal with a multiplicity of aesthetics, instead of limiting their expressive skills due to the lack of an integral musical experience.

Our emphasis on instrumental practice does not imply that we consider it the only way of developing the musical thought. This could also be achieved through singing, or through a deep, conscious and active listening of musical processes, or by means of the practice with virtual instruments sensitive to body gestures. Nor should this emphasis be understood as a preference for a given tool or aesthetics. We are not saying that somebody will necessary be a better composer because he or she plays Mozart's sonatas on the piano.

What we do believe is that the composers' education --or children's access to music-- should be carried out through activities that develop a connection with their bodies and that let them go out to meet the affective world of their peers. The authors are fully for a conception of music as a communication process addressed to others, rather than considering it an exclusively intellectual construction.

Artists interact with their works: this is what makes artistic creation one of the most enriching of human experience. A superficial scope of technological media considers them as a possibility of *making without effort*, emphasizing the generation of artistic product from an efficiency logic. However, artists know that it is not a matter of only achieving the product: even or more important than that is to go through the processes that lead to it. We do believe that --as in old fairy tales-- the experiences that take place alongside the way are what allow the protagonist, at the beginning clumsy and inexperienced, to find at the end the truth.

Conclusions

The formation of "composers/users", by means of their exclusive training on computers, is an actual possibility. These new operating ways imply a change in the paradigm of what was called composition so far.

New technological media should not be adopted noncritically for educational uses. When making use of them, teachers and institutions devoted to music education should privilege the developments that favor a connection of new musicians with their bodies.

Music ideation processes need a connection with sensitive reality for their development. Instrumental practice is one of the most efficient ways of achieving this connection, and at the same time of introducing the student to artistic communication.

Acknowledgments

The authors want to thank to Lic. Estela Tarrab and María Silvia Mazzanti for their advice on psychological issues, and Prof. Marina Mayor for her help with the English translation.

References

Berlioz, H. (without indication of date). *Traité d'Instrumentation et d'Orchestration.* Paris: Henry Lemoine.

Dodge, Charles & Jerse, Thomas (1985). *Computer Music - Synthesis, Composition and Performance.* New York: Schirmer.

Eco, Umberto (1978). *La estructura ausente. Introducción a la semiótica.* Barcelona: Lumen.

Edlund, Bengt (1994). Musical Interpretation and the Ontology of the Music Work. *Musical connection: tradition and change. Proceedings of the 21st World Conference of the International Society for Music Education.* Tampa, Florida: ISME.

Iazzetta, Fernando (1994). Um Novo Músico Chamado 'Usuário'. *Annais de I Simpósio Brasileiro de Computação e Música.* Caxambu: Universidade Federal de Minas Gerais.

Lapierre, André & Acouturier, Bernard (1977). *Los Contrastes - y el descubrimiento de las nociones fundamentales.* Barcelona: Editorial Científico-Médica.

Loy, Gareth (1989) Composing with computers - a Survey of Some Compositional Formalisms and Music Programming Languages. *Current Directions in Computer Music Research.* Cambridge: Massachusetts Institute of Technology.

Moore, F. Richard (1990). *Elements of Computer Music.* Englewood Cliffs, NJ: Prentice-Hall.

Mulder, Axel (1994). Virtual Musical Instruments: Accessing the Sound Synthesis Universe as a Performer. *Annais de I Simpósio Brasileiro de Computação e Música.* Caxambu: Universidade Federal de Minas Gerais.

Nattiez, Jean Jaques (1987). *Musicologie Général et Semiologie.* Mayenne: Christian Bourgois Editeur.

Quantz, Johann Joachim (1966). *On Playing the Flute.* London: Faber and Faber.

Spaulding, Crispin (1994) Performance and Physiology: New liaisons in Educating the Professional Musician. *Proceedings of the Seminar Meeting of the ISME Comission on the Education of the Professional Musician.* Gainesville: University of Florida.

Swingler, Tim (1994). Choreografy as Composition - Movement into Music: Using Soundbeam as a New Key to Creative Eloquence. *Musical Connection: Tradition and Change. Proceedings of the 21st World Conference of the International Society for Music Education.* Tampa, Florida: ISME.

Taffanel, P. and Gaubert, Ph. (1958). *Méthode Complète.* Paris: Alphonse Leduc.

Theoretical outline of a hybrid musical system

DAMIÁN KELLER, CONRADO SILVA

Departamento de Música
Universidade de Brasília
Brasília - DF - Brazil
E-mail: {d3121966, conrado}@guarany.cpd.unb.br

Abstract

Current approaches to musical research place a strong emphasis either on sound production - synthesis, algorithmic composition - or on cognitive models - information processing. As a way to start ploughing a common ground we outline a framework where musical structure is determined by the interaction of three inter-dependent aspects: sound system (physical), perceptual system (psychological) and social system. These are interlinked dynamically and their interaction provides a transformation of the state of each system. Information is defined by the range and rate of transformation of the elements in the physical system, being periodicity and entropy the two opposing factors that act on it. The probability that a specific information will be processed and transmitted by the perceptual system defines its state.We view each musical syntax as a sub-set of a greater system that includes the different forms of western music, defining observation in three levels: sound, syntax and morphology. The final output of the model - which depends on the probability that the combination of the present states will allow the processing of a given information - should provide a way to observe and compare the reaction of systems to the content of various musical phenomena.

Introduction

The approach taken in this paper follows a line that is represented by sporadic works that try to gather diversified fields in a common musical interdisciplinary ground (Birdlack, 1992; Boon et al, 1990; Eco, 1968; Georgescu & Georgescu, 1990; Moles,1969; Malt, 1994). Although the need to define and formalize such a perspective has been pointed out by many researchers the difficulties encountered outnumber the useful tools available from each field (Bregman, 1990; Ferneda et al, 1994; Harnald, 1987; Parncutt, 1989; Sloboda,1985). So two main reasons can be mentioned regarding the slow process towards unification: a tradition in the scientific and artistic community to separate and fragment areas of knowledge and the danger to transpose mechanically concepts developed in one field to another (Roederer, 1975; Vriend, 1981).

Our work addresses the problem of constructing a framework to deal with musical objects focusing on theoretical aspects and excluding actual implementation issues. This is based on the fact that the latter have already been treated in a large volume of publications and there is software available to observe the behavior of some parts of the proposed system - i.e. neural networks for pitch processing and perceptual tasks, simulation of a system behavior, mapping for the phase space, calculation of entropy, representation of signals in the time-frequency domain, correlation - (Arfib, 1991; Boon et al, 1990; LabVIEW, 1994; Matlab, 1994 ; Ramirez, 1985; Todd & Loy, 1991). Thus we discuss the most salient features and their theoretical implications in an effort to draw a global - but not complete - picture and to underline conceptual weakness of parts of a model whose details could be completed through future research.

Although analogies drawn from related fields have proven to be very useful for the understanding of music they cannot be taken as a ready-to-use tool (Roederer,1975). Therefore when the concept of dynamical system is applied to music the characteristics of the system should be inferred from the behavior of real musical systems before we can establish any conclusion about the behavior of a musical system or any prediction on the results of a specific transformation. That is why we do not adhere to a one-sided physical or psychological description. Following the same line of thought, an early mathematical formalization may incur in simplifications that don't fit the complexity and unpredictability of musical phenomena (Knopoff & Hutchingson, 1981; Georgescu & Georgescu, 1990).We believe that it is a step that should eventually be taken when many of the posed (and also the ones not considered!) conceptual problems have been discussed and clarified.

At that stage, confirmation bias - the fact that the observer actively searches for evidence that supports his beliefs - and the complementarity principle - which states that observation depends on choices made by the observer, should have been introduced (Mitroff, 1974; Werner & Wells, 1990; Wolf, 1981). The whole musical process could be divided in three interlinked subsystems: sound system, perceptual system and social system. This subdivision comes out from the trends observed in musical research, where this areas can be clearly distinguished in relation to their methodology and object of study - i.e. signal processing, psychology of perception, developmental psychology (Deutsch, 1982; Flanagan, 1972; Hargreaves, 1948; Kubovy & Pomerantz, 1981; Oppenheim & Schafer, 1975; Pierce, 1983; Prince, 1972).

The sound system is defined as a time series - variations of a chosen variable as a function of time - that is, the unfolding of a sequence of acoustical signals (Boon et al, 1990). Information provided by the sound system is processed by a perceptual system whose dynamics are described by its states - or representations - which are modified by processes - or operations. Global constrains - established by the social system - set the range of possible perceptual states and thus the probability that the whole musical process will take place in a specified way.

Representation

From a philosophical point of view two main approaches can be traced regarding the process of knowledge acquisition. They are conceptually opposed but complementary in their application: elementism, that proposes an interaction of units or elements which combine in complexes to form perceptual objects, ideas, etc.; symbolism, that establishes direct acquisition of complex structures - symbols - which are related hierarchically to form new concepts or perceptual objects (Lischka, 1991; Marsden & Polple, 1989; Massaro & Cowan, 1993; Rock, 1984; Wertheimer, 1974).

The existing wide spectrum of proposals on representation mechanisms range from logic-rules to complete absence of representation (Lischka, 1991; Pylyshyn,1984; Suppes et al, 1994). No conclusive general results have been attained on the mental mechanisms activated by auditory stimuli, but some factors should be considered when treating specifically musical phenomena.We'll discuss some issues related to three perspectives: analytical, computational and psychological.

Psychological perspective

The traditional approaches in psychology of perception placed a strong emphasis on detection of thresholds (Zwicker et al, 1957; Luce & Clark, 1967; Roederer, 1974). Since the mid-sixties with the apparition of Signal Detection Theory a shift in paradigm called attention to the fact that the transmission and processing of information could be better understood by probability weights (McNicol, 1972; Garner, 1974; Green, 1988; Green, 1972; Green & Berg, 1991; Green & Swets, 1966; Green et al, 1984; Green et al; 1985; Lufti; 1992). Concepts as just-noticeable-difference were left aside (it is strange that in the music field this is still cited and used).

Springing from the pioneer works in Information Theory (Shannon, 1948; Coons & Kraehenbuel, 1958; Knopoff & Hutchingson, 1981; Knopoff & Hutchingson, 1983), but evolving to its own conceptual field, Information Processing (IP) has become one of the most influential paradigms in psychology (Massaro & Cowan, 1993; Suppes et al, 1994). The most general properties of IP are: *Informational description*: all environmental and mental processes can be described in terms of type and amount of information. *Recursive decomposition*: each stage of processing can be broken down into substages. *Flow continuity*: information is transmitted forward in time. *Flow dynamics*: each stage or operation takes time - there are no instantaneous mental processes. *Physical embodiment*: information processing occurs in a physical system. At last, coincidentally with the approach of this paper, IP establishes that information is embedded in states of a system - or representations -, and processes - or operations - are used to transform these representations.

An important distinction formulated by IP is that data should not be treated as information until it is processed by the receiver. From this point of view knowledge should be understood as information actually available by the individual and not the raw data present in the environment (Massaro & Cowan, 1993). A separation between physical (previous to mental processing) and psychological systems seems the simplest way to deal with this issue.

Rule-based models

Several limitations have been pointed out regarding the approaches based on linguistic analogy (Cook,1994; Leman, 1989; Todd & Loy, 1991; Remez et al, 1994). Taking in count the characteristics of musical material, continuous representations which allow for decisions based on probability weights approximate the uncertainty intrinsic of musical stimuli. This is excluded from rule-based approaches where decisions are taken in relation to pre-established constrains. Lischka (1991) points to the fact that continuous processes can hardly be explained by discrete categories and that combinatory rules and various decision stages do not meet with the speed requirements of fast complex processes that occur in perception.

Hierarchical organization of layers does not consider interaction among interdependent factors such as macro and micro-structure in music (see Lerdahl, 1987; Lerdhal & Jackendoff, 1983; Bregman 1990; Deutsch, 1982b; Dowling & Harwood, 1986). The alternative approach of setting a system, where modification of any parameter at any level would cause a transformation of the system as a whole, seems to correspond better to real musical phenomena. A mental experiment would be to play a sonata by Mozart with a piano timbre and afterwards with white noise filtered to reproduce relative pitch variations - as in Hesse's experiment (1982). Where would harmonic functions, hierarchical structures, tonal closure, dissonance and consonance go?. However its macro-structure would be kept exactly the same.

Parallel distributed processing models

Among the IP currents an approach that grew from a tentative to simulate the physiological structure of the brain and expanded to various applications is the parallel distributed processing or connectionist model. As with any present model several problems can be pointed out, nevertheless it is an expanding field that has implemented some useful tools (Rumelhart & McClelland, 1986; Todd & Loy, 1991).

Marsden & Pople (1989) describe it as consisting of a network of processing units, each producing a quantitative output which is a simple function of its input, again quantitative. Except where this output is the overall output of the model, it is routed to the input of other processing units. The arrangement and strength of the connections is contrived in such a way as to produce the appropriate overall output from the model, given appropriate input.

The drawback - and at the same time the advantage - of neural networks applied to auditory perceptual tasks is the possibility to predict an outcome - given the necessary conditions - without a pre-established model of auditory processing. The lack of explicit stages, not allowing time control of independent variables, the superpower of the model which permits different net structures to have the same outcome so alternative models cannot be verified and the difficulty to introduce distal causation (or correlations in events separated in time) are some of the weaknesses that have been pointed out (Laske,1991; Massaro & Cowan, 1993). The pretension to validate the model on neurological bases suffers from some shortcomings:

1. Leman (1989) observes the lack of correspondence between reaction times in neuronal systems and computer simulations. The basic computing elements of the brain operate in the range of a few milliseconds - a million times slower than current electronic devices. The reaction times of complex behaviors - such as hearing - are carried out in a few hundred time steps. He concludes that brain processes should be carried out by a massive parallel network of connected units. However this argument does not take in count the differences in reaction time that has been observed for different types of tasks (Pachella, 1974; Keller & Tróccoli, 1995; Dai & Green, 1993; Bernstein & Green, 1987b; Perfetti & Bell, 1983; Pitt & Crowder, 1992; Semal & Demany, 1991).

2. Wertheimer (1974) - supporting direct acquisition of complex structures - describes an experiment where micro-electrodes implanted in visual nerves of animals showed a group of nerve cells reacting to complex stimuli - the angle and direction of the movement of a line. On the other hand when specific parameters - as luminosity and wave length - were used no answer was elicited. Thus the scheme of input units that react to specific parameters can not be generalized.

Computational perspective

When we set to the task of analysing, transforming or producing musical information the first problem encountered is how to translate that information into a code that reflects the physical and perceptual characteristics of the signal. A digitized waveform only represents the variations of amplitude and frequency within a preassigned range and level of quantization (Garnett, 1991).

From a computational perspective information can be represented as plain data or as a knowledge system incorporating all characteristics that define that system, or an intermediate combination of this two possibilities. The issue here is the tradeoff between efficiency and easy of use. Following Garnett (1991) we would be talking about computational dynamics (the underlying operational flow of data and instructions) and computational semantics (the user interface). How this two levels relate has been minutely discussed in Representation of musical signals (DePoli et al, 1991). Three types of music signals have been proposed. Acoustic: time-pressure waveforms, analytic: such as FFT or autocorrelations, and parametric: derived from models such as linear predictive coding coefficients, synthesis parameters (Arfib, 1991; Flanagan, 1972; Roads, 1991).

The implementation of signals as a sequence of values indexed by positive integers - as in general signal processing - has proven to be the simplest and most efficient standard that can be adopted for their physical representation. The analytic transformation to a time-frequency representation can easily be done by means of Short Time Fourier Transform or Wavelet Transform (Arfib, 1991; Kronland-Martinet & Grossmann, 1991). Once in the frequency domain correlation can be applied to rate the signals in a predictability scale (Ramirez, 1985). High correlation would match most predictable states and low correlation would correspond to entropic states - as in random noise (Boon et al, 1990).

The advantage of using STFT and WT is that they allow for energy conservation, so they are reversible processes (Kronland-Martinet & Grossmann,1991). This should be one of the important factors in deciding what representation to apply in a system where data is repeatedly transformed and handled through various stages. This simplifies controlling and testing outcomes of each stage for alternative models - e.g. a signal is transformed, modified and transformed back to its first representation to compare original and modified signal -.The approach adopted in the present paper proposes to deal with different types of data representation depending on the transformation being done. Thus data in the perceptual system should incorporate selective knowledge - an irreversible process - contrasting with the reversible processes of the sound system.

Musical analysis

The perspective that has the loosest level of formalization from the ones considered is musical analysis, including in this category analysis itself and related tasks as composing, performance, analysis-synthesis (Bent, 1987; Boulez, 1992; Karkoschka, 1972; Karkoschka,1987; Menezes, 1987; Risset, 1991; Risset & Wessel, 1982; Schaeffer, 1993; Schoenberg, 1974; Wessel, 1979) - which imply a complex knowledge of musical structure that is in part intuitive.

Many criticisms have been pronounced in relation to this lack of formalization questioning the utility of this approach and trying to find a place where its production could be useful to research (DeLio, 1980; Demster & Brown, 1990; Kunst,1987; Polansky & Bassein, 1992; Sloboda, 1985; Vriend, 1981). Cook's proposal of using music theory as a tool to open new musical frontiers appears to fit in the present panorama of research. We would like to add that where models can not be tested by other means, as a last resort we have our auditory system, plenty of biases and social conditionings but the only way to fully apprehend musical phenomena.

A hybrid musical system

Sound System

When music is considered as a time series the transformation of a variable in relation to time can be observed. This places a strong emphasis on the process itself but blurs our view of the structural regularities. That is, we can easily find correlations in events that are near in the time axis but distant events are difficult to compare. To facilitate observation, the scale - or kernel - of the representation is modified (Mozer, 1991). This allows us to look for regularities in the sound, syntax or morphology of a musical signal. If a musical work is understood as a system in equilibrium, its structure would be a time-invariant representation embodying all states. Their time history represents the dynamic process of the system. If we plot this process in a three dimensional graph, the evolution of the system is shown as a trajectory in the phase-space sustained by the x, y, z axes. This space equals the range of the system (Boon, 1990).

Nevertheless a conceptual differentiation is necessary when a shift of perspective occurs: the same phenomenon can be described as static (time-invariant) or dynamic (time-variant). A simple example would be a succession of five pitches which are played forwards and backwards. A static representation, as in Set Theory, would define a temporal space that is filled with five elements (Bent, 1987; Forte, 1970). A dynamic approach would represent the process used to order these elements.

The stability of the musical system is proportional to its entropy. Almost completely entropic systems are the most stable ones, so strong processes - highly energetical - are needed to disrupt their equilibrium - as it happens in stochastic music or random noise where foreign events can be introduced without great modification of the system's behavior. On the other hand a system with high periodicity will suffer a strong impact from a small perturbation - such is the case in Morton Feldman's music or in short repetitive patterns where any minor modification is easily detected. We can not draw final conclusions until experimental tests are run, but Georgescu & Georgescu's (1990) statement that "steady-state (structurally-stable) music is represented by works where stratified, deterministic, causal, memory-endowed dependances interact [...] as in a Mozart's sonata" does not seem very likely when compared to any Michael Jackson's hits.

Perceptual system

When the sound system is observed, an arbitrary portion of information is extracted so that there is a loss in the transmission between the actual physical system and its perceptual representation (Berstein & Green, 1987; Dowling, 1994; Durlach et al, 1986; Mason et al, 1984; Nielzen & Olsson, 1989; Yost & Watson, 1986). This loss can be represented by a dynamic information filter whose characteristic are defined by the state of the perceptual system.

Fusion or parsing can be thought as a high level process - that acts on other processes -. If we consider how psychoacoustical models process spectral information, clues such as harmonicity, synchronicity, spacing of components, modulation will define if the sound is perceived as a unit (Feth, 1974; Green, 1988; McAdams, 1982; Richards et al, 1989; Sano & Jenkins, 1991; Slawson, 1985; Terhardt et al, 1982; Vos & Rasch, 1982). Regarding musical syntax, pitch-height, pitch-class and interval representation - as in Bharucha's model (1991) - depend on synthetic or analytic hearing, in other words whether a fusion process is activated or not (Howe et al, 1993; Melara & Marks, 1990a; Melara & Marks, 1990b; Melara & Marks, 1990c; Singh,1987).

At the time data reaches the information filter, it is parsed among available channels (Grossman, 1972). Depending on the range of the signal and its rate of transformation, the bandwidth and resolution of each channel is set. If their capacity is exceeded the fusion process is activated, then the number of channels is reduced and so is their resolution. The settings of the filter are stored in a buffer which accounts for the memory of the state of the perceptual system.

Social System

From a musical perspective the constrains imposed by the social system are mirrored in what a listener has as his musical background: all musical stimuli that he receives and how they are stored and used by him in specific tasks, i.e. producing and listening to sound (Hargreaves,1948; Prince,1972; Siegel, 1981; Sloboda, 1985). Environmental constrains are always present in production of music. Even when implementing automatic generation of random sound the limits are set by the sample space: only an infinite sample is completely random.

A variety of listening experiments has shown the influence of cultural context on the perception of music (Deliege, 1989; Jones, 1987; Krumbansl, 1990; Parncutt, 1989; Wolpert,1990). Actually an important question in experimental research has been the problem of individualizing the variables that are acquired from interaction with the environment and the ones that are innate(Aiello,1994; Deutsch,1982; Krumhansl, 1990; Roederer, 1974). Although this differentiation is relevant, while enough data is not available the alternative approach of taking cultural constrains as given data may prove useful. The other aspect of the same problem is the interaction between environment and individual, thought as a feedback process where the action of each social component modifies the environment - as it is the case in musical production -. This has not yet been formalized either.Therefore we propose that the modifications infringed on the environment be incorporated as a feedback loop for each iteration among systems.

When analysing the behavior of a social system, the global features that characterize its dynamics are observed. Each specific event looses importance and the statistical distribution of a large number of events is focused. While in state of equilibrium the system favors the features that have the higher activation weights. When this features are outweighed by repeated exposure to contrasting stimuli, the equilibrium of the system is broken and a new state is reached.

Neural networks have the ability to learn patterns and features from a given set of musical examples (Todd & Loy, 1991). So the application of this knowledge is straight forward: feed the network with a relevant corpus of musical stimuli, extract the features that characterize them and use these features to set the limits of possible behaviors of the perceptual system.

Conclusion

We have only scratched the surface of the implications of a model that brings together physical, perceptual and social information to a musical framework. The division of the sound system in three levels - sound, syntax and morphology - provides a way to treat different types of information without fragmenting the musical structure. Entropy and periodicity used to define the stability of the sound system allow for predictions on its behavior. Fusion-parsing as a high level process acting on information selection simplifies the structure of perceptual mechanisms. Constrains defined by the social system contextualize the musical process in a specific environment. Further development of this line of research should introduce attractors and control parameters for

each system and quantification devices to arrive to a mathematical formulation that does not exclude uncertainty and fuzziness.

The purpose of this exposition was just to light the fire of a discussion that's far from ended, we hope that many ideas will come to stir this musical system's equilibrium. To give an overall view as to how this system could be put to work, we propose a rough-draft of ordered procedures, with the reminder that feedback and crossed connections are missing.

Procedure

<u>Sound System</u>
Establish a kernel [sound - syntax - morphology].
Sound.
Establish domain [time - frequency].
Apply transformation [correlation].
Find the range [minimum - maximum].
Plot a static representation [system].
Define sub-range. (In case of processes under perceptual threshold, phase - in periodic signals - could be a parameter used to define time range).
Plot a time-varying representation [states].
Find the process underlying the dynamics of the system.

Syntax.
Feed signal 1.
Establish domain [duration - intensity - pitch].
Apply correlation within.
Find range.
Plot static representation [system].
Define sub-range [perceptive unit].
Plot a time-varying representation [states].
Find the process underlying the dynamics of the system.

Feed signal 2.
Establish domain [duration - intensity - pitch].
Apply correlation within.
Find range.
Plot static representation [system].
Define sub-range [perceptive unit].
Plot a time-varying representation [states].
Find the process underlying the dynamics of the system.

Feed signals 1 and 2.
Apply correlation between.
Find range.
Plot static representation.
Define sub-range [perceptive unit].
Plot a time-varying representation [states].
Find the process underlying the dynamics of the system.

Morphology.
Establish domain [information distribution (range and rate of transformation)].
Apply correlation [sound: time - frequency][syntax: duration - pitch - intensity].
Compare morphologies.

<u>Perceptual system</u>
Establish variable [sound - syntax - morphology].
Observe Garner effect - interactions.
Input stage [temporal clues - spectral clues].
Transformational stage [fusion - parsing][fit pattern][map to representation] [dynamic information filter (buffer)].

Output stage [feedback][fuzzy decodification].
Compare input and output.

<u>Social System</u>
Input stage [dominating corpus of music].
Transformation stage [statistical weight to strong activations].
Output stage [relevant features][constrains to perceptual system].

References

Aiello, R. (Ed.) (1994). *Musical perceptions*. New York: Oxford University Press.

Arfib, D. (1991). Analysis, transformation, and resynthesis of musical sounds with the help of a time-frequency representation, *Representation of Musical Signals*. Cambridge, MA: MIT Press.

Bharucha, J.J. (1991). Pitch, harmony, and neural nets: a psychological perspective, *Music and Connectionism*. Cambridge, MA: MIT Press.

Bent, I. (1987). *Analysis*. New York: London.

Berg, B. G. & Green, D. M. (1990). Spectral weights in profile listening. *Journal of the Acoustical Society of America*, 88, 758-766.

Bernstein, L. R. & Green, D. M. (1987a). The profile analysis bandwidth. *Journal of the Acoustical Society of America*, 81, 1888-1895.

Bernstein, L. R. & Green, D. M. (1987b). Detection of simple and complex changes of spectral shape. *Journal of the Acoustical Society of America*, 82, 1587-1592.

Bidlack, R. (1992). Chaotic systems as simple (but complex) compositional algorithms. *Computer Music Journal*, 16 (3), 33-47.

Boon, J.P., Noullez, A. & Mommen, C. (1990). Complex dynamics and musical structure. *Interface*,19, 3-14.

Boulez, P. (1992). *Hacia una estética musical*. Caracas, Venezuela: Monte Ávila Editores.

Bregman, A. S. (1990). *Auditory Scene Analysis: the perceptual organization of sound*. Cambridge, MA: MIT Press.

Cook, N. (1994). Perception: a perspective from music theory, *Musical Perceptions*. New York: Oxford University Press.

Coons, E. & Kraehenbuehl, D. (1958). Information as a measure of structure in music. *Journal of Music Theory*, 2, 127-161.

Dai, H. & Green, D. M. (1993). Discrimination of spectral shape as a function of stimulus duration. *Journal of the Acoustical Society of America*, 93 (2), 957-965.

Dalhaus, C. (1975). Some models of unity in musical form. *Journal of Music Theory*,19.1, 2-30.

DePoli, G., Piccialli, A., Roads, C. (1991). *Representation of musical signals*. Cambridge, MA: MIT Press.

Deliège, I. (1989). A perceptual approach to contemporary musical form. *Contemporary Music Review* 4, 213-230.

DeLio, T. (1980). Iannis Xenakis' Nomos Alpha: the dialectics of structure and materials. *Journal of Music Theory* 24.1, 63-95.

Demster, D. & Brown, M. (1990). Evaluating musical analyses and theories: five perspectives. *Journal of Music Theory*, 34.2, 247-279.

Deutsch, D. (Ed.) (1982a). *The psychology of music*. New York: Academic Press.

Deutsch, D. (1982b). Organizational processes in music, *Music Mind and Brain*. New York: Plenum Press.

Dodge, C. & Jerse, T. A. (1985). *Computer Music*. New York: Schirmer Books.

Dowling, W. J. (1994). Melodic contour in hearing and remembering melodies, *Musical Perceptions*. New York: Oxford University Press.

Dowling, W. J. & Harwood, D. L. (1986). *Music Cognition*. Orlando, FL: Academic Press.

Durlach, N. I., Braida, L. D., and Ito, Y. (1986). Toward a model for discrimination of broadband signals. *Journal of the Acoustical Society of America* 80, 63-72.

Eco, U. (1968). *Obra Aberta*. São Paulo: Editora Perspectiva.

Ferneda, E., Silva, C.A.P. da, Teixeira, L.M., Menezes Silva, H. de, (1994). A system for aiding discovery in musical analysis, *Anais do I simpósio brasileiro de computação e música*. Belo Horizonte, MG: Maurício Loureiro.

Feth, L. L. (1974). Frequency discrimination of complex periodic tones. *Perception & Psychophysics*, 15, 375-378.

Flanagan, J. L. (1972). *Speech analysis, synthesis and perception*. New Jersey, Murray Hill: Bell Laboratories.

Forte, A. (1970). *The structure of atonal music*. New Haven, CT: Yale University Press.

Garner, W. R. (1974). *The processing of information and structure*. New York: Wiley.Garnett, G.E. (1991). Music, signals and representations: a survey, *Representation of Musical Signals*. Cambridge, MA: MIT Press.

Geary, J. M. (1980). Consonance and dissonance of pairs of inharmonic sounds. *Journal of the Acoustical Society of Americc*, 67, 1785-1789.

Georgescu, C. & Georgescu, M. (1990). A system approach to music. *Interface*,19, 15-52.

Green, D. M. & Berg, B. G. (1991). Spectral weights and profile bowl. *Quarterly Journal Experimental Psychology*, 43A, 449-458.

Green, D. M. & Swets, J. A. (1966). *Signal detection theory and psychophysics*. New York: Wiley.

Green, D. M. (1988). *Profile Analysis: Auditory Intensity Discrimination*. New York: Oxford University Press.

Green, D. M., Mason, C. R., & Kidd, G., Jr. (1984). Profile analysis: critical bands and duration. *Journal of the Acoustical Society of America*,75, 1163-1167.

Green, D. M. & Mason, C. R. (1985). Auditory profile analysis: frequency, phase and Weber's law. *Journal of the Acoustical Society of America*, 77, 1155-1161.

Green, D. M. (1992). The number of components in profile analysis tasks. *Journal of the Acoustical Society of America*, 91, 1616-1623.

Grey, J. M. & Gordon, J. W. (1978). Perceptual effects of the spectral modifications of musical timbres. *Journal of the Acoustical Society of America*, 63, 1493-1500.

Grey, J. M. (1977). Multidimensional perceptual scaling of musical timbres. *Journal of the Acoustical Society of America*, 61, 1270-1277.

Grey, J. M. (1978). Timbre discrimination in musical patterns. *Journal of the Acoustical Society of America*, 64, 467-472.

Grossman, E.E. (1972). Signal detection theory applications to a developmental analysis of auditory intensity selectivity, *Doctoral Thesis in Psychology*. Fayetteville, AR: University of Arkansas.

Hargreaves, D. (1948). *The developmental psychology of music*. Cambridge: Cambridge University Press.

Harnald, S. (Ed.) (1987). *Categorical perception: the groundwork of cognition*. Cambridge: Cambridge University Press.

Helmholtz, H. von (1885). *On the Sensations of Tone*. London: Longmans.

Hesse, H. P. (1982). The judgement of musical intervals, *Music Mind and Brain*. New York: Plenum Press.

Howe, M. L., Rabinowitz, F. M., Grant, M. J. (1993). On measuring (in)dependence of cognitive processes. *Psychological Review*, 100 (4), 737-747.

Iverson, P. & Krumhansl, C. L. (1991). Measuring similarity of musical timbres. *Journal of the Acoustical Society of America*, 89 (2), 1988.

Jones, M. R. (1987). Dynamic pattern structure in music: recent theory and research. *Perception & Psychophysics*, 41, 621-634.

Karkoschka, E. (1972). *Notations in new music*. London: Universal Edition.

Karkoschka, E. (1987). Analysis of new music. *Interface* 16, 13-21.

Keller, D. & Tróccoli, B.T. (1995). Percepção auditiva: aspectos da interação entre psicologia e música, *Anais do II congresso de ciências humanas e artes*. Uberlândia: Universidade Federal de Uberlândia.

Knopoff, L. & Hutchingson, W. (1981). Information theory for musical continua. *Journal of Music Theory* 25.1 17-43.

Knopoff, L. & Hutchingson, W. (1983). Entropy as a measure of style: the influence of sample length. *Journal of Music Theory* 27.1, 75-97.

Kronland-Martinet, R. & Grossmann, A. (1991). Application of time-frequency and time-scale methods (wavelet transforms) to the analysis, synthesis and transformation of natural sounds, *Representation of Musical Signals*. Cambridge, MA: MIT Press.

Krumhansl, C. L. (1990). *Cognitive foundations of musical pitch*. New York: Oxford University Press.

Kubovy, M., & Pomerantz, J. R. (Eds.) (1981). *Perceptual Organization*. New York, Hillsdale: Erlbaum.

Kunst, J. (1987). Remarks on analysis. *Interface*,16, 1-11.

LabVIEW (1994). *Graphical Programming for Instrumentation* . Austin, TX: National Instruments Corporation.

Laske, O. (1991). Letter: connectionist composition, *Music and connectionism*. Cambridge, MA: MIT Press.

Leman, M. (1989). Symbolic and subsymbolic information processing in models of musical communication and cognition. *Interface*,18, 141-160.

Lerdahl, F. (1987). Timbral hierarchies. *Contemporary Music Review* 2, 135-60.

Lerdahl, F. & Jackendoff, R. (1983). *A generative theory of tonal music*. Cambridge, MA: MIT Press.

Lischka, C. (1991). Understanding music cognition: a connectionist view, *Representations of Musical Signals*. Cambridge, MA: MIT Press.

Luce, D. & Clark, M. Jr. (1967). Physical correlates of brass-instrument tones. *Journal of the Acoustical Society of America*, 42 (6), 1232-1243.

Lufti, R. A. (1992). Informational processing of complex sound. III: interference. *Journal of the Acoustical Society of America*, 91 (6), 3391-3401.
Malt, M. (1994). Modelos matemáticos e composição assistida por computador, sistemas estocásticos e sistemas caóticos, *Anais do I simpósio brasileiro de computação e música*. Belo Horizonte, MG: Maurício Loureiro.
Marsden, A. & Pople, A. (1989). Towards a connected distributed model of musical listening. *Interface*, 18, 61-72.
Mason, C. R., Kidd, G., Jr., Hanna, T. & Green, D. M. (1984). Profile analysis and level variation. *Hearing Research*,13, 269-275.
Massaro, D. W. & Cowan, N. (1993). Information processing models: microscopes of the mind. *Annual Review of Psychology*, 44, 383-425.
Matlab (1992). *MATLAB User's Guide*. New York: Mathworks, Inc.
McAdams, S. (1982). Spectral fusion and the creation of auditory images, *Music Mind and Brain*. New York: Plenum Press.
McNicol, D. (1972). *A primer of signal detection theory*. London: George Allen & Unwin Ltd.
Melara, R. D. & Marks, L. E. (1990a). HARD and SOFT interacting dimensions: differential effects of dual context on classification. *Perception & Psychophysics*, 47, 307-325
Melara, R. D. & Marks, L. E. (1990b). Interaction among auditory dimensions: timbre, pitch, and loudness. *Perception & Psychophysics*, 48, 169-178.
Melara, R. D. & Marks, L.E.(1990c). Perceptual primacy of dimensions: support for a model of dimensional interaction. *Journal of Experimental Psychology: Human Perception and Performance*, 16, 398-414.
Menezes Filho, F. (1987). *Apoteose de Schoenberg*. São Paulo: Nova Stella/EdUSP.
Mitroff, I.I. (1974). *The subjective side of science*. Amsterdam: Elsevier.
Moles, A. (1969). *Teoria da informação e percepção estética*. Rio de Janeiro: Tempo Brasileiro.
Mozer, M.C. (1991). Connectionist music composition based on melodic, stylistic, and psychophysical constrains, *Music and connectionism*. Cambridge, MA: MIT Press.
Nielzen, S., & Olsson, S. (Eds.) (1989). *Structure and perception of electroacoustic sound and music*. Amsterdam: Elsevier.
Oppeinheim, A. & Schafer, R. (1975). *Digital signal processing*. Englewood cliffs: Prentice-Hall.
Pachella, R. G. (1974). The interpretation of reaction time in information-processing research, *Human information processing: tutorials in performance and cognition*. Kantowitz, B.H. (Ed.). Hillsdale, New Jersey: Erlbawm.
Parncutt, R. (1989). *Harmony: a psychoacoustic approach*. Berlin: Springer-Verlag.
Perfetti, C. A. & Bell, L. C. (1991). Phoneme activation during first 40 ms. of word identification: evidence from backward masking and priming. *Journal of Memory and Language*, 30, 473-485.
Pierce, J. (1983). *Musical Sound*. New York: Scientific American Books.
Pitt, M. A. & Crowder, R. G. (1992). The role of spectral and dynamic cues in imagery for musical timbre. *Journal of Experimental Psychology: Human Perception and Performance*, 18 (3), 728-738.
Polansky, L. & Bassein, R. (1992). Possible and impossible melody: formal aspects of contour. *Journal of Music Theory*, 36.2, 259-284.
Prince, W.F. (1972). A paradigm for research on music listening. *Journal of Research in Music Education*, 28, 445-455.
Pylyshyn, Z. (1984). *Computation and cognition*. Cambridge, MA: MIT Press.
Ramirez, R.W. (1985). *The FFT. Fundamentals and concepts*. New Jersey: Prentice Hall, Englewood Cliffs.
Remez, E.R., Rubin, P.E., Berns, S.M., Pardo, J.S., Lang, J.M. (1994). On the perceptual organization of speech. *Psychological Review*, 101 (1), 129-156.
Richards, V. M., Onsan, Z. A. & Green, D. M. (1989). Auditory profile analysis: potential pitch cues. *Hearing Research*, 39, 27-36.
Risset, J.C. & Mathews, M.V. (1969). Analysis of musical instrument tones. *Physics Today*, 22 (2), 23-30.
Risset, J.C. (1991). Timbre analysis by synthesis: representations, imitations and variants for musical composition, *Representation of musical signals*. Cambridge, MA: MIT Press.
Roads, C. (1991). Asynchronous granular synthesis, *Representation of musical signals*. Cambridge, MA: MIT Press.
Rock, I. (1984). *Perception*. New York: Scientific American Books.
Roederer, J.G. (1975). *Introduction to the physics and psychophysics of music*. New York: Springer-Verlag.
Rumelhart, D.E. & McClelland, J.L. (1986). *Parallel distributed processing*. Cambridge, MA: MIT Press. Vol.1, Vol.2.
Sano, H. & Jenkins, B.K. (1991). A neural network model for pitch perception, *Music and connectionism*. Cambridge, MA: MIT Press.

Schaeffer, P. (1993). *Tratado dos Objetos Musicais*. Brasília: Edunb.
Schoenberg, A. (1974). *Tratado de Armonia*. Madrid: Real Musical.
Semal, C. & Demany, L. (1991). Dissociation of pitch from timbre in auditory short-term memory. *Journal of the Acoustical Society of America* 89, 2404-2410.
Singh, P. G. (1987). Perceptual organization of complex tone sequences: a tradeoff between pitch and timbre?. *Journal of the Acoustical Society of America*, 82, 886-899.
Siegel, J. A. (1981). Culturally Defined Learning Experience, *Documentary Report of the Ann Arbor Symposium: Applications of Psychology to the Teaching and Learning of Music*. Reston: Music Educator National Conference.
Slawson, W. (1985). *Sound color*. Berkeley, CA: University of California Press.
Sloboda, J. A. (Ed.) (1985). *Generative processes in music: the psychology of performance, improvisation and composition*. New York: Oxford University Press.
Suppes, P., Pavel, M., Falmagne, J.C. (1994). Representations and models in psychology. *Annual Review of Psychology*, 45, 517-544.
Terhardt, E., Stoll, G., & Seewan, M. (1982). Pitch of complex signals according to virtual-pitch theory: tests, examples, and predictions. *Journal of the Acoustical Society of America*, 71, 671-678.
Todd, P. M. & Loy, G. D. (Eds.) (1991). *Music and connectionism*. Cambridge, MA: MIT Press.
Vos, J. and Rasch, R. (1982). The perceptual onset of musical tones, *Music Mind and Brain*. New York: Plenum Press.
Vriend, J. (1981). "Nomos Alpha" for violoncello solo - Xenakis 1966 - analysis and comments. *Interface*, 10, 15-82.
Wenner, A.M. & Wells, P.H. (1990). *Anatomy of a controversy: the question of a "language" among bees*. NewYork: ColumbiaUniversity Press.
Wertheimer, M. (1974). The problem of perceptual structure, *Human information-processing: tutorials in performance and cognition*. Kantowitz, B.H. (Ed.). Hillsdale, New Jersey: Erlbawm.
Wessel, D. (1979). Timbre space as a musical control structure. *Computer Music Journal*, 3, 45-52.
Wolf, F.A. (1981). *Taking the quantum leap*. San Francisco: Harper & Row.
Wolpert, R. (1990). Recognition of melody, harmonic accompaniment and instrumentation. *Music Perception*,7, 253-258.
Yost, W. A. & Watson, C. S. (Eds.) (1986). *Auditory processing of complex sounds*. New York, Hillsdale: Erlbaum.
Zwicker, E., Flottorp, C. & Stevens, S. S. (1957). Critical bandwidth in loudness summation. *Journal of the Acoustical Society of America*, 29, 548-557.

Acknowledgements

Financial support was provided by Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq-UnB. Software was bought with funds from Fundação de Apoio à Pesquisa do D.F. - FAPDF. Thanks to Dr. Bartholomeu T. Tróccoli, Dr. Juan M. Capasso, Lic. Diana M. Korchak, Ariadna Capasso, Ana Lúcia and Nahuel. Special thanks to Dr. Eduardo Miranda for regarding the time lag.

Dynamic MIDI Editor: A Windows Application for Computer Music

LUIZ ANTONIO PINHEIRO MARTINS
Oficina de Análise e Síntese da Imagem e do Som
Departamento de Ciência da Computação - UFMG
Caixa Postal 702 - 30161.970 - Belo Horizonte - MG - Brazil
email: *pinheiro@dcc.ufmg.br*

Abstract

The Dynamic MIDI Editor (DyME) is a Windows application for computer music that allows the user to apply the editing process of MIDI instruments over a musical sequence. This paper introduces DyME and tries to show that it extends the capabilities of MIDI protocol, allowing the definition of dynamic instruments. DyME offers the real time capabilities of MIDI and dynamic control of timbre, which is actually possible only in SWSS systems. DyME writes "parameter change" commands in MIDI files, integrating editing and sequencing processes.

Introduction

Systems for computer music has been developed along two main lines, in spite of the possibilities of interaction between them (Ballista, Casali, Chareyron, & Haus, 1992; Jaffe, Smith, & Porcaro, 1994) and the different synthesis methods developed: Software Sound Synthesis (SWSS) and Musical Instrument Digital Interface (MIDI). In the former, musical material is completely generated by software, which defines notes and instruments. In the latter, computer stores only notes and other control information, with the timbre defined by the instrument selected in a synthesizer.

Synthesis power of SWSS systems is greater than that of any MIDI synthesizer, because it is not limited by hardware and synthesis method of a specific synthesizer model. The various types of modules found in a synthesizer can be arbitrarily defined by software. Thus, any commercial synthesizer can be rebuilt, using acoustical compilers such as cmusic, cmix, or csound (Pope, 1993). Besides, each instrument defined in an acoustical compiler can use a different synthesis method, simulating different synthesizers.

On the other hand, the editing process of MIDI instruments is interactive. While in SWSS a relatively long processing time is needed before the sound file can be played, in MIDI the synthesis process is controled in real time, while the parameters that define an instrument are modified.

However, the instrument editing usually is used to define in advance an instrument that, further, can be selected to play a musical passage. In MIDI, the possibilities of instrument transformation during play are very limited, because editing and sequencing are normally separate processes.

DyME is an environment for the definition of dynamic MIDI instruments. These instruments are conducted along the MIDI sequence by sending "system exclusive - parameter change" messages. That way, the editing process is implemented dynamically, in agreement with the expressive needs of the musical flow.

Static Editing

After the introduction of the MIDI protocol, the music industry experienced an enormous development and from day to day more musicians have access to MIDI instruments. The definition of instruments by editing process became part of the composer's work, besides his or her other musical tasks. MIDI devices are very popular thanks to features such as high reliability, ease of use, low cost and related software availability.

However, it is clear that MIDI instruments do not present the same flexibility as a conventional instrument with relation to parameters such as dynamics, articulation, vibrato, attack and tune, for example. Besides the inherent problems with the MIDI protocol (Moore, 1988), the behavior of MIDI instruments is musically poor because it is maintained almost constant along the musical flow, except by alterations defined by touch and periodic oscillations. Editing is usually a static process, and the behavior of the instrument is bounded in advance by that process.

Dynamic Editing

The possibilities of dynamic control over the editing process of MIDI instruments are very limited, unless a special tool is used. Similar instruments can be selected sequentially by the "program change" command to simulate the desired effect, but in this case the number of changes would be limited by the number of synthesizer programs, which is not enough for smoothly controlling that process. Besides, the "program change" command can be used only over pauses, because it introduces sound discontinuities (clicks) when it is applied over notes in execution.

"System exclusive is the great escape hatch of MIDI" (Loy, 1985). In some commercial software sequencers, it is possible to send "system exclusive" strings to the synthesizer, including the "parameter change" messages used by DyME. In this case, for each parameter change it is necessary to consult the address and range of the parameter that will be modified, to translate it to hexadecimal form and to write, one by one, those strings. These programs are not efficient to manage hundreds of "parameter change" strings, as can be done with DyME, because they are not a special environment for this task.

DyME tries to fill the lack of resources for dynamic control of timbre observed in MIDI, by increasing the flexibility of MIDI instruments. In this manner, various kinds of timbre transformations can be implemented: Sometimes it is desirable to implement many changes on weak parameters to produce a smooth transformation in timbre. Otherwise, it is convenient to change a strong parameter that has immediate effect on the sound characteristics (Jaffe, 1994). In any case, the dynamic editing process is associated to the idea of dynamic instruments which adapt to expressive needs of the musical flow.

The dynamic control of timbre can be implemented on SWSS systems, where the definition of instruments is almost unlimited: related instruments can be called sequentially along the score to reach that objective, or a control stream can be used on the instrument. DyME does not propose a very original process in computer music, but claims to extend the capabilities of the MIDI instruments in this field. It takes advantage of the fact that the MIDI synthesis process is more interactive than the same process using SWSS systems. Besides, its high level interface controls the dynamic editing process in an easy and efficient way.

Naturally, the dynamic editing process implemented by DyME is limited by the hardware of the synthesizer used. However, the musical result of this process is very attractive, greatly overcoming the possibilities normally presented by that synthesizer. DyME shows clearly that the musical limitation of MIDI instruments is caused by the absence of resources for dynamic control of timbre in the MIDI protocol. Unfortunately, the only way to solve this problem is using system exclusive commands, that are specific to a particular synthesizer model.

A Dynamic Editing Session

The main characteristic observed in DyME is its operative simplicity. Anybody who has already performed a conventional editing of a MIDI instrument can implement dynamic editing by just adding one step to that process: applying the editing process previously defined on the selected part of a sequence. DyME compares the configuration preceding the instrument editing with the subsequent one and approaches them, through "parameter change" messages, which are written on a midifile. In this process, it interpolates the parameter values depending on which option has been selected.

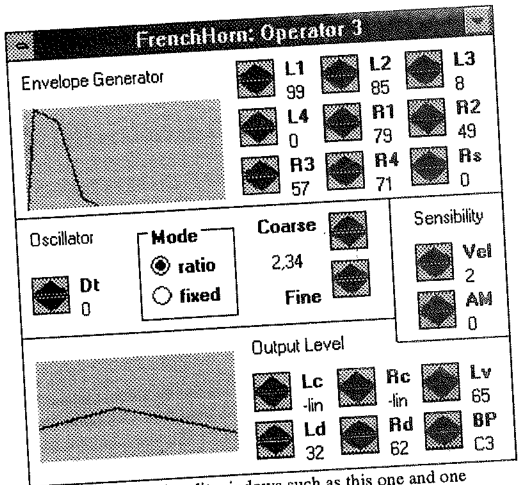


Fig. 1: There are six edit windows such as this one and one edit window for general parameters of the TX802.

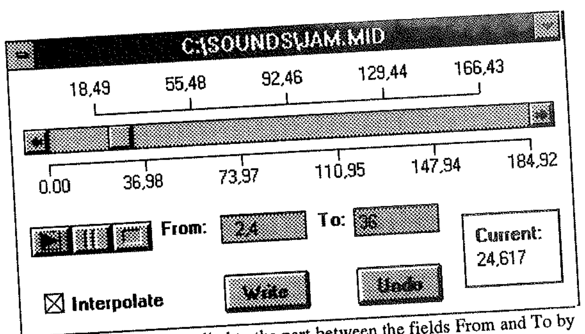


Fig.2: The editing is applied to the part between the fields From and To by the button Write. The Undo button undoes the last write operation.

The dynamic editing process is as follows. The user opens a midifile recorded in advance and selects the option Receive which requests the parameters from the current instrument in the synthesizer. The next part resembles a static editing process: in the edit windows (there are seven edit windows in DyME) the parameters are modified. Thus, another parameter configuration and a new timbre are defined.

However, there is a difference between this process and the static one: the user has in mind the musical passage which will receive that dynamic editing. Rather than defining a closed instrument to be further selected, the user is controling a dynamic transformation from the previous timbre to that defined by him or her. In the last part of the process, the alterations implemented on the parameters are applied to the musical sequence by the Write button, that is present in the dynamic edit window. In this window the sequence can be played, and the editing process previously defined is reproduced during play. If the result of the last operation has not been interesting, it can be undone by the Undo button. When the user returns to the edit windows to restart the process, they are updated to the point where the execution of the midifile has stopped.

DyME supports midi files format 1.0. Each write operation creates a new track of "parameter change" messages in the midifile. It is possible to record up to 32767 "parameter change" tracks minus the number of tracks recorded previously by the sequencer. A unique write operation can record up to hundreds of "parameter change" messages on the midifile.

Implementation

DyME was implemented in the Microsoft Visual Basic programming system, with a DLL (dynamic linking library) defined in C++, using the compiler Borland C++ 3.1 (Norton, & Yao, 1992). The ease for interface development observed in Visual Basic and the support offered by this platform to API windows (including multimedia) justify its use. The DLL are used to tasks such as communication with the synthesizer, and reading and writing of midi files, where interpreted Basic code would threaten the processing efficiency.

The synthesizer used is the Yamaha TX802, that uses FM synthesis. The use of this synthesis method (Chowning, 1973) is coherent with the objectives of the work. On this module, it is possible to solve the MIDI deficiency on the microtonal field and to control the synthesis process since its source, which is not true for synthesizers that use sampled sounds. Further versions of DyME will be determined by the system exclusive of the synthesizer used. All the parameters that can be changed by that system exclusive will be able to be dynamically controled by DyME.

OOP Concepts Used by DyME

Visual Basic is an object-based (not object-oriented) programming system (Booch, 1994). In VB, the user never explicitly use concepts like inheritance, class, and polymorphism. Besides its object-based characteristics, Visual Basic is event-oriented also. The program is structured in a bottom-up way, and the user is responsible for the execution order of the program.

The manner the program was built reflects itself in its operation. For example, when the user creates an edit window called Operator 1, it is an object of the Operator class (in Visual Basic classes are represented by foms and controls). This object shares the functionality of that class, being updated by the musical play. A great number of buttons are used in the edit windows. The Spin button class follows generic methods, which are implemented in a different manner in each instance of a button object.

Limitations

Like any system that controls "system exclusive" streams, DyME is dedicated to a synthesizer model, and its interface defined especially for that synthesizer model. Naturally, it is possible and desirable to implement different versions of DyME for other commercial synthesizers.

It was observed in the TX802 (version 07/25/87) module that it is not possible to select the program that will receive the "parameter change" message by the channel information present in that string. The synthesizer received "parameter change" strings only in the program indicated by its cursor. So, using that module it was not possible to send "parameter change" messages to various channels simultaneously and to take advantage of its multitimbral resources.

Conclusion

The day is very near in which cheap SWSS systems will be able to work in real time. Maybe then it will not be necessary to use DyME, which is limited by the specific architecture of a synthesizer model and inherent problems of MIDI protocol. For the present, DyME proposes to extend the possibilities of MIDI synthesizers to timbral control, gaining the advantages of their great diffusion and real time capabilities.

References

Ballista, A., Casali, E., Chareyron, J., & Haus, G. (1992). A MIDI/DSP Sound Processing Enviroment for Computer Music Worksation. *Computer Music Journal* 16(3): 57-72.
Booch, G. (1994). *Object-Oriented Analysis Design with Applications*. Benjamin/Cummings.
Chowning, J. (1973). The Synthesis of Complex Audio Spectra by Means of Frequency Modulation. *Journal of Audio Engineering Society* 21(7): 526-534
Jaffe, D., Smith, J., & Porcaro, P. (1994). The Music Kit on a PC. *Proceedings of the I Brazilian Symposium of Computer Music*: 63-69.
Jaffe, D. (1994). An Overview of Criteria for Evaluating Synthesis and Processing Techniques. *Proceedings of the I Brazilian Symposium of Computer Music*: 53-60.
Loy, G. (1985). Musicians Make a Standard: The MIDI Phenomenon.. *Computer Music Journal* 9(4).
Microsoft Windows: Multimedia Programmers Reference. Microsoft Press, 1990
Microsoft Visual Basic Programmer's Guide. Microsoft Press, 1993
Moore, R. (1988). The Dysfunctions of MIDI. *Computer Music Journal* 12(1):19-28
Norton, P., & Yao, P. (1992). *Programando em Borland C++ para Windows*. Berkeley Brasil Editora.
Oppenheim, D. (1989). *MIDI File Especification*. International MIDI Association.
Petzold, C. (1992). *Programando em Windows 3.1*. Makron Books.
Pope, S., T. (1993). Machine Tougues XV: The Three Packages for Software Sound Synthesis. *Computer Music Journal* 17(2): 23-53.
TX802 Owner's Manual - Yamaha Music Foundation, 1988

Acknowledgements

This work is part of my Master degree dissertation, defended in May 23, 1995 in the Department of Computer Science of Federal University of Minas Gerais (UFMG). It was supervised by Wilson de Pádua Paula Filho, and supported by National Council of Scientific and Technological Development (CNPq).

Software Auto-Instrucional em Teoria Musical

MÁRCIO DA COSTA PEREIRA BRANDÃO
MÍRIAM RICA SAMBUICHI
Laboratório de Processamento Espectral
Departamento de Ciência da Computação
Universidade de Brasília - Brasília, DF
CEP 70910-900 BRASIL
e-mail: brandao@cic.unb.br

RESUMO

Um courseware multimídia introdutório em teoria musical, desenvolvido para o ambiente Windows, é aqui descrito. Os tópicos básicos abordados são agrupados nas seguintes lições: definição de música e seus componentes; claves e localização das notas no teclado; divisão proporcional de valores; compassos; tom, semitom e alterações; classificação dos intervalos naturais; escalas maiores e menores. A apresentação é feita utilizando a notação musical convencional, sendo ilustrada com exemplos sonoros pertinentes. A navegação se dá através de botões que permitem ao aluno seguir em frente, voltar a telas anteriores ou mesmo mudar de lição. O aprendizado pode ser avaliado através de exercícios de fixação apresentados ao longo das lições ou de testes de avaliação no final de cada lição.

1. Introdução

Um software auto-instrucional - ou courseware - possibilita que se aprenda sozinho, de uma forma interativa, num ritmo individualizado e utilizando o próprio tempo disponível. Eles ainda não substituem por completo o processo de ensino/aprendizado tradicional, mas mostram-se eficazes na execução de tarefas tais como repetir, ordenar, calcular, relacionar, ler e escrever, por exemplo. Atualmente, observa-se que um número cada vez maior de pessoas tem se interessado por esta forma de aprendizado graças à integração de recursos multimídia (texto, sons e imagens) na apresentação de informações de forma agradável e interativa.

O objetivo deste trabalho é descrever as etapas de desenvolvimento (Franco & Rêgo, 1991) de um courseware multimídia introdutório em teoria musical (Sambuichi, 1994). O conteúdo deste courseware limita-se a alguns pontos da teoria musical, sendo a linguagem voltada para um público que não necessita possuir um conhecimento prévio do assunto.

2. Etapas do Desenvolvimento

A metodologia adotada para a construção deste courseware seguiu as seguintes etapas:

- Análise
- Definição dos Objetivos.
- Planejamento.
- Desenvolvimento
- Programação
- Validação
- Revisão

Análise

Na etapa de Análise, além da determinação do conteúdo a ser abordado, deve ser delineado qual é o público-alvo. No presente trabalho esta análise foi realizada junto a alunos dos cursos de teoria musical e de canto coral do Conservatório de Música de Brasília, e também junto a pessoas sem conhecimento formal de música. As respostas aos questionários aplicados indicaram a conveniência do desenvolvimento de um curso desta natureza, voltado para um público-alvo composto principalmente por adolescentes e adultos, estudantes de música ou não. Para atender a este perfil de usuários, optou-se por abordar apenas tópicos introdutórios da teoria musical, como mostra a Figura 1.

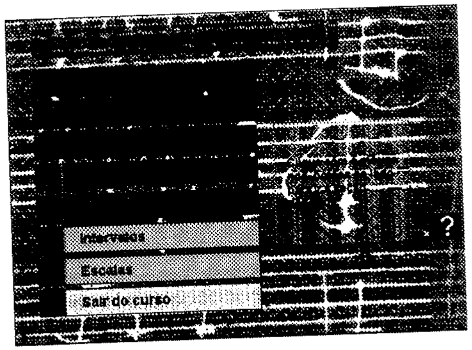


Figura 1 - Tela de apresentação do curso

Definição dos Objetivos

No início da abordagem de cada tópico selecionado, o aluno é informado sobre os objetivos instrucionais específicos, conforme a Figura 2. As avaliações se baseiam nestes objetivos, que estão descritos em termos do desempenho esperado.

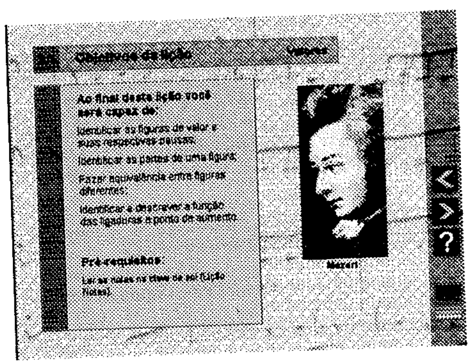


Figura 2 - Tela de objetivos de uma lição

Planejamento

Na fase de Planejamento, modularizou-se o material instrucional em lições, cada uma com objetivos, conceitos e avaliações, de forma a constituir uma sessão fechada e independente da instrução. No início de cada lição o aluno é informado sobre os eventuais pré-requisitos necessários, como mostra a Figura 2. Exercícios de avaliação foram introduzidos ao longo das lições com o objetivo de fixar parte do conteúdo apresentado, sendo possível ao aluno rever conceitos abordados para auxiliar na resolução. Já ao final de uma lição, todo o conteúdo apresentado é avaliado através de testes constituídos primordialmente de questões de múltipla escolha, conforme mostra a Figura 3. Caso o resultado obtido nestes testes não seja satisfatório, é sugerida uma revisão do conteúdo da lição.

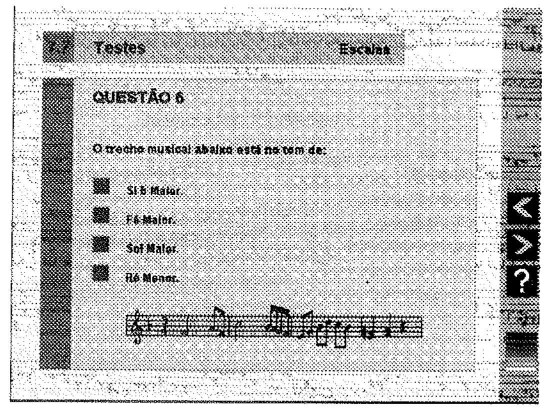


Figura 3 - Questão de múltipla escolha dos testes

Desenvolvimento

Na etapa de Desenvolvimento foi realizada a diagramação das telas que compõe o courseware. Aqui foram especificados os pontos de interação do aluno com o computador tais como: botões ativos, mensagens, respostas e feedbacks para os testes. Também foram definidos padrões para as telas de apresentação de informação, que foram divididas em três zonas: (I) zona de identificação para orientar o aluno quanto à sua posição no curso; (ii) zonas de apresentação do conteúdo; (iii) zonas de mensagens opcional; (iv) zona de botões para a navegação no curso.

Programação

Utilizou-se o software de autoria Toolbook 3.0 (Asymetrix Corp., 1994), que facilita a implementação de sistemas de menus e de bancos de dados, além de integrar textos, gráficos e sons de uma forma simples. Esta facilidade advém do fato do Toolbook tratar gráficos, botões, campos de texto e campos de registros próprios de bancos de dados como objetos, sendo cada objeto uma entidade separada e distinta dos outros.

Para a elaboração das telas contendo símbolos musicais convencionais, foram criados arquivos BMP através do sistema de editoração e sequenciamento musical Encore 2.5 (Passport Design, 1992),

Geração dos Sons

Como as placas de sons disponíveis atualmente na maioria dos equipamentos multimídia apresentam sintetizadores de som de baixa qualidade, voltados principalmente para a sonorização de jogos, optou-se pelos sons previamente digitalizados. Esta decisão repercutiu na forma de distribuição do curso - em CD-ROM - já que os arquivos de sons digitalizados ocupam um espaço grande quando comparados aos arquivos do tipo MIDI.

Foram gerados arquivos do tipo WAV a uma taxa de amostragem de 44100 Hz, utilizando-se módulos de som com uma maior qualidade sonora, quando comparados com os sintetizadores internos à maioria das placas de som atualmente disponíveis. Os equipamentos envolvidos nesta etapa, além de um microcomputador PC-486 DX-33 multimídia, foram:

- Music Workstation W-30 (Roland Corp., 1989).
- Interface MIDI MPU-IPC/T.
- Módulo de som multi-timbral SC-33 (Roland Corp., 1992)
- Placa de som Sound Blaster 16 ASP.

Validação e Revisão

Estas fases estão por ser iniciadas e consistirão na aplicação do courseware em uma turma-piloto com os seguintes objetivos:

- detectar erros ou ambiguidades na apresentação do conteúdo.
- detectar dificuldades no entendimento do conteúdo.
- detectar erros de navegação.
- verificar se há pontos de desmotivação.
- coletar sugestões.

As respostas aos questionários aplicados aos voluntários desta turma-piloto subsidiarão as alterações finais a serem efetivadas na etapa de revisão.

3. Considerações Finais

O courseware aqui descrito foi inicialmente desenvolvido na disciplina "Estágio Supervisionado", do bacharelado em Ciência da Computação da Universidade de Brasília, no segundo semestre de 1994 (Sambuichi, 1994). Atualmente está sendo aperfeiçoado para ser comercializado pela MSD Software, empresa incubada no Centro de Desenvolvimento Tecnológico da Universidade de Brasília.

4. Referências

Asymetrix Corp., (1994),*Toolbook User's Manual*, Asymetrix Corp.

Franco, L. H. S & Rêgo, P. R. R. (1991). Estudos dos Requisitos para a Construção de um Software Instrucional. *Trabalho de Graduação*, Departamento de Ciência da Computação, UnB, Brasília.

Sambuichi, M. R. (1994). Software Educacional de Teoria Musical. *Trabalho de Graduação*, Departamento de Ciência da Computação, UnB, Brasília.

Hannafin, M. J. & Peck, K. L. (1988). *The Design, Development and Evaluation of Instructional Software*. Macmillan Publishing Company.

Passport Designs (1992), *Encore User's Guide*, Passport Designs.

Roland Corp. (1989), *W-30 for FD Owner's Manual*, Roland Corp.

Roland Corp. (1992), *SC-33 Owner's Manual*, Roland Corp.

SETMUS
Uma ferramenta computacional para o Ensino da Música.

Eloi Fernando Fritsch, Rosa Maria Viccari

Instituto de Informática - UFRGS,

Curso de Pós-Graduação em Ciência da Computação - CPGCC, Av. Bento Gonçalves, 9500 Bloco IV - Agronomia - Campus do Vale, CEP 91501-970 Porto Alegre - RS - Brasil,

Caixa Postal: 15064 FAX: (051) 336-5576,

E-mails: {fritsch, rosa}@inf.ufrgs.br

Abstract

The purpose of this paper is to present the implementation of software named SETMUS - Sistema Especialista para Teoria MUSical. The SETMUS was implemented in Hypercard using the object oriented philosophy. The program has a data base with scales and arpejos. Each scale and arpejo is a card with an address that can be called from another card. Besides this it also has a musical calculator to solve users doubts.

The SETMUS objetive is to teach the theory and the musical perceptive through the user program interaction. The theory is treated in a clear and simple way because the system presents a graphical interface that makes easy its understanding. The user uses the system touching the notes in the screen with the aid of the "mouse".

One of the applications of SETMUS is to serve as a system with exercises to those who are studing musical theory, more specifically, musical harmony. The SETMUS is an expert software that accumulates the functions of teacher (at the exercise hour), didactical book (with the scores to be played by the student) and of the musical instrument. There is also a "PlayBack" function implemented. It is always used when the user wants to listen and to visualise the notes he had played and wants to compare with the correct and incorrect scales and arpejos that were played by the user. The comparasion is both visual and sonore providing the training of the user's musical perception.

1. Introdução

Este artigo apresenta um protótipo utilizado como ferramenta não tradicional em ensino de teoria musical, o SETMUS. São abordados aspectos de interface gráfica, calculadora musical, "PlayBack", sistemas de navegação, reconhecimento de escalas e arpejos, didática pedagógica, recursos de som e demais características de funcionalidade.

Devido ao grande avanço tecnológico, a maneira de fazermos música mudou. Por essa razão, os sistemas de aprendizagem musical também devem assimilar essas mudanças[YAV92]. A busca de métodos não tradicionais, enriquecidos de criatividade, abrem novas perspectivas no campo da música[FRI92]. Tais métodos combinados aos métodos tradicionais, quando apropriados, podem levar-nos a um resultado ainda melhor no processo educacional, principalmente com a utilização do computador.

O grupo de Inteligência Artificial do Instituto de Informática da Universidade Federal do Rio Grande do Sul tem investigado a área de tutores inteligentes sob os aspectos teóricos [OLI94] e práticos, como é o caso do presente trabalho que aborda principalmente interface.

2. Apresentação do SETMUS

O SETMUS (Sistema Especialista para Teoria Musical) foi implementado em HyperCard utilizando a filosofia de orientação a objetos. O programa possui um Banco de Dados formado por escalas e arpejos de todos os vinte e quatro modos maiores e menores.

Cada escala ou arpejo constitui-se num cartão com um determinado endereço que pode ser chamado a partir de outro cartão. Além disso dispõe de uma calculadora musical para solucionar dúvidas do usuário.

3 . Funcionamento do sistema

O SETMUS tem como função o ensino da teoria e percepção musical através da interação do usuário com o programa. A teoria é tratada de forma clara e simples, pois como pode ser observado na Figura 1, o sistema apresenta uma interface gráfica que facilita a compreensão. O usuário utiliza o sistema tocando as notas na própria partitura com o auxílio do "mouse". Uma das aplicações a que o SETMUS destina-se é servir como um sistema de exercícios para alunos que estão estudando a teoria musical, mais especificamente, a parte de harmonia musical[FAR63]. O SETMUS é um protótipo especialista pois substitui o conhecimento do especialista (professor) na hora do exercício e acumula as funções do livro didático (com as partituras a serem tocadas pelo aluno) e o instrumento musical para a execução (ou solfejo melódico) das notas lidas pelo aluno.

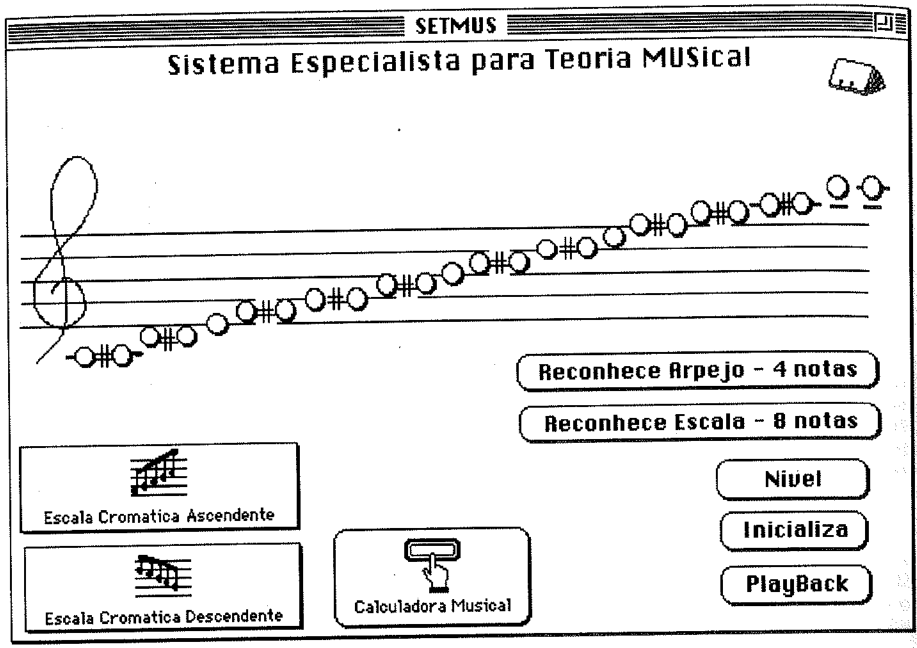


Figura 1. Tela principal.

3.1 Processo de reconhecimento das notas tocadas

Após o usuário tocar toda a sequência de notas que desejada, basta escolher a opção de reconhecimento, seja ela escala ou arpejo. Se a escala ou arpejo tocado existir, então o sistema responderá com a classificação correta e perguntará se o usuário quer mais detalhes sobre a estrutura musical apresentada. Se a escala selecionada não existir, então o sistema apresentará uma mensagem acusando que a escala não está disponível no sistema e aconselhará ao usuário perguntar ao sistema o porquê da escala não estar disponível. Caso o usuário use a possibilidade de explanação, então deverá digitar uma das perguntas listadas na sintaxe de perguntas possíveis de serem formuladas ao SETMUS. Digitanto a pergunta e confirmando, o SETMUS apresentará a resposta solucionando assim a dúvida do usuário. Se o SETMUS, ao reconhecer determinada estrutura musical, apresentar maiores detalhes por opção do usuário, então novas explicações com escalas serão mostradas ao usuário.

3.2 Função de "PlayBack" e seu papel de diagnóstico

Existe também uma função de "PlayBack" implementada por um botão presente na tela principal do software, *Figura1*. Esta função também é encontrada ao longo da pilha de cartões correspondentes às escalas e arpejos. Nestes cartões aparece sob a forma de um botão com a letras PB como pode ser observado na *Figura 2*. Ela é usada pelo usuário sempre que este desejar ouvir e visualizar as notas que tocou e comparar com a escala ou

arpejo correto. O "PlayBack" fornece um diagnóstico do erro já que compara escalas e arpejos corretos com incorretos que foram tocados pelo usuário. A comparação é tanto visual quanto sonora, possibilitanto o treinamento da percepção musical do estudante.

3.3. A Calculadora Musical

A calculadora musical, *Figura2*, pode ser utilizada pelo usuário sempre que este enfrentar dúvidas ao tocar as notas. Para utilizar a calculadora basta saber as notas musicais e os dois tipos de acidentes mais comuns, o sustenido e o bemol.

3.3.1.Selecionando escalas e arpejos

Existe uma sequência lógica na qual os botões devem ser escolhidos para a calculadora fornecer a escala ou arpejo desejado. Primeiramente o usuário seleciona o botão da tela principal para invocar a calculadora musical.

Após o painel da calculadora aparecer na tela, *Figura 2*, é necessário, primeiramente, que o usuário escolha a nota seguida do modo e, se for o caso, escolha também o acidente. Para visualizar um novo cartão com a resposta, o usuário deverá ainda escolher se a resposta será fornecida em foma de escala ou arpejo.

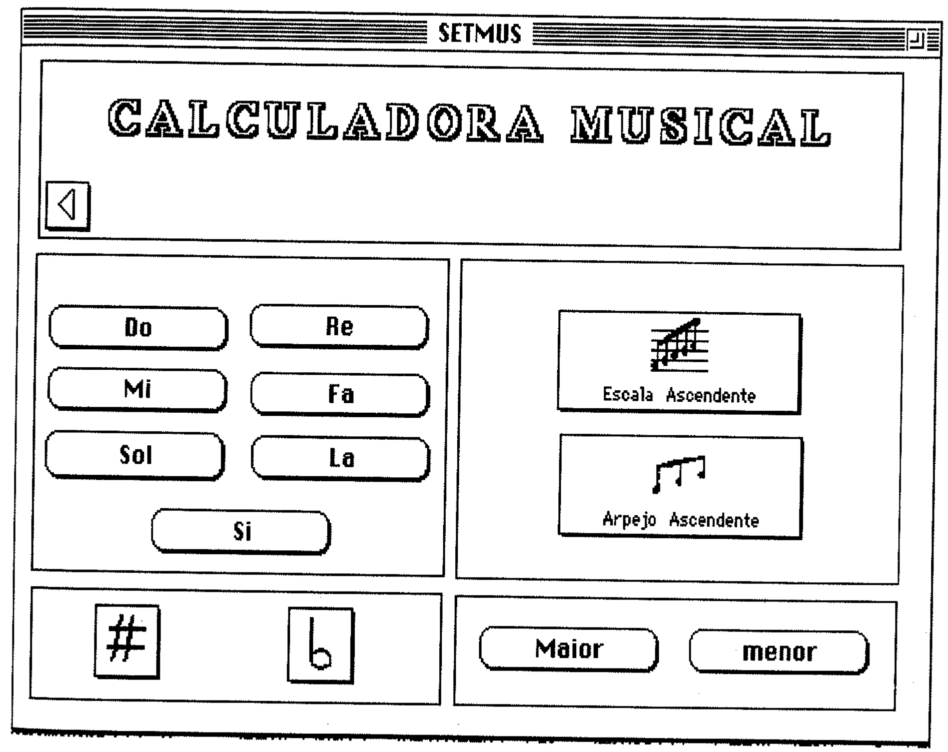


Figura 2 - Painel de controle da calculadora musical

Tomando como exemplo que o usuário aprendiz tenha dúvidas com relação a escala de RéMaior e, antes de executar as notas na tela principal do SETMUS, queira consultar a calculadora. O usuário poderá escolher a nota Ré, o modo maior e finalmente a opção de escala musical. O cartão correspondente à escala solicitada, *Figura 3*, será selecionado e aparecerá na tela para esclarecimento do estudante de teoria musical.

Os cartões explicativos possuem a escala ou arpejo que pode ser tocado pelo usuário bem como possibilidade de navegação para a relativa do tom em questão. Caso o estudante não queira selecionar as notas da escala com o "mouse", ele ainda pode escolher o botão que executa automaticamente a escala da partitura. Também existe a possibilidade do usuário acessar outro cartão com a mesma escala apenas trocando o modo. Isto serve principalmente para o discernimento entre os modos maiores e menores esclarecendo ao estudante tanto na forma audível quanto visual como pode ser observado no cartão mostrado na *Figura 3*.

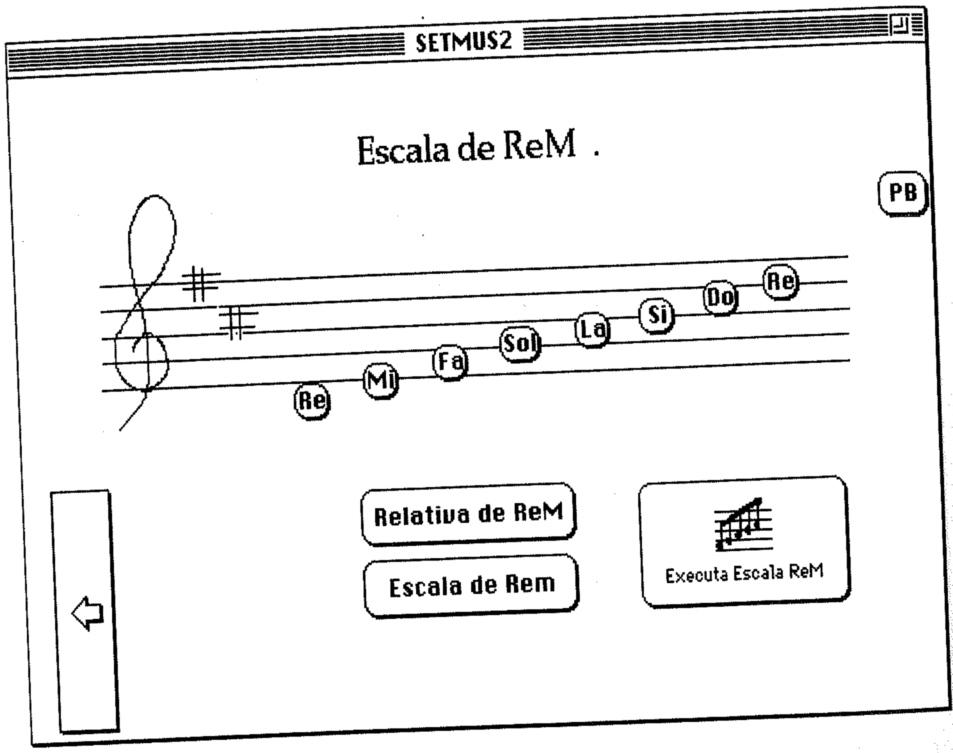


Figura 3 - Cartão explicativo sobre a escala de RéMaior.

4. Implementação

O Hypercard foi escolhido para a programação do software pela existência de rotinas prontas tanto no tratamento de som monofônico quanto para a interatividade com o usuário. O controle sobre o "mouse" e a programação visual orientada à objetos são de máxima importância para a construção de uma ferramenta que necessite incorporar didática e intuitividade na apresentação de conteúdos musicais[ROA85] .

4.1. Processo de classificação da sequência de notas reconhecidas:

O SETMUS apresenta notas musicais compreendidas em duas oitavas para que o usuário possa escolhê-las.

Ao escolher uma nota será produzido o som da mesma através do comando play do HyperTalk[GOO90]. Cada uma das notas tocadas diretamente nessa partitura inicial, consiste num botão que armazena seu número correspondente numa fila para possibilitar o ordenamento da sequência de notas tocadas para posterior análise.

Para fazer o reconhecimento (classificação) da escala ou arpejo, é usado um algoritmo para obter os intervalos entre as notas[MIR90][HAN84][ADO63]. Após a obtenção dos intervalos, uma série de regras são testadas até determinar se a estrutura musical é ascendente/descendente, maior/menor e também a nota que se baseia a escala ou arpejo.

4.1.1. Algoritmo para reconhecimento de escalas e arpejos

Se
(2°nota - 1°nota = 2) e
    (3°nota - 2°nota = 2) e
    (4°nota - 3°nota = 1) e
    (5°nota - 4°nota = 2) e
    (6°nota - 5°nota = 2) e
    (7°nota - 6°nota = 2) e
    (8°nota - 7°nota = 1)
então
    escala maior com nome da 1°nota.

Se (2°nota - 1°nota = 2) e
    (3°nota - 2°nota = 1) e
    (4°nota - 3°nota = 2) e
    (5°nota - 4°nota = 2) e
    (6°nota - 5°nota = 1) e
    (7°nota - 6°nota = 3) e
    (8°nota - 7°nota = 1)
então
    escala menor com nome da 1°nota.

Se (2°nota - 1°nota = 4) e
    (3°nota - 2°nota = 3) e
    (4°nota - 3°nota = 5)
então
    arpejo maior com nome da 1°nota.

Se (2°nota - 1°nota = 3) e
    (3°nota - 2°nota = 4) e
    (4°nota - 3°nota = 5)
então
    arpejo menor com nome da 1°nota.

4.2 Emissão de notas musicais utilizando o HyperTalk

Um dos motivos de usar o HyperCard é a simplicidade da linguagem HyperTalk voltada para aplicativos com navegação e a possibilidade de emitir sons monofônicos de uma maneira simples e suficiente para o alcance da trabalho em questão[YAV92].

Play é o comando que possibilita a emissão de notas musicais através do auto falante do Macintosh. Foram utilizadas, neste trabalho, três fontes sonoras: *harpsichord* (cravo) utilizada para a emissão das notas, *Boing* (efeito sonoro), *flute* (flauta). O tempo é opcional na utilização do comando Play, logo foi despresado pois o sistema não faz uso da métrica[GOO90].

A representação das notas foi feita convertendo o sistema de cifragem disponível na linguagem HyperTalk para o nome das notas. O bemol e o sustenido são empregados ao longo do programa nas notas que necessitam alguma alteração tonal. O sistema possui duas oitavas, portanto, a diferenciação de uma mesma nota tocada em oitavas diferentes, é feita por um número que identifica de onde provém a nota.

4.3 Navegação para as diversas partes do sistema

O usuário pode navegar facilmente pelas diversas partes do SETMUS. Isto é possível devido a programação dos botões (definidos como objetos em Hypercard) feita pelos respectivos "scripts" onde um simples comando go to endereça o destino. Neste caso, um determinado cartão[GOO90]. Todos os cartões foram ligados aos dois botões de reconhecimento de estruturas musicais. Sendo assim, no momento em que o usuário solicitar maiores explicações a respeito de uma escala reconhecida pelo sistema, este trará um novo cartão com a armadura, as notas e as eventuais alterações, permitindo o retorno ou o acesso a outro modo ou, então, um acesso à escala relativa .

As ligações entre os cartões, por meio do botão de relativas, obedecem a regra geral de que, nas escalas e arpejos, deve-se baixar uma terça menor para obter a relativa menor[FAR63].

As ligações entre os cartões, por meio da troca de modo, através do botão específico da operação é obtido simplesmente pela chamada do cartão que contiver a escala ou arpejo menor caso a atual seja maior.

A navegação através da calculadora se dá pela escolha sucessiva de botões presentes em seu painel. Cada botão, ao ser escolhido pelo "mouse", recebe um número que servirá para determinar a resposta que a calculadora irá fornecer.

4.4 Resposta utilizando palavras-chave

Quando o SETMUS não consegue reconhecer a escala ou o arpejo tocado, pelo usuário, é apresentada uma janela onde pode-se escolher algumas das perguntas pré-determinadas pelo sistema (como por exemplo: Por que a escala não está disponível? Por que o arpejo não está disponível). Após a escolha da pergunta , o SETMUS responde com cartões que possuem informações adicionais (como por exemplo: regras para a formação de escalas maiores e menores).

Do ponto de vista da implementação a análise das palavras-chave de cada pergunta é realizada através do comando "find" do HyperTalk que procura as no texto da pergunta fornecida pelo usuário, as palavras-chave.

5. Restrições do SETMUS

O protótipo apresentado neste trabalho é uma tentativa de representar uma pequena parte do conhecimento musical sob a forma de regras[MIR90][VIC90] [ROA85]. Apesar do sistema atualmente só aceitar as tríades dos acordes maiores e menores, existe a possibilidade de aumentar o número de regras tornando possível o reconhecimento de acordes de 4 sons sob a forma de arpejos [GOM88] Além disso, com pequenas alterações no programa, pela colocação de novas regras, é possível o reconhecimento também de outras escalas como pentatônicas e tons inteiros[ADO63].

O sistema só produz sons monofônicos e não possui implementação MIDI [GOO90] [MOR88] [FRI92].

6. Aplicações do SETMUS na música

O SETMUS, por ser um programa que apresenta um alto grau de interação com o usuário e possuir um conhecimento sobre as estruturas musicais nos modos maiores e menores, constitui-se num forte candidato para servir às necessidades de professores, alunos e profissionais da música. A combinação da partitura, o som e a didática torna o SETMUS capaz de reunir três elementos da teoria musical antes impossíveis de agrupar sem o uso do computador: O especialista musical, o instrumento e a partitura.

O SETMUS pode ser usado como um programa instrutor da parte de harmonia musical pois possui um diagnóstico exato tanto em termos sonoros quanto visuais [MOR88]. Além disso a calculadora musical pode, de maneira eficiente, responder a muitas perguntas que envolvam o raciocínio lógico sobre escalas e arpejos.

As principais vantagens do uso do SETMUS como tutor musical são:

- Serve como instrumento musical pois os exercícios para a fixação da teoria e percepção podem ser realizados no próprio SETMUS;
- O estudante aprende de maneira direta via calculadora musical ou através de tentativas e erros via escolha das notas na partitura;
- O sistema fornece justificativa e explanações mediante perguntas;
- A disposição das notas utilizada possui motivos didáticos;
- A navegação no sistema é facilitada pelos recursos de hipertexto;

- Possibilita o desenvolvimento do ouvido musical através da apresentação de intervalos e seus sons.

Outra aplicação do SETMUS é a de servir como uma bateria de testes para alunos que estão estudando a teoria musical, mais especificamente a parte de harmonia musical.

Para que o aluno possa ser avaliado ao interagir com o SETMUS, este deve ser convidado a tocar escalas e arpejos a gosto do avaliador de conteúdos. Através dos resultados obtidos o aluno pode ser observado quanto ao seu conhecimento musical.

Sobre este ângulo o SETMUS apresenta uma série de vantagens em relação à métodos convencionais de avaliação:

- Dispensa a utilização de instrumento musical pois o SETMUS pode emitir os sons necessários para o teste;
- Dispensa a presença do especialista para determinar se as respostas do aluno que está sendo avaliado foram corretas ou não, ainda havendo, em caso de dúvida, a justificativa do sistema. Dessa forma a avaliação poderá ser aplicada até por uma pessoa que entenda pouco de música, basta ver se o aluno conseguiu acertar a questão e depois comunicar os resultados ao especialista;
- Substitui lápis, papel e borracha.

7. Conclusão

Este trabalho apresentou o SETMUS, um sistema destinado ao reconhecimento de escalas e arpejos que visa o ensino dessas estruturas musicais. A interface desenvolvida para o SETMUS pode ser adequada a outros sistemas que necessitem de flexibilidade e interatividade com o usuário facilitando, dessa forma, a utilização do software.

Testes realizados com estudantes de música no próprio Instituto de Informática da UFRGS ressaltam a interface gráfica do SETMUS, juntamente com sua operacionalidade, possibilitando o seu uso didático no aprendizado do conteúdo musical. O uso de gráficos para interação é importante em sistemas tutores pois torna os diálogos menos monótonos e ajudam a fixar a atenção do usuário.

Apesar da plataforma Macintosh não ser muito utilizada nas escolas brasileiras, o projeto do SETMUS pode ser adaptado aos computadores da linha IBM PC e servir como uma ferramenta de auxílio ao aprendizado da harmonia musical.

Referências Bibliográficas

[ADO63] ADOLFO, Antônio. **O Livro do Músico:** Harmonia e Improvisação. Rio de Janeiro: Lumiar Editora, 1963.

[FAR63] FARIAS, Nelson. **A Arte da Improvisação Para Todos os Instrumentos**. Rio de Janeiro, Lumiar Editora, 1963.

[FRI92] FRITSCH, Eloi Fernando. **CAMM - Compositor Automático de Melodias Musicais** (Dissertação de Mestrado). Porto Alegre: CPGCC da UFRGS, 1994.

[GOM88] GOMES, Luis Carlos Elias. **Som Três - Pequeno Dicionário MIDI**. São Paulo: Editora 8 Três, 1988.

[GOO90] GOODMAN, Danny. **The Complete Hypercard 2.0 Handbook**. Bantam Doubleday Dell Publishing Group: New York, 1990

[HAN84] HANNON, **O Pianista Virtuoso**. São Paulo: Ricordi Editora, 1984.

[MIR90] MIRANDA, Eduardo Reck. **Música e Inteligência Artificial Paradigmas e Aplicações**. Porto Alegre: CPGCC-UFRGS, 1990. (Trabalho Individual, 200)

[MOR88] MORAES, Sylvio Messias. **Música por Computador I e II**. Rio de Janeiro: Revista Micro Sistemas, ATI editora Ltda, Rio de Janeiro.

[OLI94] Oliveira, Viccari, Coelho. **An equilibration-Oriented Conceptual Beliefs Representation.** Lisboa, II Congresso Ibero-Americano de Informática na Educacação, 1994.

[ROA85] ROADS, Curtis. **Research in music and artificial inteligence**. Computing Surveys, Cambridge, v17, n.2, 1985.

[YAV92] YAVELOW, Christopher, **Music & Sound Biblle**, San Mateo, California: IDG Books WordWide, Inc., 1992.

[VIC90] VICCARI, Rosa. **Tutor Prolog- idealização, projeto e desenvolvimento.** Lisboa: Universidade de Coimbra, 1990, Tese de Doutorado.

UNA EXPERIENCIA DE DISEÑO Y PROGRAMACIÓN DE SOFTWARE PARA EDUCACIÓN MUSICAL

PATRICIA S. SAN MARTÍN
Instituto Rosario de Investigaciones
en Ciencias de la Educación (I.R.I.C.E.)
Blvd. 27 de febrero 210 bis - 2000 - Rosario - República Argentina
Tel.: (54-41) 82-1769/-1770 Fax: (54-41) 821772

ABSTRACT

"TOBOGÁN" Y "TREN AL SUR" son dos programas para educación musical inicial, adecuados a los contenidos curriculares de la educación escolar inicial y primaria de Argentina.
Las pautas de desarrollo consideraron la simplicidad en el manejo de la computadora: los programas permiten al niño interactuar con ella, a partir de la emisión vocal como acción musical, estimulando una correcta emisión con distintas intensidades y dicción de vocales. Utilizando un micrófono y el mouse, un entorno lúdico le permite abordar el aprendizaje de nociones del lenguaje musical, como la duración y la variación de altura del sonido graduando las dificultades a través de mapas de recorrido que él puede crear.
El programa REPORTE, permite al docente plantear más adecuadamente sus estrategias de enseñanza. El proyecto previó la utilización del software no como un fin en sí mismo, sino como una herramienta para optimizar la construcción del proceso de aprendizaje musical.

INTRODUCCIÓN

La computadora como generadora de sonido al igual que los instrumentos musicales electroacústicos cuentan con varias décadas de historia en las expresiones musicales de la segunda mitad del s. XX, sin embargo su utilización en la escuela como herramientas para la educación musical es prácticamente nula y los desarrollos metodológicos donde se utilicen estos medios son escasos e incipientes.

Las problemáticas de costo de equipamiento, la falta de comprensión de los docentes de computación y/o informática educativa sobre terminología básica de los programas utilizables para educación musical y la no adecuación de los planes de estudio de los futuros educadores musicales a las posibilidades tecnológicas actuales fueron variables importantes para determinar esta realidad. La reducción del costo hoy en día hace posible la adquisición de este equipamiento por parte de las escuelas pero la subutilización de estas tecnologías persiste ya que no se establecen las relaciones interdisciplinarias pertinentes entre los docentes del área tecnología y los de educación musical. La capacitación docente continua que contemple esta problemática dentro de la Educación General Básica (E.G.B.) de la Argentina estaría prevista a partir de 1995 según lo dispuesto en la nueva Ley Federal de Educación.

La interrelación de estas áreas es insoslayable para que el alumno comprenda integralmente conceptos básicos sobre nuestra cultura posmoderna y esté motivado para abordar la utilización de la tecnología digital en forma exploratoria y creativa experimentando diversas formas de comunicación expresiva.

El desarrollo de un marco teórico didáctico configurado en el ámbito de la E.G.B. donde se construyan procesos de enseñanza-aprendizaje de la música estableciendo una relación analítica con los medios tecnológicos constituye el objetivo del estudio investigativo que realicé en el Instituto Rosario de

Investigaciones en Ciencias de la Educación (IRICE) durante los años 1992/93/94. Dicho proyecto se constituyó en tres etapas.

Primera Etapa: año 1992

1) La función de la Educación Musical en la escuela.
2) Las posibilidades que la computadora aporta al proceso de enseñanza-aprendizaje de la música.
3) La modalidad de taller en la educación musical escolar.
4) Relevamiento de experiencias de educación musical que incorporen la modalidad de taller con el uso de computadoras e instrumentos electroacústicos. Entrevistas a los docentes responsables de la actividad en las ciudades de Buenos Aires y Rosario.

Segunda Etapa: año 1993

1) Relevamiento de software: análisis clasificatorio y evaluativo a partir del enfoque psicopedagógico propuesto de productos que se ofrecen en el mercado comercial para "Música".
2) Planificación y realización de un curso teórico/ práctico de capacitación para educadores musicales sobre la integración de la computadora y los instrumentos musicales digitales a la educación musical escolar.
3) Desarrollo de pautas de diseño y programación de software para educación musical.

Tercera Etapa: año 1994

1) Diseño de los programas "Tobogán" y "Tren al Sur"
2) Programación de "Tobogán"
3) Optimización del programa a partir de su operación en la escuela.
4) Reelaboración del diseño de "Tren al Sur". Programación .

Desde mediados de 1993 se integró al estudio Javier Santoro realizando una pasantía en IRICE sobre "Posibilidades de programación de las placas de sonido Sound Blaster" y desarrollando la programación de Tobogán y Tren al Sur.

PAUTAS DE DESARROLLO

El análisis crítico realizado en la Segunda Etapa de la investigación conjuntamente con las conclusiones e intercambios realizados con los docentes que asistieron al curso de capacitación resultaron significativos para la elaboración de las siguientes pautas de desarrollo:

* Crear software que se adecuen a los contenidos programáticos que se plantean en el nivel inicial y primario de escolaridad en Argentina.
* Investigar posibilidades de programación de las placas de sonido Sound Blaster.
* Acotar la temática de contenido musical como metodología de desarrollo ya que era la primera experiencia en el tema.
* Lograr máxima simplicidad en su operación configurándose el programa como un medio facilitador de nivel inicial para familiarizarse con la computadora.
* Interactivo a partir de la emisión vocal como acción musical, revalorizando la voz como instrumento de ejecución, estimulando una correcta emisión con distintas intensidades y dicción de vocales.
* Entorno lúdico e iconográfico.
* Que el programa no sea un fin en sí mismo sino una herramienta para optimizar la construcción del proceso de enseñanza- aprendizaje musical.

DISEÑO Y PROGRAMACIÓN

"TOBOGÁN" y "TREN AL SUR" son dos programas para Educación Musical que se configuran como herramientas que se pueden utilizar con niños a partir de los cuatro años. Se constituyen como juegos que plantean problemáticas básicas para el aprendizaje de nociones iniciales del lenguaje musical como la duración (campo analógico y métrico), la intensidad y la variación de altura del sonido en forma escalar y continua (glissando).

El niño interactúa con la computadora utilizando un micrófono y realizando sencillas operaciones con el mouse. Todos los juegos tienen una demostración y pantallas de ayuda además de las explicaciones que se adjuntan para el adulto en el manual.txt. Las indicaciones generales para el niño, las demostraciones y los mensajes que indican los posibles errores están grabados por la voz de una niña de cinco años constituyéndose en una guía auxiliar para el desempeño de los juegos.

El docente cuenta con un programa REPORTE donde dispone de información sobre el desempeño de juego de cada niño con fecha y número de veces que entró al juego. Además se puede consignar la edad, el grado o nivel de escolaridad y si sabe leer música. Puede archivar y/o imprimir toda la información según sus necesidades.

En el Reporte la totalidad de los errores posibles en cada juego, pantalla o mapa según corresponda se encuentran discriminados en tablas. Los porcentajes de error que se consignan al final de cada fila surgen de un criterio que se estableció a partir de cada problemática planteada y se explican detalladamente en el manual. Se desea destacar que no se deben entender como un score de Videogame sino que el docente debe considerarlo como un dato que le permite adecuar más claramente sus estrategias de enseñanza.

El hecho de ubicar al Reporte como un programa independiente se fundamenta en que éste es un instrumento de evaluación que como tal debe ser interpretado por el docente no teniendo sentido su comprensión por parte del niño.

Tobogán:

Consta de 3 juegos. El escenario es una plaza donde un "personaje" juega de distintas maneras animado por la voz del niño. Para jugar podrá cantar cualquier vocal o sílaba que estimule una emisión no aireada del sonido.

Consideramos al primer juego como un aprestamiento a la utilización de esta tecnología con respecto a las habilidades que requiere Tobogán. Los otros dos plantean una problemática musical más compleja. Superada la dificultad del primer juego, se pueden abordar indistintamente los otros de acuerdo al nivel de maduración del niño y al contenido musical que el docente desee trabajar.

Primer Juego:

Contenido: Sonidos vocálicos largos, cortos y silencios en campo analógico. Sincronización de la emisión sonora con la orden gestual propia.
Acción: El "personaje" debe realizar un recorrido con su patineta. Al llegar a los "pozos" deberá saltarlos para completar su recorrido.

Figura I:

Consigna: Marcar con el botón izquierdo del mouse sobre el personaje e **inmediatamente** emitir sonido vocálico utilizando el micrófono. Cortar el sonido cuando la punta de la patineta llega al borde del pozo. A partir de allí hacer silencio hasta que la patineta vuelve a apoyar **todas** las ruedas sobre el piso e **inmediatamente** volver a emitir sonido hasta llegar al próximo pozo y así sucesivamente hasta terminar la pantalla.

En este juego aparecen tres pozos por pantalla ubicados por el programa en forma aleatoria, lo que permite distintas longitudes de recorrido (duraciones). Sólo se debe hacer click con el mouse sobre el "personaje" al comienzo de cada pantalla.

Mensajes de error: Si se escucha...*Tropezó en el camino*...se visualiza una animación acorde al mensaje en el sitio donde se cometió el error.

El Reporte registrará las siguientes posibilidades:
1) demoró más de 1" en emitir sonido luego de hacer click sobre el personaje.
2) emitió un sonido más corto que el indicado.
3) hizo silencio cuando correspondía sonido.

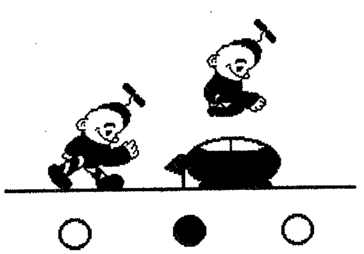
...*Casi cae al pozo:* el sonido fue más largo o emitió sonido cuando correspondía silencio.

Luego del error se inicia una nueva pantalla. Si durante el tiempo que el niño juega logra completar dos pantallas sin errores aunque sean no consecutivas aparece un mensaje audiovisual de estímulo; si continúa jugando y logra completar una más aparece una nueva pantalla indicando su logro total (tres). Consideramos que si el niño puede realizar esto tanto en este juego como en el siguiente el objetivo que plantea la actividad está logrado.

Segundo Juego:

Contenido: Sonidos largos, cortos y silencios a partir de la unidad en campo analógico métrico proporcional. Velocidad lenta y rápida.
Acción: El personaje camina o corre (según la opción elegida) y debe saltar grupos de niños.
Bajo el eje horizontal se encuentra la marca de pulsación audiovisual (16 pulsos por pantalla).

Figura II:



Consigna: Hacer click con el botón izquierdo del mouse sobre el personaje. Esperar en silencio escuchando la pulsación hasta llegue al obstáculo. Cantar la duración correspondiente y luego hacer silencio hasta el próximo agrupamiento y así sucesivamente.

La duración del sonido dependerá del número de niños acurrucados que deba saltar, asociándose los mismos a cantidad de pulsos completos. Los grupos se constituyen en cada pantalla aleatoriamente con componentes de uno a cuatro privilegiándose en la programación los de dos y tres. Sólo se debe hacer click sobre el personaje al comienzo de cada pantalla.

La marca auditiva reproduce la melodía de presentación del programa pero su ritmo fue variado a duraciones iguales. En este juego se planeó una inversión con respecto al primero en cuanto el alumno al comenzar la pantalla da la orden con el mouse pero debe permanecer en **silencio** escuchando la velocidad de pulsación y observando en qué momento debe comenzar a emitir.

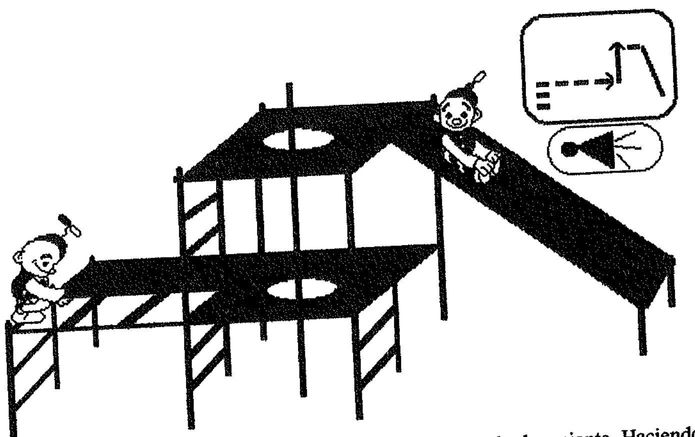
Los mensajes de error fueron previstos de forma semejante a los del primer juego.

Tercer Juego:

Contenido: Variación de altura en forma escalar o continua analógica. Sonido igual corto o largo en campo analógico no métrico.
Acción: El personaje juega en un complejo tobogán siguiendo un recorrido.

Según la opción elegida el alumno cantará los mapas presentados o creará los propios. En el diseño se previó que la variación de altura escalar no requiera afinación temperada. El fundamento didáctico es que el alumno debe conceptualizar la idea de variación de altura y dirección de la misma más allá de la posibilidad de afinación de relaciones de alturas determinadas, estas son dos problemáticas diferentes que en general no son trabajadas correctamente, en varias oportunidades nos encontramos con alumnos que cantan correctamente canciones o intervalos pero confunden la direccionalidad y son incapaces de escribir un diseño melódico simple. También para niños con problemas de afinación este juego puede ser positivo para abordar este aspecto.

Figura III:



Juega con Mapa: se diseñaron ocho mapas distintos ordenados en dificultad creciente. Haciendo click con el mouse sobre el mismo se visualizan los distintos recorridos.

Consigna: Para iniciar el juego marcar con el botón izquierdo del mouse sobre el personaje y luego emitir sonido vocálico siguiendo las instrucciones del camino que marca el mapa.

En los cuatro primeros el personaje realiza algunas porciones del mapa automáticamente. Estos recorridos están asociados a la simplificación de cada mapa con respecto a la secuencia de combinaciones de variación de altura y a la imposibilidad de continuar al no lograr la repetición de la última altura cantada.
Los mensajes de error dicen: *Se Quedó Corto...*, *Más Agudo...*, (cantado sobre un acorde mayor ascendente), *Más Grave...* (cantado con intervalos descendentes), *Igual...* (dicho sin variar la altura).
Se diseñó el ícono "Bocina" para poder escuchar luego del error el último sonido emitido y así referenciar el siguiente. El juego puede continuar desde ese punto, reiniciar el mapa o cambiarlo. Una vez concluido el mapa haciendo click sobre el personaje se lo puede repetir.

Dibuja el Mapa: Se puede elegir el camino ha realizar a partir de las opciones que se indican en pantalla. Estas opciones siempre van prefigurando un recorrido de izquierda a derecha como una forma de convención de lectura de código. Al marcar sobre las elegidas se va dibujando el mapa en el lateral superior derecho. Si se desea borrar el mapa que se está dibujando se lo puede marcar con el botón izquierdo del mouse y reiniciar la operación. Concluido el mapa se proseguirá jugando de igual forma que en la opción anterior. Aquí no hay recorridos automáticos y las posibles combinaciones pueden llegar a resultar suficientemente complejas para favorecer la construcción del concepto a partir de la creación de un recorrido y su ejecución.

La rutina que se desarrolló para el reconocimiento de la variación de altura (utilizada en ambos programas) tuvo en cuenta las características tímbricas de la voz con respecto a las sílabas con distintas vocales y los rangos de frecuencia de la voz humana, ajustándose a tesituras medias más afines con las posibilidades de un niño. Se adoptó un sistema de lectura original ya que la lectura del corte por cero de la amplitud no era aplicable a este caso. Al no tomarse un standard la compatibilidad con otras tarjetas de sonido no pudo ser posible hasta el momento pero se continúa el estudio. El lenguaje de programación utilizado fue "TURBO C".

Para la programación fue importante lograr la velocidad de detección del error y una dinámica operatoria ágil con el mínimo requerimiento de hardware posible teniendo en cuenta la edad de los niños y las realidades de equipamiento de las escuelas. El ajuste entre la realidad perceptiva del niño y la programación en cuanto a los márgenes de error fue hecho a partir de la observación en campo escolar, la detección del igual no es en relación a una frecuencia sino al conjunto de frecuencias que psicoacústicamente percibimos como altura igual, además en las pruebas de optimización se detectaron problemas de lectura al utilizar micrófonos de baja calidad siendo esto un factor extrínseco pero de gran peso para la operación de estos programas.

Considerando la realidad acústica del ámbito escolar, se incorporó un programa para la configuración del micrófono a entornos sonoros cambiantes, con esta prevención se evitan errores de discriminación por la perturbación de ruidos externos.

Tren al Sur:

Fue programado y rediseñado a partir de la experiencia de "TOBOGÁN". Este programa está pensado desde una unidad temática: "El Tren", integrándose los contenidos sonoros a desarrollar en distintas pantallas que significan diferentes momentos de un recorrido a realizar.

Desde una pantalla general de configuración el programa permite graduar gráficamente la calidad del espectro del sonido requerido al cantar cualquier vocal, a esto lo denominamos "Sensibilidad al ruido". También es posible entrenar y seleccionar vocales que mediante el programa de reconocimiento permitirán resolver algunas instancias del juego, al programa de reconocimiento de vocales se le puede indicar en forma gráfica el margen de error. Un vúmetro permite visualizar los niveles de intensidad para hacer un ajuste correcto de la emisión ya que la saturación ocasiona errores de reconocimiento. El entrenamiento se puede archivar bajo el nombre del niño, lo que permite no entrenar cada vez y establecer una dinámica grupal ágil dentro del aula. En el programa Reporte se accede a la lista de entrenamientos pudiendo borrar uno o todos.

Los mensajes de error sonoros son acompañados de una simbología específica al caso que aparece en un recuadro pequeño. En los casos de reconocimiento de vocal ese recuadro habilita a escuchar la vocal grabada como referencia y luego de la emisión indica si no fue esa, si se dijo más aguda o más grave, si fue ruido o si saturó.

Acción: El tren con la locomotora en espera cargará el cereal y luego subirán pasajeros. A partir de un mapa recorrerá planicies, donde a veces se cruza una vaca, puentes de distinta longitud, zonas de inundaciones, pasos a nivel donde están cruzando autos, zonas de montaña para llegar finalmente a destino.

La configuración del recorrido queda a criterio del docente y/o del niño según lo que se plantee en el proceso didáctico-musical. La elección de los elementos constitutivos del mapa (uno o dos puentes, pasos a nivel, zonas de inundación, montañas) está condicionada al grado de dificultad que plantea cada uno y al tiempo de atención del niño para realizar un recorrido completo. Las planicies son puestas por el programa en función del mapa elegido al igual que el factor aleatorio "vaca". El niño puede entrar a partir que inicia el recorrido al "Mapa" y observar las sucesivas posiciones del tren según la evolución del juego. Este mapa es distinto en su concepción al del "TOBOGÁN", se asimila a la imagen de un mapa geográfico y no a una partitura analógica de variación de altura. Es posible saltear pantallas aunque estas se hallan elegido en la configuración, consideramos que abandonar una pantalla sin concluir o modificar el recorrido en su transcurso puede ser útil a la dinámica de juego que establece cada niño. El docente tendrá registrado esto en el Reporte siendo un dato importante para el planteo de sus estrategias educativas.

Desde las pantallas de juego se puede acceder a la configuración general y modificar los datos consignados de acuerdo a la realidad del niño en ese momento. El juego permite reiniciar cada pantalla cuantas veces se desee si se ha cometido un error, cuando está lograda pasa automáticamente a la siguiente. Finalizado el recorrido en la estación se informará si el tren llegó a "horario" o "con atraso". Llegar a horario significa haber concluido el 70% de las pantallas elegidas independientemente de cuantas veces haya tenido que repetirla para lograrla.

Contenidos:

*Intensidad: se visualiza su variación permanentemente a partir del tamaño del humo de la locomotora. Los cambios de intensidad afectan en algunos recorridos la velocidad de desplazamiento del tren. Esto fue diseñado para el control de la duración del soplo de acuerdo a la intensidad con que se emite.

*Duración: planicies- sonido largo (nivel más simple). Planicies con vaca- sonido semilargo, corto, muy largo. Puentes de tres y de cinco emisiones: sonidos de igual duración secuenciados. Zona inundada de nueve emisiones: sonidos cortos secuenciados (difícil). Vía Libre- sonidos muy largos (7" de máximo).

*Variación de Altura: cargar cereal (bajar), acción simple en forma de intervalo descendente o en glissando descendente. Bajar montaña, glissando lento descendente (4" de duración). Subir pasajeros y subir montaña, ídem anterior pero con cambio de dirección.

*Dicción de Vocales: salida del tren, dicción de la vocal elegida. Paso a Nivel, dicción de 2 vocales elegidas para lograr que los autos crucen y el tren tenga Vía libre.

*Sincronización de la Emisión: planicies, puentes, inundación. Emisión inmediata luego de cada click.

*Calidad de la Emisión: se controla en todas las pantallas.

Requerimientos del Sistema:

* Computadora 386/ 486 IBM o Compatible
* MSDOS
* 640 kb. (mínimo 240 kb. para Tobogán)
* Memoria Expandida (mínimo 770100 Bytes: Tren al Sur y mínimo 1 Mbytes para Tobogán)
* Disco Rígido: Tobogán ocupa 3 Mb. y Tren al Sur necesita como mínimo 7.210.000 Bytes.
* Tarjeta de Sonido Sound Blaster (8 o 16 bit en cualquiera de sus tipos)
* Micrófono para voz
* Parlantes
* Mouse
* Tarjeta de Video VGA
* Monitor color

CONCLUSIÓN

Los resultados obtenidos fueron alentadores para continuar la producción de programas dando cuenta del interés que despiertan en los niños y la necesidad de los docentes de contar con **Software para la Educación Musical** apropiados a la realidad escolar latinoamericana.

Expert Piano: Um Ambiente Educacional para Auxiliar o Estudo de Piano e Música

José Honório Glanzmann
Neide Santos
Ana Regina Rocha
COPPE/Sistemas
Universidade Federal do Rio de Janeiro - UFRJ
Caixa Postal 68.511 - CEP - 21.945-970 - Rio de Janeiro - RJ
E-mail: dede@cos.ufrj.br - Fax: (0511) (021) 290-6626

Abstract

Expert Piano is an educational environment composed by an intelligent tutoring system, multimedia databases and MIDI (Musical Instrument Digital Interface) interface devices. Our environment aims to improve the piano playing techniques, supplying different piano studies options, such as to practice the whole musical piece or specific parts or to play with software cooperation. The environment also provides measure movement, tones transposition, practice with separated hands, score visualization simultaneously with the music execution or performance.

The intelligent tutoring system consists of a music teachers and pianists knowledge base, and its role is to diagnose the errors committed in a study session, through the correct processing of the musical events produced by the student. According to the student performance, the software issues a report, highlighting the musical pieces measures, where there is errors occurrence. Then, **Expert Piano** gives feedback visually and/or orally and addresses the student to the suitable subject. We think that the showing of traditional musical notation and the playing of correct measures execution via MIDI are most efficient kinds of feedback than only written feedback.

1 - Introdução

Nas últimas décadas observamos a adoção de tecnologia computacional nas mais diversas áreas do conhecimento humano, e com a música não poderia ser diferente. Hoje, podemos constatar a grande utilização de computadores e outras tecnologias apoiando e melhorando a qualidade da produção musical.

Yavelow (1989) divide as aplicações músico-computacionais em áreas que estão diretamente ligadas à produção musical (laboratório de som, composição, edição de partituras, performance e edição de sons pós-produção), e outras como a *educação musical*, que não estão envolvidas diretamente com o primeiro grupo. O que nós podemos observar na prática e em bibliografias especializadas é que com o barateamento do custo, microcomputadores, sintetizadores e outros equipamentos estão sendo cada vez mais incorporados à educação musical (Rudolph, 1991; Holton, 1991; Webster 1991). Com isso, tecnologias antes só disponíveis em grandes estúdios e laboratórios estão sendo adotadas por professores, ganhando espaço nas salas de aula tradicionais.

A área de Informática Educativa tem-se mostrado uma poderosa tecnologia de apoio a novas formas de ensino e aprendizagem (Santos & Segre 1991). Segundo Rueda (1993), a adoção de técnicas de Inteligência Artificial podem contribuir para a elaboração de produtos de software educacional mais flexíveis e apropriados às necessidades de cada aluno. Coerentes com este pensamento e aliando tecnologia músico-computacional apresentamos o ambiente educacional **EXPERT PIANO** que objetiva auxiliar o estudo de piano e música.

Um dos problemas principais enfrentados pelos professores de música é a correção dos vícios adquiridos pelo aluno quando este estuda sozinho. Se o aluno não estiver bem seguro dos conceitos teóricos e suas equivalentes interpretações no instrumento, ele poderá facilmente incidir em erros de execução musical e este comportamento repetidamente provocará a fixação destes erros, dificultando a sua correção. Neste sentido, o ideal é que o aluno estude com a presença do professor ou de outra pessoa conhecedora de música pelo menos nos primeiros anos do curso de instrumento. Como isto nem sempre é possível, nos sentimos motivados a desenvolver um produto de software educacional que assistisse o aluno de música enquanto este estivesse estudando.

O ambiente **Expert Piano** têm objetivos específicos que são voltados para o treinamento e aperfeiçoamento da técnica de instrumento, no caso o piano. O estudo de um instrumento sempre exigiu do aluno várias horas diárias de dedicação e inúmeros exercícios são necessários para que o estudante adquira a postura correta, agilidade e o desenvolvimento da coordenação psicomotora. Por esse motivo, os componentes principais do nosso ambiente são direcionados a análise de *performance,* detectando, quando for o caso, erros de execução musical e apresentando apontamentos e sugestões sobre como o estudante deve proceder para correção dos mesmos. Outros componentes complementam o ambiente educacional **Expert Piano** e vão propiciar recursos extras que normalmente não estão disponíveis em uma educação musical tradicional. O ambiente disponibiliza ao estudante informações suplementares sobre as peças musicais estudadas e seus compositores, além disso, possibilita o acompanhamento do aproveitamento do aluno.

2 - Características Gerais e Elementos Envolvidos no Ambiente *Expert Piano*

O ambiente **Expert Piano** tenta simular o comportamento de um professor que esteja assistindo à interpretação do aluno, analisando permenentemente sua *performance*. Para isso, ele propicia recursos comuns ao ensino de piano e incorpora características específicas de um sistema computacional.

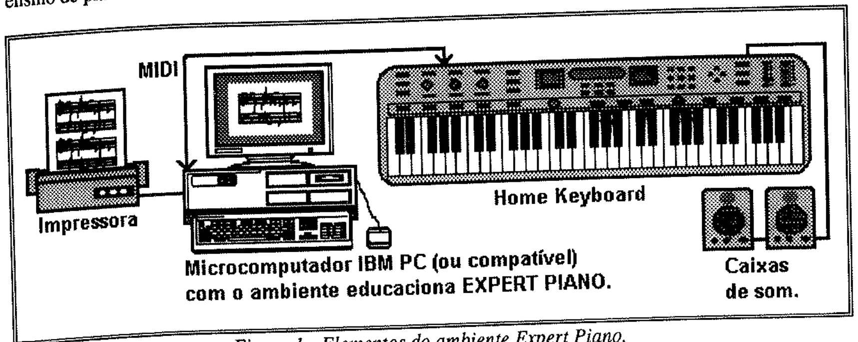


Figura 1 - Elementos do ambiente Expert Piano.

Como podemos verificar na figura 1, existem vários elementos interrelacionados no ambiente **Expert Piano**:

a) *Home Keyboard* : trata-se do instrumento eletrônico que servirá de interface musical podendo ser utilizar também um sintetizador ou piano eletrônico. O instrumento em questão deverá suportar o padrão MIDI para possibilitar a conecção com o microcomputador.

b) MIDI : refere-se à interface responsável pela comunicação entre o microcomputador e o instrumento musical . É recomendado a utilização de um cabo de boa qualidade para que não haja perda de dados na transmissão. Serão necessários a utilização dos conectores *IN* e *OUT* em um esquema de interconecção bidirecional (Loy, 1985).

c) Microcomputador : deve ser compatível com o IBM PC e possuir características de *hardware* que possibilitem um bom desempenho no ambiente *MicroSoft Windows*. Além disso, é necessário uma placa de som com interface MIDI embutida ou outro dispositivo equivalente.

d) Impressora : pode ser matricial ou laser, lembrando-se que a qualidade do material impresso dependerá da impressora utilizada.

3 - Descrição do Ambiente Educacional *Expert Piano*

O ambiente **Expert Piano** tem como público alvo alunos do curso de instrumento, normalmente matriculados em escolas de música e conservatórios. A utilização do produto deverá acontecer paralelamente às aulas normais ministradas pelos professores da escola.

A qualidade da música produzida e a interface computacional desenvolvida foram planejadas para que o aluno não só se sinta motivado a utilizar o ambiente, como também o faça de maneira amigável. Levando-se em consideração que o aluno na maioria das vezes utiliza o ambiente sozinho, é natural e desejável que a interação eles seja fácil e intuitiva.

Para o funcionamento do sistema, é necessário a realização de alguns procedimentos preliminares como o cadastramento dos alunos, das peças musicais e do plano de estudo.

Para o aluno poder utilizar o ambiente educacional, ele precisa ser cadastrado previamente pelo professor e a partir disso, o ambiente pode controlar o seu desempenho e registrar informações específicas que nortearão as futuras remediações e sugestões de estudo apresentadas.

Quando o produto é instalado, a base de peças musicais está vazia, por isso, os professores devem também alimentar o sistema com as músicas e exercícios a serem estudados. O ambiente possui recursos de um seqüenciador, permitindo que o professor introduza peças musicais segundo o currículo da sua escola. Isto, facilita a adequação do ambiente à realidade de cada escola.

O ambiente ainda permite ao professor direcionar o estudo do seu aluno, criando para isso um plano de estudo. Este plano de estudo contém um conjunto das opções a serem disponibilizadas pelo sistema durante a utilização do aluno para uma determinada peça musical. A partir deste plano, o professor apontará os passos a serem seguidos pelo estudante durante uma interação com o ambiente.

Satisfeitos estes procedimentos iniciais, o ambiente estará pronto para ser utilizado. A seguir descreveremos como é realizada uma sessão de estudo com o ambiente educacional **Expert Piano.**

3.1 - Uma sessão de estudo

Uma vez identificado, o aluno abrirá uma sessão de estudo e o ambiente passará a registrar os passos efetuados pelo mesmo. Durante a sessão, o aluno poderá escolher uma ou mais peças musicais a serem estudadas, dentre as disponíveis na base de músicas. O aluno estudará segundo critérios definidos previamente pelo plano de estudo. Estes critérios irão direcionar o estudo e vão influenciar as opções oferecidas para o estudante.

Uma sessão de estudo compreende várias etapas, a saber:

- escolha das opções de estudo;
- diagnóstico e análise de erros;
- apresentação de remediações e sugestões;
- relatórios de desempenho;
- acesso à informações complementares, e,
- conclusão do processo.

3.1.1 - Opções de Estudo

O ambiente admite várias opções de estudo de uma mesma peça musical, tais como :

- tocar a peça musical escolhida com níveis diferenciados de detalhes de expressão e com acompanhamento do sistema (mão alternada, mão oitavada, outras vozes ou outros instrumentos),

e,

- escutar a peça musical original ou a própria execução com visualização paralela da partitura.

As funções acima ainda dispõe dos seguintes recursos :

- exercitar a peça musical toda ou trechos selecionados;
- estudar com as mãos juntas ou separadas;
- transposição de tons;
- mudança de andamento, e,
- auxílio de um metrônomo.

3.1.2 - Diagnóstico e análise de erros

Depois do aluno concluir a execução da peça musical escolhida, o sistema está capacitado a apontar os erros cometidos, relacionando os compassos correspondentes. Neste processo, o ambiente apresenta os compassos errados (executados pelo aluno) e os corretos (armazenados na base de dados de peças musicais), a fim de que o aluno possa melhor visualizar e comparar as falhas de interpretação ocorridas. Esta é a principal função do sistema, e é neste ponto que as técnicas de Inteligência Artificial são necessárias para o processamento adequado do comportamento do estudante..

O ambiente **Expert Piano** contém um componente que incorpora as principais características e funcionalidades de um Sistema Tutorial Inteligente (STI). Ele absorve em sua base de conhecimentos "dicas" de especialistas (professores de piano e pianistas), que são utilizadas no diagnóstico e na análise dos erros do aluno.

3.1.3 - Remediações e sugestões

O componente STI também faz as remediações e apresenta sugestões para a correção dos erros que o aluno venha a cometer. Para cada erro diagnosticado e de acordo com os registros das *performances* anteriores do estudante e as características da peça em questão, o ambiente apresenta ao aluno informações apropriadas que deverão auxiliá-lo na compreensão do seu desempenho. Para isto, o ambiente utiliza recursos visuais (notação musical tradicional e textos) e auditivos (execução via MIDI dos compassos musicais em questão).

3.1.4 - Saída dos Relatórios de Desempenhos

O ambiente gera relatórios sobre o desempenho do estudante, tanto na tela quanto na impressora. Este relatório traz informações sobre os erros e apresenta os compassos originais (corretos) e os executados pelo aluno. Este relatório pode ser *customizado* pelo aluno que poderá selecionar as informações que ele deseja ter acesso. Estas informações ficam armazenadas no ambiente e podem ser acessadas também pelo professor.

3.1.5 - Informações complementares

Além dos dados fornecidos nos processos de diagnóstico e remediação, o ambiente educacional dispõe de informações complementares em formato multimídia, com ligações tipo hipertexto. Estas informações compreendem dados sobre a vida e a obra de compositores, peculiaridades sobre as peças musicais e dicionário de termos musicais incluindo o significado de símbolos da notação musical tradicional.

Os professores podem, dependendo do nível de recursos pretendidos, fazer a manutenção também desta base de dados.

3.1.6 - Conclusão do processo

Ciente dos resultados obtidos, o aluno pode reiniciar o processo de estudo, modificando opções segundo o seu desejo e permissão do plano de estudos. No caso de reincidência de erros, o ambiente **Expert Piano** remeterá o aluno a novas remediações e sugestões. Ao final de uma sessão de estudo, o aluno pode solicitar ao ambiente "dicas" para estudar em casa, no seu próprio instrumento.

Se o aluno não cometer erros após o exercício, o plano de estudos indicará novas opções de estudo até que a peça seja corretamente tocada, com todos os detalhes de interpretação pertinentes a música em questão.

4 - Estrutura do ambiente *Expert Piano*

A fim de atender as funções acima mencionadas, apresentamos e descrevemos a estrutura do ambiente **Expert Piano** (figura 2) com os seus diversos módulos e principais características. A estrutura reflete a organização dos módulos e as interligações necessárias para que as especificidades do problema proposto possam ser solucionadas.

4.1 - Módulo *Interface*

O módulo *Interface* é responsável pela comunicação entre os usuários (alunos, pianistas e professores) e ambiente educacional. Ele controla o fluxo de informações entre os dois, organizando e interpretando as ntradas e saídas de dados e ações. O módulo também é encarregado da integração entre os demais módulos lo ambiente educacional. Ele responde ainda pela segurança de acesso ao sistema e a determinados módulos, ıão permitindo a utilização por pessoas não autorizadas.

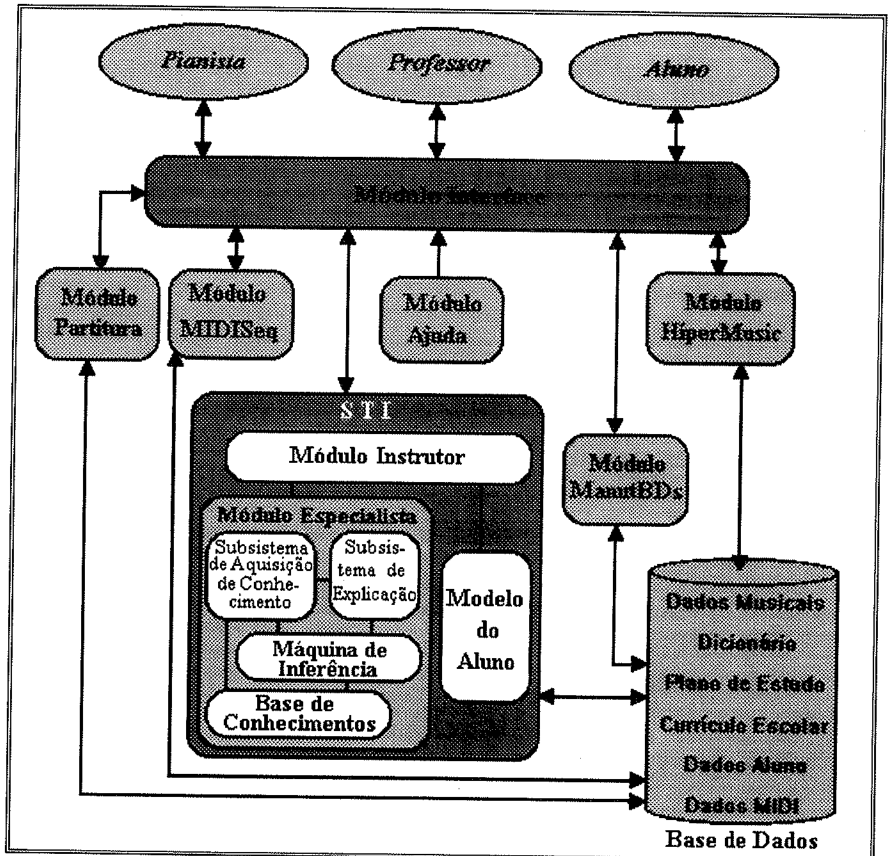


Figura 2 - Estrutura do Ambiente Expert Piano.

4.2 - Módulo *HiperMusic*

O módulo *HiperMusic* gerencia o acesso a uma base de dados multimídia, fornecendo ao estudante nformações complementares que o auxiliarão na sua formação musical, além de tornarem o ambiente mais e ico e interativo. Esta base de dados compreende as seguintes informações:

- vida e obra dos compositores (particularmente os adotados no currículo escolar);
- dicionário de termos musicais;
- significado de símbolos musicais (notação tradicional, etc);

- particularidades sobre as peças musicais estudadas, e,
- anotações de professores e outros alunos sobre determinadas peças musicais.

4.3 - Módulo *ManutBDs*

Este módulo tem a função de manutenção das bases de dados do ambiente **Expert Piano**. É de responsabilidade do professor a atualização dos dados cadastrais dos alunos, bem como dos dados sobre o plano de estudos e currículo escolar. Os outros componentes da base de dados (dados MIDI, dados musicais e dicionário) também são mantidos a partir deste módulo, só que devido a natureza de suas informações, as manutenções são realizadas sob orientação e controle especiais.

4.4 - Módulo *MIDISeq*

Toda a comunicação entre o microcomputador e o sistema MIDI é realizado por este módulo. Através do módulo *MIDISeq* o usuário pode monitorar e controlar o seu equipamento MIDI, verificando as capacidades e possíveis ocorrência de falha dos equipamentos MIDIs disponíveis.

O módulo incorpora as funções básicas de um seqüenciador, provendo os principais recursos para gravação e reprodução de peças musicais em formato MIDI. Durante uma sessão de estudo, a interação do aluno com os recursos MIDI serão providenciadas pelo módulo *MIDISeq*, sob supervisão do módulo Interface.

4.5 - Módulo *Partitura*

O módulo *Partitura* é responsável pelo tratamento dos dados musicais em formato de notação musical convencional. Ele possibilita a visualização na tela ou impressora das partituras completas ou dos trechos selecionados pelos usuários. O usuário pode, ainda, optar pela impressão da peça musical original ou pela peça executada por ele.

Este módulo é muito importante para a comunicação entre o ambiente e o aluno, pois quando necessárias as remediações do sistema são feitas através de notas e compassos musicais. Isto certamente facilita o entendimento do aluno no que se refere aos eventos musicais realizados.

4.6 - Módulo *Ajuda*

O ambiente proporciona um recurso adicional de auxílio às dúvidas de operação e manuseio do sistema. O módulo *Ajuda* é sensível ao contexto e contém explicações sobre as diversas opções e modos de operação do ambiente **Expert Piano**. Este módulo provê ligações tipo hipertexto que remetem o usuário a outros contextos. Ainda estão disponíveis outros recursos como localização de palavras, histórico de tópicos já percorridos e consultas a um índice alfabético de palavras chaves.

4.7 - Módulo *Sistema Tutorial Inteligente*

O módulo Sistema Tutorial Inteligente é o principal dos módulos do ambiente **Expert Piano**. Ele é responsável pelo diagnóstico e tratamento dos erros cometidos pelo aluno, e também tem a finalidade de remeter o estudante às remediações e sugestões apropriadas contidas na base de conhecimento especialista. Este módulo é subdividido em vários outros a saber : Módulo Instrutor, Módulo Especialista e Módulo Modelo do Aluno.

4.7.1 - Módulo *Instrutor*

Nos STIs tradicionais, o módulo Instrutor (ou módulo Tutor) é o responsável pela escolha e apresentação do material educacional a ser empregado para um determinado estudante, analisando também o seu rendimento. Nosso ambiente, como foi dito anteriormente, visa um trabalho conjunto com o professor, sendo assim, a escolha das peças musicais a serem estudadas com o auxílio do **Expert Piano** deverá ser feita por ele.

No momento em que uma peça musical foi escolhida, os conceitos teóricos e práticos necessários à sua interpretação são conseqüentemente também selecionados. Neste sentido, o Módulo Instrutor do ambiente educacional tem as seguintes funções:

- orientar o exercício do aluno segundo o plano de estudos e diretrizes do professor: as várias opções de estudo durante uma sessão são disponibilizadas a partir do plano de estudos traçado pelo professor. Na falta do plano específico o aluno pode utilizar o plano de estudos padrão fornecido pelo ambiente ou estudar segundo os seus próprios critérios;

- processar os eventos MIDI gerados pelo aluno durante o exercício : através de comunicação com o módulo MIDISeq a *performance* do aluno será gravada em formato MIDI, possibilitando depois o tratamento dos eventos musicais;
- apresentar ao aluno as remediações e sugestões apropriadas: através de ligações com os módulos MIDISeq e Partitura o ambiente proporciona *feedback* ao aluno, inclusive com recursos visuais e auditivos;
- analisar e acompanhar o rendimento do aluno: o módulo registra o desempenho do estudante e alimenta o Modelo do Aluno;

4.7.2 - Módulo *Especialista*

O Módulo *Especialista* é um sistema especialista que contém o conhecimento dos especialistas musicais (professores e pianistas) e o mecanismo que manipula adequadamente este conhecimento. O conhecimento armazenado precisa ser suficiente para as duas funções : diagnóstico dos erros e elaboração de remediações, sugestões e explanações relativas ao desempenho dos alunos.

Este módulo é composto por :

a) ***Subsistema de Aquisição de Conhecimento*** : permite a inclusão e modificação das informações contidas na base de conhecimentos. A base de conhecimento poderá ser ampliada ou alterada segundo critérios adotados pelos professores da escola, o que possibilitará a adequação do modo de agir do ambiente segundo a filosofia de ensino de cada escola. A utilização do subsistema de aquisição de conhecimento será admitida somente por professores de música e pianistas com reconhecida autoridade no assunto ou por um Engenheiro do Conhecimento (interpretador entre o especialista e o sistema computacional).

b) ***Subsistema de Explicação*** : através deste módulo os usuários do ambiente **Expert Piano** poderão acompanhar o raciocínio adotado pelo sistema para um determinado diagnóstico de erro ou remediação. Caso queira, o aluno poderá indagar ao ambiente quais foram os motivos que o levaram a prescrever uma ação especificamente, ou seja, os passos seguidos para o processamento e solução de um evento em questão. Isto, contribui para que aumente a confiabilidade dos diagnósticos, propostas e sugestões do ambiente.

c) ***Máquina de Inferência*** : trabalha com a base de conhecimentos e é formada por um conjunto combinado de métodos de raciocínio e mecanismos de inferências, que levam a solução dos problemas de forma correta e eficiente, simulando o comportamento de um especialista no domínio. No ambiente **Expert Piano**, como veremos a seguir, a base de conhecimentos contém vários formatos de dados, entre eles a representação dos eventos musicais no formato MIDI, portanto, a MI deve ser capaz de processá-los adequadamente.

d) ***Base de Conhecimentos*** : contém as regras e heurísticas coletadas dos especialistas, necessárias para o diagnóstico dos possíveis erros durante a execução de uma peça musical. A BC contém também, informações que permitem ao ambiente remeter sugestões e remediações aos estudantes. A partir do diagnóstico de um determinado erro, e levando em considerações outros fatores como por exemplo, o percurso do estudante e as características da peça em questão, o ambiente pode definir qual a melhor remediação a ser gerada para o aluno. Portanto, o conteúdo da BC do ambiente **Expert Piano** deve atender a duas necessidades básicas : conhecimento especialista (musical) para diagnóstico de erros e conhecimento especialista (pedagógico) para ensinar ao aluno como resolver estes erros.

Devido a natureza do nosso domínio, fica difícil imaginar como o ambiente educacional poderá avaliar o comportamento do aluno durante uma interpretação musical. O número de erros musicais que podem ser cometidos é bem grande, principalmente se levarmos em conta a especificidade de cada erro.

Podemos citar um exemplo simples para ilustrarmos nosso raciocínio : um aluno que em um determinado compasso deveria tocar duas notas sucessivamente, pode simplesmente executar a primeira corretamente e errar a segunda A segunda nota errada pode ser qualquer uma das 87 notas restantes, considerando um teclado de piano. Podemos verificar que o número de combinações possíveis fará crescer enormemente a nossa BC, isto levando-se em conta que no exemplo citado não analisamos outros fatores como tempo das notas, expressão, etc. Para resolvermos este tipo de problema, criamos um catálogo de erros (*bug catalog*). Para tanto, inicialmente agrupamos os principais erros segundo a sua natureza e tipo (nota errada, tempo de duração da nota diferente, etc). A estes elementos de erros chamamos de **Unidade de Erro (UE)**. A UE é o erro mais elementar que o sistema irá detectar e está normalmente associada a eventos MIDI. Devemos observar que uma UE que indica nota tocada errada servirá para todos os eventos equivalentes, independentemente da nota geradora do erro. Além disso, as informações complementares como o nome da nota responsável pelo erro não será perdida. A remediação para o aluno conterá o tipo de erro e em qual compasso ele ocorreu.

Podemos constatar que só as UEs não são suficientes para cobrir todos os erros possivelmente encontrados durante uma interpretação musical. Com o intuito de resolvermos este problema, adotamos um outro conceito que é o **Erro de Contexto (EC)**. O EC é formado por combinação de UEs e outras informações pertinentes a natureza da educação musical. Em relação a estas informações levamos em consideração as características da peça em estudo, dados do aluno, currículo da escola, plano de estudo, etc.

O EC tem prioridade sobre a UE. Entretanto, para que um EC seja pertinente, todas as suas condições tem que ser satisfeitas, caso contrário, o ambiente remete ao aluno somente as considerações relativas às UEs encontradas Vale ainda lembrar que tanto as UE quanto os EC são válidos para todas as peças musicais disponíveis no ambiente **Expert Piano**.

.O sistema de aquisição de conhecimentos é capaz de realizar a manutenção na BC, e caso o professor deseje pode modificar a BC original do ambiente **Expert Piano**. As novas regras deverão respeitar os dados modelados e já disponíveis do sistema, caso contrário, será necessário a intervenção de especialistas em informática.

Uma grande vantagem que podemos constatar com a adoção das UEs e dos ECs, é que fica factível a geração e ampliação do conhecimento especialista na BC. Podemos inicialmente testar a eficiência do sistema com um número de regras menor, e a medida que validamos o conhecimento existente, ampliamos gradativamente a BC com novas regras.

4.7.3 - *Módulo Modelo do Aluno*

O *Modelo do Aluno* visa verificar o estado de conhecimento de cada aluno, gerando hipóteses sobre suas concepções e estratégias de raciocínio. Normalmente, os STIs representam o conhecimento do estudante como um subconjunto da BC, os itens da BC recebem uma medida correspondente ao domínio do aluno sobre ele (*modelo de overlay).*

Como vimos anteriormente, a BC do **Expert Piano** é formada principalmente por um catálogo de erros musicais, neste sentido, o *Modelo do Aluno* compreende o conjunto dos erros cometidos pelo aluno, o número de ocorrência e em qual contexto eles foram identificados.

Todas as ações do aluno durante uma sessão de estudo são processadas e armazenadas no modelo do aluno e, dependendo do caso, na base de dados. É importante realçarmos que o comportamento do aluno influencia nas remediações feitas pelo sistema. O módulo Instrutor se vale das informações existentes na BC e também do que foi armazenado no Modelo do Aluno para apontar quais serão os próximos passos que o aluno deverá seguir nos próximos estudos.

Através do Módulo Modelo do Aluno podemos ainda analisar estatisticamente o desempenho do estudante e acompanharmos a evolução dele sobre erros anteriormente cometidos. Ao final do estudo de uma peça musical, o aluno deverá ter compreendido os conceitos teóricos e práticos que envolvem a peça em questão, tendo condições de interpretá-la corretamente. Isto também será registrado no Modelo do Aluno, pois trata-se de um conhecimento referente a um conteúdo dominado pelo estudante.

4.8 - *Base de Dados*

Na Base de Dados do ambiente **Expert Piano** encontramos as informações necessárias para o funcionamento dos demais módulos. Parte dos dados são representados em formato multimídia e requerem um tratamento apropriado no tocante a apresentação e manutenção. A Base de Dados basicamente compreende as seguintes informações :

a) Dados Musicais : dados sobre os compositores (sua vida e principais obras), peças musicais (particularidades e observações de professores e alunos), instrumentos musicais, conjuntos musicais e orquestras.

b) Dicionário Musical : significado dos principais termos musicais e símbolos da notação musical convencional.

c) Plano de Estudo : como as opções de estudos serão disponibilizadas para cada aluno e para cada peça musical.

d) Currículo Escolar : programa da escola contendo informações referentes às séries escolares e as suas respectivas peças musicais (exercícios e músicas).

e) Dados do Aluno : além das informações cadastrais dos alunos que utilizam o ambiente, contém os registros das ações realizadas em cada sessão de estudo e juntamente com o Modelo do Aluno fornecem subsídios para o ambiente acompanhar o desempenho dos estudantes.

f) Dados MIDI : cada peça musical disponível para estudo no ambiente **Expert Piano** tem um arquivo no formato MIDI associado. Além disso, outras informações complementam a base de dados MIDI facilitando a fase de diagnóstico. Estes arquivos são essenciais ao sistema pois são utilizados como fonte de referência para que o ambiente possa verificar o desempenho do aluno. Basicamente, o ambiente compara os eventos MIDI tocados pelos alunos com os eventos MIDI armazenados previamente pelos professores/pianistas. Assim sendo, partimos do princípio de que o que foi tocado pelos especialistas musicais está correto, e na comparação com a *performance* do aluno, o que diferenciar é considerado como erro.

5 - Conclusão

Um protótipo do ambiente educacional apresentado neste artigo está sendo desenvolvido no Programa de Engenharia de Sistemas e Computação da COPPE/UFRJ, como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências.

Para as fases de modelagem e especificação do Módulo STI, foi adotado o método KADS-estendido proposto por Werneck (Werneck, & Rocha, 1994). A implementação está sendo realizada com a ferramenta de autoria *Asymetrix Multimedia ToolBook 3.0,* que é um ambiente normalmente usado para o desenvolvimento de aplicativos de multimídia baseados no *MS-Windows.* O *Toolbook* contém uma linguagem de programação, *OpenScript* que incorpora elementos comuns às linguagens de programação tradicionais e outros próprios de linguagens orientadas para objetos. Uma vantagem na utilização do ambiente *Toolbook* e que estamos explorando no desenvolvimento desse protótipo, é a capacidade do *OpenScript* de chamar as Bibliotecas de Ligações Dinâmicas (DLLs). Através deste recurso podemos acessar as funções API do *Windows* além de rotinas desenvolvidas em C++.

6 - Referências Bibliográficas

Harmon, P. & King, D. (1988). *Sistemas Especialistas*. Rio de Janeiro, Editora Campus.

Holton, A. M. (1991). The Need for Synthesized Music in the School Curriculum. *SIGCUE Outlook*. ACM Press. Vol. 21-2.

Loy, G. (1985). Musicians Make a Standard: The MIDI Phenomenon. *in "The Music Machine - Selected Readings from Computer Musical Journal"*. Edited by Curtis Roads. The Mit Press.

Rudolph, T. (1991). Technology and Music Education. *SIGCUE Outlook*. ACM Press. Vol. 21 # 2 - Fall.

Rueda, F. (1993). ¿ Que Puede Aportar la Inteligencia Artificial al Desarrollo de La Informatica Educativa? *Revista Informática Educativa,* Vol. 6, nº 3 - Universidade de Los Andes, Santafé de Bogotá - Colômbia.

Santos, N. & Segre, L. (1991). *Informática na Educação e Incorporação das Novas Tecnologias da Informação nos Processos de Trabalho*. COPPE-Sistemas/UFRJ. Rio de Janeiro.

Webster, P. R. (1991). Music Teacher as Software Authors. *SIGCUE Outlook*. ACM Press. Vol. 21 # 2 - Fall.

Werneck, V. & Rocha, A. (1994). *Processo de Desenvolvimento para Sistemas Baseados em Conhecimento*. Publicações Técnicas - COPPE-Sistemas/UFRJ. Rio de Janeiro.

Yavelow, C. (1989). Music and Microprocessors: MIDI and the State of Art. *in "The Music Machine - Selected Readings from Computer Musical Journal"*. Edited by Curtis Roads. The Mit Press.

AQUISIÇÃO DO CONHECIMENTO EM HARMONIA: UM AMBIENTE DE APRENDIZAGEM

Luciênio de M. TEIXEIRA
UFPB - CH - DART - LIAA
Av. Aprígio Veloso s/n — 58.109-970 Campina Grande - PB
Fone: (++55) (+83) 333-1000 R.135 e_mail: lucienio@dsc.ufpb.br

Evandro de BARROS COSTA
UFAL - CCEN - MAP
Campus A.C.Simões, Tabuleiro dos Martins — 57000-000 Maceió - AL
Fone (++55) (+82) 322-2406 e_mail: evandro@dcc.ufal.br

Carlos Alan PERES DA SILVA
UFPB - CH - DART - LIAA
Av. Aprígio Veloso s/n — 58.109-970 Campina Grande - PB
Fone: (++55) (+83) 333-1000 R.135 e_mail: peres@brufpb2.bitnet

Edilson FERNEDA
UFPB - CCT - DSC
Av. Aprígio Veloso s/n — Cx.Postal 10.106 — 58.109-970 Campina Grande - PB
Fone: (++55) (+83) 333-1929 — e_mail: edilson@dsc.ufpb.br

RESUMO

O interesse maior deste trabalho é analisar as etapas de aprendizagem de um aluno artificial (máquina), seu processo de interação *com um professor humano (especialista) e a sua evolução* ativa, *bem próxima do raciocínio empírico do especialista e a sua aplicação no domínio da Harmonia Musical, mostrando um sistema de aquisição do conhecimento baseado na aprendizagem humana. A aquisição de conhecimento é* incremental *de acordo com a troca de informações entre os diversos agentes que compõem o protocolo de aprendizagem* MOSCA.

ABSTRACT

This paper is concerned with analysing the learning steps to be used by an artificial student (a machine), its interaction with a human teacher (an expert), and its active evolution towards a closer representation of the expert's empirical reasoning. All of this, leads to a knowledge acquisition system reflecting the way we aquire our knowledge. The system is illustrated by an application to the domaine of Musical Harmony. The knowledge acquisition is incremental, in accordance with the exchange of information between the several agents composing the learning protocol named MOSCA.

INTRODUÇÃO

Quando falamos em aprendizagem, consideramos a existência de dois agentes: um aprendiz e um professor. Nesse processo de aprendizagem consideramos o papel do aprendiz como sendo desempenhado ora por uma máquina, ora por um humano, devendo os dois agentes interagirem e cooperarem no sentido de:

(i) captar e processar informações;

(ii) organizar dados;

(iii) apreender e relacionar conceitos;

(iv) perceber e resolver problemas;

(v) criar conceitos e soluções (McDonald, 1965).

Esta visão privilegia o aspecto cognitivo do ser humano e esta abordagem será considerada como ideal para o ambiente de aprendizagem que propomos.

Neste ambiente, o papel de professor será desempenhado por um humano (especialista) e o aprendiz será representado pela máquina. O processo possui duas partes distintas:

(i) a fase de aquisição de conhecimento da máquina, com a presença de especialistas no domínio de conhecimento e em pedagogia, e

(ii) a fase de transmissão de conhecimentos.

Neste trabalho, vamos nos limitar à primeira fase. A segunda fase depende fortemente da primeira, e será o momento em que a máquina passa a desempenhar o papel de professor (ambiente tutorial).

Em nosso domínio de conhecimento, a Harmonia, propomos uma abordagem que tem por objetivo não o tratamento apriorístico (mais comumente apresentada nas salas de aula), como se a mesma fosse uma linguagem universal, mas uma concepção da Harmonia como fenômeno cultural (Kaplan, 1991), onde cada período da história da música ocidental é determinado por uma prática harmônica própria com suas características específicas. Nesta abordagem, incluímos as regras propriamente ditas da Harmonia e, posteriormente, os conceitos estéticos intimamente relacionado com a ampliação do campo auditivo percebido pelo ser humano (Ferneda, Peres da Silva, Teixeira & Silva, 1994).

Para tanto, faz-se necessário que a máquina possua um mínimo de conhecimento (estruturado e bem representado), que evolua a partir de novas informações ou através de críticas e que seja adequado para o reconhecimento de diversos contextos do domínio. A aprendizagem da máquina ocorre através do fornecimento de exemplos e de explanações, dentro de um processo dialógico.

Após apresentar brevemente o domínio de conhecimento, mostraremos como a proposta de construção de conhecimento de acordo com um diálogo entre os agentes envolvidos no processo de ensino/aprendizagem é adequada ao domínio da Harmonia. Esse diálogo é estruturado dentro do protocolo de aprendizagem denominado MOSCA (Reitz, 1992), o qual considera um esquema de negociação no diálogo. (Baker, 1992; Billet-Coat, Reitz, Hérin-Aimé & Guin, 1993; Billet-Coat, 1994)

HARMONIA: UMA VISÃO CONTEXTUAL

O estudo da teoria musical, é similar ao estudo de qualquer linguagem, pois analogamente nos debatemos com aspectos de vocabulário, gramaticais, sintaxe e de retórica de gramática. Para tal, é necessário termos uma visão histórica que contempla os principais problemas da música em determinada época (Kerman, 1987).

Fica bem claro que, um aluno de Harmonia que estude utilizando somente baixos e melodias dadas raras exceções, terminará se tornando apenas um hábil condutor de vozes, consequentemente, incapaz de uma obra musical que possua uma estrutura harmônica lógica e coerente (Schoenberg, 1979). É eviden tipo de abordagem nada esclarece com relação à Estruturação Tonal, real objetivo do estudo harmônico, bem como nas suas relações dinâmicas entre os acordes.

Um exemplo bem simples, é a questão de consonância e dissonância. Estas definições, não são pura e simplesmente, uma rotulação "abstrata" para definir sons agradáveis ou desagradáveis; não possuem unicamente relações físico-matemática intrínsecas. Mesmo as correntes que defendem a relação desses conceitos com a distância entres os graus na sucessão dos harmônicos, concordam que, na realidade, o ouvido é quem amplia ou não, as fronteiras de tais rótulos. O que em uma determinada época era ouvido como um intervalo dissonante, na época seguinte passa a ser assimilado pelo ouvido, tornando "suave" o que era, até então considerado "áspero" (Kaplan, 1991).

Isto torna bem claro que estes conceitos são relativos e que o correto é buscar saber como tal intervalo era considerado em um determinado período. Na Figura 1 podemos visualizar as maneiras com as quais os compositores iam acostumando o ouvido às "dissonâncias".

Inicialmente a nota dissonante aparecia no acorde anterior como consonante, era mantida na mesma altura e voz no acorde posterior, seguindo e resolvendo em um acorde consonante (Figura 1a) .

Com o uso seguido o ouvido já começa a se acostumar com o novo acorde, permitindo o uso do mesmo sem preparação (Figura 1b), pois já não existe um desconforto auditivo com a mesma intensidade anterior. O passo seguinte é a não necessidade de preparação e nem de uma resolução consonante (Figura 1c).

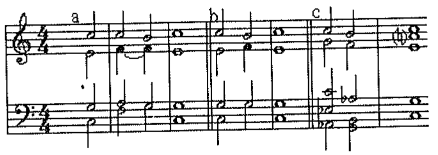


Figura 1: Exemplo de tratamento para as dissonâncias

Este tipo de estudo permite ao aluno não só conduzir as vozes mas, dentro de um contexto histórico, determinar, não apenas qual a época, como também classificar as dissonâncias como tais ou como dissonâncias em transição (Kaplan, 1991).

Um outro ponto que deve ser levado em consideração, além da parte formal (regras), diz respeito às questões Estéticas e de Percepção de forma a capacitar o aprendiz (máquina) para o reconhecimento, análise e construção de exemplos. A Estética leva em consideração a construção de uma estrutura harmônica lógica e coerente (Schoenberg, 1979; Ferneda, Peres da Silva, Teixeira & Silva, 1994). Evidentemente tal questão é formalizada através das regras, ficando as exceções por conta das quebras de estilo e forma próprias da criação. A Percepção, como já foi comentado acima, contribui de maneira muito sutil na aceitação de determinadas sonoridades.

O PROTOCOLO MOSCA

Nesta seção, apresentaremos a estrutura do protocolo de aprendizagem MOSCA (Reitz, 1992) e o esquema de diálogo proposto sobre esse protocolo.

MOSCA é um modelo de interação composto de cinco papéis: o MESTRE, o ORÁCULO, a SONDA, o CLIENTE e o APRENDIZ. Cada um desses papéis representa um comportamento específico que pode ser definido de acordo com os objetivos pretendidos. Por exemplo, em (Ferneda, Py, Reitz & Sallantin, 1992) encontramos uma definição de tais papéis onde também a máquina é o aprendiz e os outros papéis são desempenhados por agentes humanos, com o objetivo de dispormos de um ambiente de apoio à descoberta em Geometria Euclidiana Plana. Já em (Costa, Lopes & Ferneda, 1995) considera-se a situação de um Sistema Tutorial Inteligente, onde o aprendiz é

desempenhado por um humano e os outros papéis pela máquina. Esses papéis podem ser descritos informalmente como:

(i) o APRENDIZ, que dispõe de uma hipótese aprendida tendo uma relação de adequação com uma amostra;

(ii) o ORÁCULO, que produz problemas resolvidos cuja solução não é refutável;

(iii) o CLIENTE, que submete ao aprendiz problemas e espera soluções;

(iv) a SONDA, que, como um oráculo, produz problemas resolvidos mas cujas soluções são refutáveis. Seu objetivo é simplesmente obrigar o aprendiz a argumentar;

(v) o MESTRE, que analisa a argumentação do aprendiz e o critica.

O ambiente de aprendizagem é resumido no esquema da Figura 2.

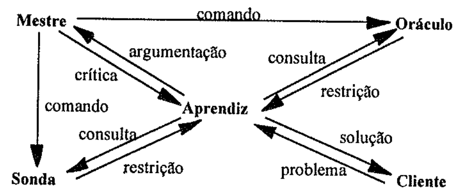


Figura 2: *O protocolo* MOSCA.

Mais detalhes sobre a aplicação deste protocolo orientada à Análise Musical pode ser encontrada em (Ferneda, Peres da Silva, Teixeira & Silva, 1994).

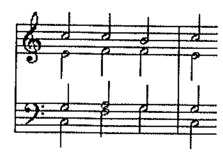
O DIÁLOGO

Como vimos, é necessário uma especificação precisa para cada agente do MOSCA e os papeis que eles desempenham. O núcleo do diálogo está no APRENDIZ, aqui representado pela máquina, que nos fornecerá respostas baseadas no seu conhecimento atual. Estas respostas, em nosso domínio, podem ser tanto uma análise como uma construção harmônica. Ilustraremos agora um diálogo típico entre a máquina e o especialista.

Ao se defrontar com um *problema* enviado pelo CLIENTE, o APRENDIZ buscará encontrar uma *solução*. Nessa busca, o Aprendiz negociará com o MESTRE uma solução aceitável. Essa negociação envolve um processo de diálogo envolvendo argumentações partidas do APRENDIZ para o MESTRE e as críticas no sentido contrário. Os papéis de ORÁCULO e de SONDA representam o conhecimento de apoio ao MESTRE no desempenho da sua interação com o APRENDIZ. Como resultado teremos uma solução válida em relação ao ORÁCULO e relevante em relação ao MESTRE. Essa solução é então encaminhada ao CLIENTE.

Partindo agora do pressuposto que o domínio de conhecimento do APRENDIZ é a Harmonia Tradicional, apresentaremos um problema que, embora simples, é representativo deste domínio: um encadeamento para ser identificado. Partimos do pressuposto que nosso APRENDIZ dispõe de conhecimento preliminar sobre escalas, tonalidades, progressão por quintas, alterações, valores das notas e das pausas, sinais de indicações de compasso, clave de sol e de fá e, principalmente, intervalos em todas as suas formas.

Problema apresentado (C maior):

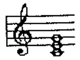
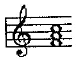


Neste momento, como o nosso diálogo acontece através do confronto de argumentos e críticas, o APRENDIZ envia ao CLIENTE a sua solução, ao mesmo tempo em que justifica ao MESTRE a sua resposta.

Solução do APRENDIZ:

Encadeamento I - IV - ? - I.

Argumentação do APRENDIZ *para o* MESTRE:

Eu aprendi que  é um acorde de C (tônica);

Eu aprendi que  é um acorde de F (subdominante).

Por sua vez, o MESTRE aceita esta justificativa como relevante como argumentação de sua solução. Como a resposta apresentada está incompleta, nessa situação, o MESTRE apresenta, via SONDA, um questionamento referente à parte problemática.

Comando do MESTRE *para a* SONDA:

Selecione um acorde do tipo V7 e apresente ao Aprendiz.

Objeto apresentado pela SONDA *ao* APRENDIZ:

 é um acorde de dominante.

Frente a esse objeto, o APRENDIZ, que não reconhece tal acorde, justifica sua resposta ao MESTRE.

Argumentação do APRENDIZ *para o* MESTRE:

Eu aprendi que  é um acorde de V.

Como resposta, o mestre envia uma crítica sobre essa justificativa. :

Crítica:

Justificativa falsa.

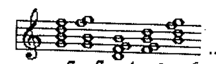
A partir de agora o nosso diálogo muda para o nível de aprendizagem, uma vez que foi detectada a limitação do conhecimento do APRENDIZ em reconhecer esta variação de um acorde de dominante. Nesse caso, o MESTRE instrui o ORÁCULO a explanar o conceito de acorde dominante com sétima:

Comando do MESTRE *ao* ORÁCULO:

apresente ao Aprendiz o conceito V7
e um conjunto de variações como exemplo.

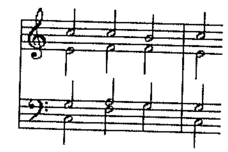
ORÁCULO *para o* APRENDIZ:

V7 é um acorde de V ao qual se acrescenta
a sétima de sua fundamental.



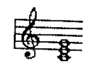
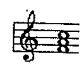
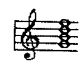
$V^7\ V^7\ V^4_3\ V^2\ V^6_5$ ...

Nestes exemplos oferecidos pelo ORÁCULO, pode-se, ou incluir o restante das possíveis variações, como omissão da tônica ou ir paulatinamente alimentando a base de conhecimento do APRENDIZ. Agora, quando o primeiro problema é reapresentado, o APRENDIZ terá condições de dar a resposta completa e correta:

Problema apresentado:

Solução do APRENDIZ:

Encadeamento I - IV - V7 - I.

Justificativa do APRENDIZ *para o* MESTRE:

Eu aprendi que  é um acorde de C (tônica);

Eu aprendi que  é um acorde de F (subdominante);

Eu aprendi que  é um acorde de G7 (dominante).

Se entendermos que as regras formais armazenadas inicialmente na base de conhecimento do APRENDIZ correspondem aproximadamente ao três primeiros capítulos do livro de Paul Hindemith (Hindemith, ??), veremos que o nosso APRENDIZ teve um bom desempenho. No entanto, o MESTRE preferiu ampliar o conhecimento atual do APRENDIZ, devolvendo-lhe uma crítica que, na realidade, funcionou como partida para o nível de aprendizagem de novos conhecimentos.

Optamos por um exemplo relativamente simples para evidenciarmos a forma do diálogo entre os vários agentes do protocolo MOSCA e como o MESTRE pode continuar este processo de ampliação do conhecimento do APRENDIZ, não só tratando das regras harmônicas, mas incursionando através dos domínios de Estética, Percepção e História da Música (Ferneda, Peres da Silva, Teixeira & Silva, 1994), como descrito no início deste artigo. De posse de um conhecimento razoável em Teoria, o APRENDIZ ficaria apto a absorver estes conceitos inerentes à criação musical. Sua atuação não se resumiria apenas a um comportamento racional ao nível de análise, mas também a nível de construções harmônicas baseadas em estilos, formas, etc.

CONCLUSÃO

Neste artigo apresentamos um ambiente de aquisição de conhecimento em Harmonia. Este ambiente considera uma situação em que a máquina se comporta como um aprendiz e onde o papel de professor é desempenhado por especialista humano.

Nosso objetivo, utilizando o protocolo MOSCA, foi mostrar como um diálogo pode ser feito entre o MESTRE (humano) e o APRENDIZ (máquina), tendo como resultado o crescimento da base de conhecimento do aprendiz de forma *ativa* e *interativa*. Enfatizando, neste caso, os elementos de Teoria, Estética e Percepção, conseguiríamos da máquina um comportamento racional na análise e construção em Harmonia.

Apresentamos a funcionalidade no processo de aprendizagem através de um problema envolvendo um encadeamento harmônico.

Atualmente estamos em fase de especificação do ambiente em questão. Uma vez essa primeira fase de aprendizado da máquina concluída, utilizarmos esse conhecimento aprendido pela máquina para um ambiente tutorial inteligente.

REFERÊNCIAS BIBLIOGRÁFICAS

M. Baker "Le rôle de la collaboration dans la construction d'explications", *Actes des Journées Explication du PRC-GDR-IA*, Sophia-Antipolis, França, 1992.

S. Billet-Coat, Ph. Reitz, D. Hérin-Aimé, D. Guin "Protocole comportamental de l'interaction didactique entre un agent artificiel et un agent humain", *Actes des 3èmes Journées EIAO*, Cachan, França, 1993.

S. Billet-Coat "Processus interactif d'acquisition de connaissances basé sur un apprentissage humain", *RFIA'94 - Reconnaissance de Formes et Intelligence Artificielle*, Paris, França, 1994.

E. de B. Costa, M. A. Lopes, E. Ferneda "MATHEMA: A Learning Environment based on a Multi-Agent Architecture", trabalho submetido ao X Simpósio Brasileiro de Inteligência Artificial - SBIA'95, Campinas, outobro 1995.

E. Ferneda, C. A. Peres da Silva, L. de M Teixeira, H de M. Silva "A System for Ainding Discovery in Musical Analisys", Anais do I Simpósio Brasileiro de Computação e Música, Caxambu (MG), julho 1994.

E. Ferneda, M. Py, Ph. Reitz, J. Sallantin "L'Agent Rationnel SAID: une application en géométrie", *Proceedings of the First European Colloquium on Cognitive Sciences*, Orsay (França), junho 1992.

J. A. Kaplan "Harmonia", *Notas de aula*, Departamento de Música, Universidade Federal da Paraíba, João Pessoa, Paraíba, 1991.

J. Kerman "Musicologia", Martins Fontes, São Paulo, 1987.

F. J. McDonald "Educationa psychology", Belmont, Califórnia, Wadsworth Publishing Company, 1965.

Ph. Reitz "Contribution à l'étude des environnements d'apprentissage. Conceptualisation, Spécifications et Prototypage. *Tese de doutorado*, Université Montpellier II, Montpellier, França, fevereiro 1992.

A. Schoenberrg "Tratado de Armonía", Real Musical, Madrid, Espanha, 1979.

II Simpósio Brasileiro de Computação e Música (II SBC&M)

CD Programme Notes*

Electrocañas I

Carlos Cerana (B flat clarinet; programming)
Diego Losa (alto sax)
1995

Electrocañas I (in English, Electroreeds I) is the first result of a project exploring the interaction between reed instruments and digital media. Resources include sounds of live instruments, their digital processing, and control of electronic sounds by means of pitch-to-MIDI conversion and Max processing. The piece is generated from a melodic cell, which appears both in written parts and in improvised solos. A second section introduces electronics and transformation of sound.

Equipment used: Boss SE-50 digital sound processor, Yamaha TG-77 synthesiser, Richsound pitch-to-MIDI converter, Roland pedals (expressive and switch), and Macintosh Powerbook running Max.

Carlos Cerana was born in Buenos Aires, Argentina, in 1958. He studied clarinet with Néstor Tomassini and composition and electroacoustics with Francisco Kröpfl. He taught at the National Conservatory and at the Municipal Conservatory of Buenos Aires. In 1988 he was appointed as a researcher at the Laboratorio de Investigación y Producción Musical (LIPM) in Buenos Aires. In 1992 he was invited to work at the Center for Research in Computing and the Arts of the University of California in San Diego (CRCA) and at the Center for Computer Research in Music and Acoustics at Stanford University (CCRMA), as part of an exchange programme supported by the Rockefeller Foundation. In 1994 he participated in a concert tour visiting several universities in the United States and Canada. His piece *Fall* was awarded the prize for electroacoustic composition by the National Endowment for the Arts of Argentina. Cerana received a scholarship from the Antorchas Foundation and is presently researching electroacoustic music performance with a grant from the National Endowment for the Arts. He is a member of the Argentine Federation of Electroacoustic Music (FARME) and of the International Computer Music Association (ICMA).

Diego Losa was born in Buenos Aires, Argentina, in 1962. He studied traverse flute with Bruno Bragato, Alfredo Ianelli and Pablo Levin, sax with Gustavo Dinerstein and harmony with Julio Viera. He is an active performer in the popular music field, and is presently a member of the Bandgap jazz band. He is also a sound engineer at the LIPM, where he teaches courses on equalization and mixing techniques. Losa participates in musical research with electroacoustic media. He has premiered several pieces for sax and electronics and has shared projects with visiting composers from the USA, as part of an exchange program with LIPM supported by the Rockefeller Foundation.

Piece of Mind

Celso Aguiar
1995

The title of the piece is a word play on two homophonous words in the English language ("piece" and "peace") and stands as a reaction to the different ways we face fun and pleasure. Exploring the concept "Fun is dangerous" (which Aguiar was told once in the USA), *Piece of Mind* is about the violence of fun or the fun of violence, at a more extreme level than is normally encountered in everyday life.

The piece can also be seen to operate as a continuum between two sound processing techniques in a blend of *musique concrète* and powerful spectral modelling synthesis. Julius Smith predicted that time domain techniques (such as sampling and granular synthesis) will be absorbed into frequency domain techniques (such as spectral modelling). This idea has evolved using particular but complementary resources from each technique to benefit the musical discourse.

In 1994, Aguiar worked with Xavier Serra at CCRMA to recreate an IFFT algorithm for additive synthesis. Some of the instruments used in the piece were created in this way. The piece was composed using the Lisp tools CLM and Common Music on the NeXT computer at CCRMA, Stanford University.

* Compiled by Eduardo Reck Miranda from information given by the composers.

Celso Aguiar was born in Palo Alto, California, in 1957 and grew up in Salvador, Bahia, Brazil. He began his musical studies with Swiss-Brazilian composer Ernst Widmer at the Federal University of Bahia (UFBA). Since then he became particularly interested in electroacustic music and in consequence developed a project at UFBA to construct his own computer-controlled digital synthesiser. He has written music for traditional instruments as well as electroacustic means and his main approach in this area has been the application of new digital signal processing techniques to composition. With these objectives in mind he is currently studying for a Doctorate in Composition at Stanford University.

Figuras Flamencas
Mario Verandi
1995

The source material for this piece includes samples from Spanish flamenco music and texts by Latin-American writer F.G. Lorca. Verandi used several sound manipulation techniques (for example, shuffling and time stretching) to create a *surreal sound world* from the *flamenco music world*. The musical discourse flows as a sequence of superimposed and juxtaposed transformations and confrontations between the flamenco world and the surreal world; Lorca's texts serve as a scaffolding-like structure to establish the relation between these two worlds. In this piece the intention is to evoke a metaphorical journey across a flamenco dreamland which is corrupted by surreal sonic creatures.

The piece was composed from December 1994 to March 1995 at Birmingham University, in England. The main software used in the composition were GRM tools, Sound Designer and Pro-Tools on the Macintosh computer. *Figuras Flamencas* belongs to a larger piece which Verandi is currently composing.

Mario Verandi was born in Buenos Aires, Argentina. He studied Computer Science at the University of Belgrano and Music at Rosario University. In 1989 he went on to study in Barcelona at the Joan Miro Foundation's Phonos Laboratory.

He is currently pursuing an MMus in Composition at Birmingham University under the supervision of Jonty Harrison. He is particularly interested in multimedia projects and has composed music for art installations, dance, video and theatre. His work has been broadcast, performed and recorded in Europe and the USA.

Bat out of Hell
Stephen Trevis Pope
1983

Bat out of Hell is computer-generated tape music for ballet. It is envisaged as a solo percussion piece for a virtuoso with 168 microtonally-tuned bells. The two short sections of the work are intended to evoke certain gestures and shadows in dancers, who act out the entry into a ritual place for the passion of fear. *Bat out of Hell* was realised at the CMRS studio in Salzburg using the ARA LISP programs to generate Music11 notelists for the bell sounds. The composition works with a small number of elements which are reflected by various mirrors within the score. The low-pitched bells are simply the sighs, breath, and heartbeat of the performer.

Stephen Travis Pope was born in the USA in 1955. He studied Electrical Engineering at Cornell University, and Music Theory and Composition at the Vienna Music Academy and the *Mozarteum* in Salzburg. During his 15-year involvement with computer music, he has moved between academia and industry several times, and is now active as a software consultant through The Nomad Group and a research associate at the Center for New Music and Audio Technologies (CNMAT) at the University of California in Berkeley. He has been an officer of the International Computer Music Association and was elected a lifetime member in 1990. He is editor of *Computer Music Journal*, now in its 19th year, published by the MIT Press. He has realised his works at several studios in the USA and Europe and has published widely in the fields of Computer Music, Artificial Intelligence, Human-Computer Interfaces, and Object-Oriented Programming.

Pericón
Conrado Silva
1989

This piece is composed of variations on a well known folk tune from Uruguay, called *Pericón*. Silva selected four short excerpts of this tune (each only a couple of bars long) and transformed them using several musical variation techniques (for example, dynamic amplification and note permutation) with the addition of contrasting musical passages and thematic distortion. The piece works as a kind of real-time improvisation, where the computer controls three synthesisers in response to Silva's keyboard improvisation. In his *Pericón*, Silva attempts to achieve a readily identifiable listening environment, by using well known tunes imbued with some mocking sound colours.

Conrado Silva was born in Montevideo, Uruguay, in 1940 but has lived in Brazil since 1970. He studied both Music and Engineering at Montevideo, Munich and Berlin. Silva is one of the most influential leaders of the contemporary music scene in Latin America: he created the *Latin-American Courses for Contemporary Music*, taking place from 1971 to 1989 in different Latin-American countries. He was also one of the founders of the *Festival Música Nova* of São Paulo. He currently teaches Composition and Musical Acoustics at University of Brasilia and is a member of the newly created *Brazilian Society for Electroacoustic Music*. He has composed many pieces for voice, instruments and electronic devices, including: *Music for 10 portable radios* (1964), *Marat-Sade* (1966), *Compulsion Hombrehistorica* (1970), *Parasexteto* (1971), *Polaris* (1977), *Natal Del Rey* (1978) and the acclaimed electronic chamber opera *Espaços Habitados* (1994).

Olivine Trees
Eduardo Reck Miranda
1994

Olivine Trees is perhaps the first piece of electroacoustic music ever composed using a high-performance parallel computer. The piece is specifically composed using sounds synthesised by *Chaosynth* on the Connection Machine CM-200 computer. *Chaosynth* is a sound synthesis system developed by Miranda at Edinburgh Parallel Computing Centre (EPCC), in Scotland. The synthesis technique of *Chaosynth* is inspired by the granular synthesis technique; it functions by generating a large amount of short sonic events, or particles, in order to form larger, complex sound events. It produces a wide range of bubbling sounds in various flow speeds and tone colours; most sounds resemble the morphology of the sounds of flowing water.

Olivine Trees is inspired by Van Gogh's painting, "Olive Trees". The varied and individually identifiable brush strokes of this painting inspired the composition of the sounds of the piece; in direct correlation, colour relates to timbre and length of brush stroke relates to the duration of individual, "granular" sounds. Other signal processing techniques, such as convolution, were also used during the mixing process at the University of Edinburgh's electroacoustic music studio.

Eduardo Reck Miranda was born in Porto Alegre, Brazil, in 1963. He graduated in Data Processing Technology at Vale do Rio dos Sinos University (UNISINOS), Brazil, in 1985 and went on to study Philosophy at the Pontificial Catholic University of Rio Grande do Sul (PUC-RS) and Music at Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil. From 1986 to 1990 he attended several electroacoustic and experimental music courses in South America, including the renowned *Latin-American Courses for Contemporary Music* in 1989. In 1991 he gained a Master degree in Music Technology (MSc) at the University of York, England. In 1992 he also studied computer music at Zentrum für Kunst und Medientechnologie (ZKM), in Karlsruhe, Germany. In 1995 he was awarded a PhD in Music and Artificial Intelligence at the University of Edinburgh, Scotland.

Academic publications include research papers in major international journals, including *Computer Music Journal*, *Interface*, *Leonardo* and *Contemporary Music Review*. Latest music compositions include *Italo Calvino takes Jorge Borges on a taxi journey in Berlin* (electroacoustics on tape), *Entre l'absurde et le mystère* (chamber orchestra), *Noises* (electroacoustics on tape), *The Turning of the Tide* (electroacoustics on tape and prepared violin) and *Mônadas* (percussion ensemble). His piece *Electroacoustic Samba II* was recently awarded a Le Puy prize at Bourges, France. He is one of the creators of NUCOM (the Computer Music branch of SBC - the Computer Science Society of Brazil).

Pyrócua
Ralf Ollertz
1994

Pyrócua is inspired by a poem by Toula Limnaios. It was composed at the electroacoustic music studios of Folkwang School of Music in Essen, Germany. The basic material of the piece are sounds sampled from three traditional acoustic musical instruments: piano, flute and violin. Ollertz also used samples of the sounds produced by a waterheater. Equipment used include Digidesign's Sound Designer and Pro Tools on the Macintosh computer, a sampler Akai S1000 and a Lexicon 480L.

Ralf Ollertz was born in Mönchengladbach, Germany, in 1964. He studied composition, electronic music and conducting with Vivienne Olive, Wilfried Jentzsch and Werner Andreas Albert at the Meistersinger Conservatory in Nürnberg. He also studied composition with Günther Becker at the Robert Schumann School of Music in Düsseldorf.

After a scholarship in Italy, where he studied composition with Salvatore Sciarrino, he studied for a Master's degree in electroacoustic music and studied conducting with Armin Klaes and Peter Eötvösz at the Folkwang School of Music in Essen. Since 1994 Ollertz has studied dance and choreographic composition with Jean Cebron. He has worked recently with artists, including Bettina Elmpt, Jens Kaul, Clarence Barlow and Hartmut Geerken.

In addition to instrumental and electroacoustic music he has realised many interactive installations, radioplays, music-choreographies and music for dance and theatre. He is the director of *go ahead* (an ensemble for new music) and the musical director of the Claudia Lichtblau Dance Company.

Noite
Victor Lazzarini
1995

This piece is a setting of some fragments of poetry by the Portuguese poet Fernando Pessoa, from the book *Ficções do Interlúdio III, Poemas de Álvaro de Campos*. The text consists of imagery associated with the night:

Na noite terrível, substância natural de todas as noites
Na noite de insônia, substância natural de todas as minhas noites

Noite igual por dentro ao silêncio, noite
Com as estrelas lantejoulas rápidas
No teu vestido franjado de infinito

Nesta noite em que não durmo e o sossêgo me cerca
Como uma verdade de que não partilho
e lá fora o luar, como a esperança que não tenho, é invisível
pr'a mim.

The piece has a straightforward structure (according to Lazzarini) based on the three stanze of the text, which constitute well-defined blocks, followed by musical commentaries. The whole piece works as a diminuendo, starting from a dramatic setting of the first verses and ending with the serenity of the last words being whispered.

Noite was composed at the University of Nottingham's electroacoustic music studio. It was fully produced on the PC486 DX2 computer running the Composer's Desktop Project (CDP) software. Most of the sounds were sourced from readings of the text, eventually processed using Mark Dolson's Phase Vocoder and Trevor Wishart's spectral manipulation software. Other sounds were synthesised using the Csound synthesis package.

There is also a second version of this piece for tape, live electronics and orchestra. It integrates a larger work, *Magnificat*.

Victor Lazzarini was born in Londrina, Brazil, in 1969. He began his musical studies at the local conservatory, where he learned music theory and piano. He went on to study composition with Almeida Prado and Damiano Cozzella at the University of Campinas (UNICAMP). In 1993 he received a scholarship to pursue a post-graduate degree at the University of Nottingham, under the supervision of Nicholas Sackman.

/cartas/rs95.car
Aluízio Arcela
1995

/cartas/rs95.car is a composition generated entirely by CARBON, a program for algorithmic composition developed by Arcela at University of Brasilia Computer Music Laboratory (LPE). Arcela regards this piece as a *time tree theorem*; CARBON processes the theorem and produces scores, which are interpreted by two other programs: SOM-A and ILUSOM (both developed at LPE). The former produces sounds whilst the later produces images. SOM-A is an additive synthesiser using 132 generators (or *instrumentos ortoestereofônicos*); it can produce sounds with up to 52 partials. ILUSOM is a program that produces a visual representation of the time tree threorem. */cartas/rs95.car* was originally composed for live performance with violin, marimba, projected images and synthesised sounds.

Aluízio Arcela was born in Rio de Janeiro, Brazil. He attained his Doctorate in Computer Science at the Pontificial Catholic University of Rio de Janeiro (PUC-RJ) in 1984. In his thesis he devised a technique, *time trees* (*árvores de tempo*), to represent the inner structure of musical intervals. He also developed a series of software for sound synthesis and algorithmic composition based upon time trees (for example, SOM-A and CARBON). Arcela recently devised ILUSOM, a program that generates pictures from time tree representations of music intervals.

In 1989 Arcela created the Master Course in Computer Music at the University of Brasília (UnB); the first ever University-level course on computer music in South America. He currently conducts research, teaches and directs LPE and also teaches at the Department of Visual Arts at the same university.

Saudades de Ouro Preto
Robert Willey
1995

Saudades de Ouro Preto is a computer-mediated performance using the Max software on a Macintosh computer. It uses Max Mathew's idea of the "sequential drum", in which the computer plays the next note of the score each time the instrument is played. This implementation allows the performer to improvise at the keyboard rather than entering the whole score into the computer.

Robert Willey is a research associate at the Center for Research in Computing and the Arts (CRCA), at the University of California in San Diego. For the last five years he has assisted a computer music exchange between CRCA, CCRMA (Stanford University), and LIPM (Buenos Aires). During that time he also began to give support for the development of computer music in Brazil and since then he has been giving workshops and concerts at a number of universities throughout the country, most recently at Federal University of Minas Gerais' Winter Festival.

Índice por Autor