A Markovian Multiagent Musical Composer

Joel L. Carbonera[1], **João L. T. Silva**[1]

[1]Universidade de Caxias do Sul, Centro de Computação e Tecnologia da Informação
Caxias do Sul, Brazil, 95700-000

`{jlcarbon,jltsilva}@ucs.br`

Abstract. *In this paper we propose a computational model for stochastic melodic synthesis based on an approach developed through reactive multiagent systems. In general automatic musical composition approaches, the states transition probabilities are obtained from statistical analysis of musical segments provided by the user, which tends to generate musical results very similar to the initial musical segments. As alternative to these characteristics, we consider a computational model in three layers: the first layer aims to generate transition probabilities matrices (pitches and durations of musical notes, in this context) to supply the second layer. The second layer produces probabilistic models for generation of the melodic segments ($PMMSG$), through the interaction with the user, in order to supply the stochastic melodic synthesis process at the third layer. This model aims to reduce the influence of the composer in the final composition, in order to build melodic structures not originally conceived by the composer. Nevertheless, this work does not cope with the aesthetic quality of the melodic results, since the work focus is the spontaneous generation of the initial information that will guide the stochastic melodic synthesis.*

Resumo. *Neste artigo, é proposto um modelo computacional para síntese melódica estocástica baseado em uma abordagem desenvolvida sobre sistemas multiagentes reativos. As abordagens tradicionais de composição musical automática obtém probabilidades de transição de estados através da análise estatística de fragmentos musicais fornecidos pelo usuário, o que tende a gerar resultados musicais muito semelhantes aos fragmentos musicais iniciais. Buscando alternativas a estas características, consideramos um modelo computacional em três camadas: a primeira camada é responsável pela geração de matrizes de probabilidades de transição de estados (alturas e durações de notas musicais). A segunda camada, através da interação do usuário, gera modelos probabilísticos para geração de segmentos melódicos ($MPGSM$) a partir das matrizes. A terceira camada finaliza o processo de síntese melódica estocástica com estes segmentos. Este modelo tem como objetivo reduzir a influência do compositor na geração final, a fim de buscar a construção de estuturas melódicas que não foram pensadas inicialmente pelo compositor. Desta forma, este trabalho não trata computacionalmente a análise da qualidade estética do resultado melódico, uma vez que o foco é a geração espontânea das informações iniciais que nortearão a síntese melódica estocástica.*

1. Introduction

The computer music is an inherently multidisciplinary field involving all areas of knowledge related to Computer Science and Music. The computer music area involves knowledge ranging from computer modeling of musical aspects to the study of aesthetic conceptions of artistic expression musical, including the study of techniques for the generation, storage and processing of sound, etc. Many works have been produced in the area of sound and music composition processing and renowned composers have been systematically using computational tools for musical composition [Cruz 2001, Fischman 2003, Burraston and Edmond 2005, Miranda 2000, Xenakis 1992, Papadopoulos and Wiggins 1999, MacGee and Schaffer 1997, Bryan-Kinns 2004].

Many musical synthesis systems produced by traditional initiatives are usually dependent on the user. In this sense, the musical constructions are not completely automatic. Traditionally, initial samples are provided manually by the user and the system of musical synthesis is limited to process these samples or data, in a static and pre-defined way. This paper investigates a close relation between music and Artificial Intelligence from a multiagent point of view. We intend to demonstrate how the collective behavior of reactive agents can lead to a quasi-automatic compositional system.

Concerning with stochastic models to musical synthesis, our work differs from others in many aspects. For instance, in the framework for generating Palestrina-style music using Markov models [Farbood and Schoner 2001], the authors aim to generate counterpoint segments rules from state-transition matrices. In this work, probabilistic counterpoint rules are implemented as a probability table starting by probability zero when the system finds illegal transitions. Thus, the generation of a counterpoint line is obtained by multiplying the individual values from each transition probability table, where each table captures one aspect of counterpoint line, as for instance, harmonic table, melodic table, cadence table, chromatic table and so on. The main idea is to analyze a previous composition and then to infer a probabilistic description of counterpoint rules. As argued by the authors, the final result can be comparable to those created by a knowledgeable musician, corroborating our claim about the similarity between the input and output compositions. Also, McCormack [McCormack 1996] has pointed some drawbacks on Markov process for music composition related to the generality of the transition table and the management of higher order models. When some real existing music is used as input into the system, the result is a Markov model which only generates music of the same style of the input. Also, higher order Markov models will require extra computational effort that can hinder real time performance and, also, probably to provide little support for structure at higher levels. In these cases, multiple layered models can be used where each event represents an entire Markov model in itself or another type of structure. In our work, we present a layered architecture which may cope in the future with these problems related to higher order Markov models.

2. Model Overview

The three layered architecture proposed in this work is illustrated in Figure 1. The lower layer (*Reactive generation of transition probability matrices*) encapsulates the process of reactive generation of probability transition matrices. In this layer, a multiagent environment is responsible for stochastic generation of transition probabilities based on a reactive behavior. Two distinctive environments are in charge of built transition probability matrices for pitches and for durations through interaction of reactive software agents. The middle layer (*Definition of probabilistic models for melodic generation*), at Figure 1, represents the process of construction of probabilistic models for generating melodic segments based on the matrices produced by the multiagent environments. Finally, the uppermost layer (*Stochastic melodic synthesis*) represents the stochastic process of melodic synthesis.

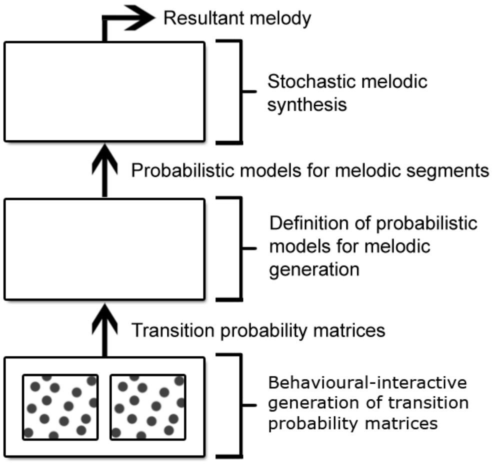


Figure 1. Three layered model of the emergent stochastic melodic composition system.

Next we present the three layers in details mainly focusing on emergent melodic generation model through reactive multiagent environments allied to Markov decision model.

2.1. Transition Probabilities Generation Layer

The state transition matrices (of pitches and durations) feed the process of stochastic melodic composition. The model to generate the probability matrices used in this work is based on dynamic interaction among reactive agents. In a computational context, reactive agents are autonomous pieces of software that act in a determined environment without human intervention. These agents are capable to interact with other agents, to perceive the environment where they are and to react based on percept stimulus without reasoning.

The multiagent environments are instances of an euclidean plane whose dimensions are defined by the user and where the agents move themselves. Each element from this euclidean plane represents NxN co-occurrence matrices where N represents the 12 pitches of the musical notes, i.e., the 12 semitones (from C, $C\#$, ..., $B\flat$, B). The environment that generates duration transition probability matrices, N is 7, which represents the total number of musical notes duration in this work (*Semibreve, Minim, Crotchet, Quaver, Semiquaver, Demisemiquaver, Hemidemisemiquaver*) (see Figure 2).

On the environment, the agents pursue some goals that are user-defined points called *Target-points*. These *Target-points* are defined through a point (euclidean coordinates x, y) located inside the plane limits. The *Target-points* carrying out an attraction

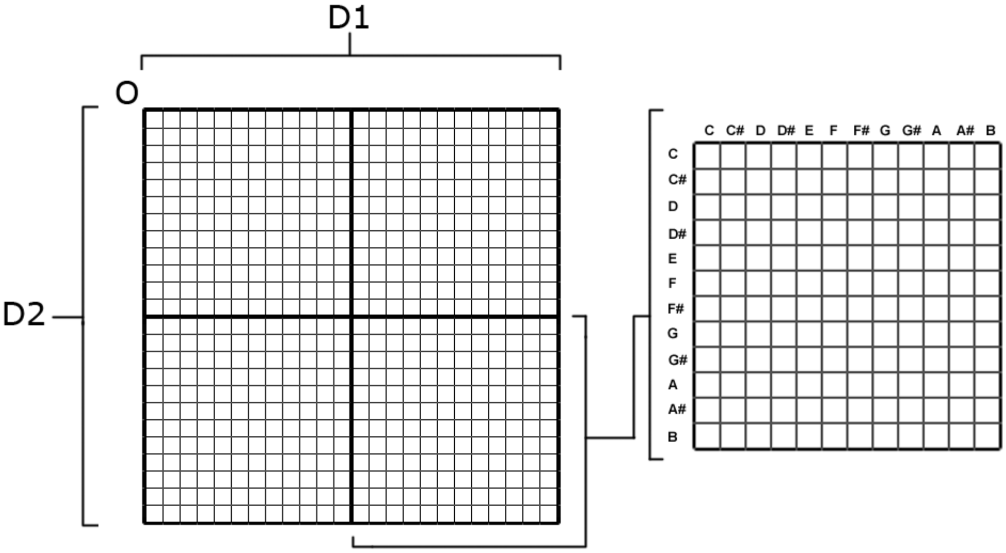


Figure 2. Example of environment structure composed by four co-occurrence matrices of pitch transition probability

force into the agents in such a way that each agent is obliged to pursue some of these *Target-points* while they move themselves randomly on the environment.

In this work, the agent behaviors are based on the subsumption architecture [Brooks 1986] and on potential fields approach [Mezencio 2002]. The behaviors are arranged in a hierarchical two level structure based on the behavior priority (Figure 3). The first level (greatest priority) represents the agent evasive behavior (including trajectory adjustment to avoid invasion of the perimeter of another agent) while the second level represents the behavior "go forward the target" (including trajectory adjustment to trace a direct line to the target-points).

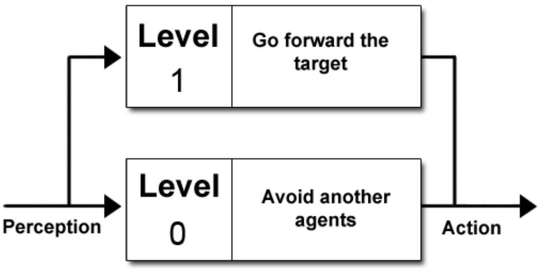


Figure 3. Hierarchical agent behaviour based on subsumption architecture.

The action of the agent consists in adjust the vector of direction and to move towards this vector at each perception-action cycle. The resultant vector is obtained through the generation of basic agent behaviours through the overlapping of the potential fields that influence the agent at some time. This overlapping is defined as the sum of the vectors of each one of the forces (repulsion - vec_rep and attraction - vec_atr) that influence the agent obtaining a vector that defines the speed and the direction of the agent movement (equations (b) and (c) on the Algorithm 1). The agent defines an action in the subsumption hierarchy through its perception information. The *Invasion-Limit-Range* (ILR) algorithm 1 determines if some of the other agents have violated the restriction of minimal distance (r) in order to trigger an *Avoid-Agent-Collision behaviour* (equations (a) to (e) on the Algorithm 1) or a *Go-Forward behaviour* (equations (e)).

```
Input: set of perceptions ($\mathcal{P}$), set of $n$ agents ($\mathcal{A}$), lim range ($r$)
where $pos_{self}, pos_{target} \in \mathcal{P}$ ;
Output: direction ($dir$)
foreach agent $g \in \mathcal{A}$ do
    Calculate ILR($g$,$pos_{self}$);
    if ILR($g$,$pos_{self}$) $\leq r$ then
        /* Avoid-Agent-Collision Behaviour */
        $avr\_pt = \sum_{i=1}^{n} pos_i/n;$                         (a)
        Calculate $vec\_rep(avr\_pt, pos_{self})$;                  (b)
        Calculate $vec\_atr(pos_{self}, pos_{target})$;             (c)
        Calculate $dir = vec\_rep + vec\_atr$;                      (d)
    else
        /* Go-Forward Behaviour */
        Calculate $dir = vec\_atr(pos_{self}, pos_{target})$;       (e)
    end
end
```

Algorithm 1: The *Invasion-Limit-Range* (ILR) algorithm that select the action based on the subsumption hierarchy.

Before running the system, the user provides some parameters to the environment and defines some Target-points. At the start of the execution, each agent is randomly placed over the environment and each one start their behaviors in order to reach the closer Target-point, always avoiding other agents. While moving themselves, the agents increase by one a value at the element on the co-occurrence matrices (first initialized with zero). This behaviour distributes some values over the co-occurrence matrices building the state transition matrices for pitches and durations.

2.2. The PMMSG layer

A probabilistic model to melodic segments generation (PMMSG) constitutes a data structure built from the set of transition probability matrices generated in the previous layer. The main goal associated to the PMMSG is to drive the melodic stochastic synthesis in an interactive way.

The user is able to define models that encapsulate a pair of transition probabilities matrices, which will define the distribution of pitch and duration in distinct segments of the melody. Thus, each PMMSG has the potential of generate local melodic segments with particular features, within the global melody. Formally:

Definition A Probabilistic Model to Melodic Segments Generation (PMMSG) is a couple $PMMSG\langle \mathcal{Q}^P, \mathcal{Q}^D \rangle$, where $\mathcal{Q}^P$ is a probability transition matrix of Pitches and is $\mathcal{Q}^D$ is a probability transition matrix of Durations.

These probability transition matrices are defined in terms of a Markov model definition. A Markov chain [Meyn and Tweedie 1993] is a specific kind of discrete-time stochastic process that given some present states, the future ones are chosen independently of the past states, which is called Markov property. In our work, the melodic synthesis is modeled as a succession of musical notes where these notes are defined by their pitch and duration.

In this work we use a traditional Markov Chain definition:

Definition A Markov Chain with state space $\mathcal{S} \subseteq \mathbb{I}$ is a discrete stochastic process $\{X_n; n \in \mathbb{I}\}$ if each X_n assumes values only in $\mathcal{S}$ and

$$\begin{aligned} P(X_{n+1} = x | X_n = x_n, ..., X_0 = x_0) = \\ P(X_{n+1} = x | X_n = x_n) \end{aligned} \tag{1}$$

holds for all $n \in \mathbb{I}$ and $x_0, x_1, ..., x_n, x \in \mathcal{S}$.

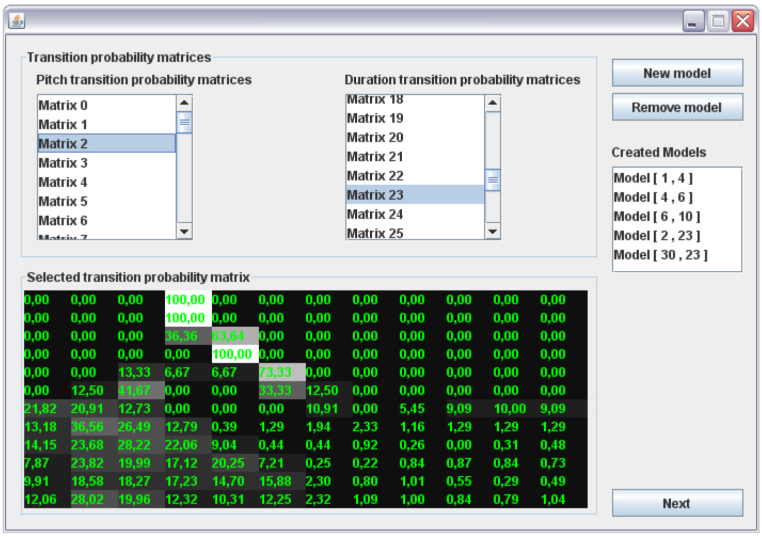
Definition A **probability transition matrix** (PTM) is a square matrix NxN, whose entries are all nonnegative and whose rows sum to 1. Each PTM describes the probabilities of a current state to change to a future state in a dynamic system. Considering a PTM $\mathcal{Q}$, $\mathcal{Q}(x'|x)$ represents the probability of going to the future state x' given that the existing state is x, and $x, x' \in \mathcal{Q}$.

Definition A **probability transition matrix of Pitches** $\mathcal{Q}^P$ is a regular PTM where $N = 12$, describing the probabilities related to the 12 pitches of the musical notes, i.e., the 12 semitones (C, $C\sharp$ or D$\flat$, D, $D\sharp$ or $E\flat$, E, F, $F\sharp$ or $G\flat$, G, $G\sharp$ or $A\flat$, A, $A\sharp$ or $B\flat$, B) from the octave between the $C5$ (the Do two octaves above the piano's center Do) and $C6$.

Definition A **probability transition matrix of Durations** $\mathcal{Q}^D$ is a regular PTM where $N = 7$, representing the probabilities related to the set of 7 musical notes duration (*Semibreve, Minim, Crotchet, Quaver, Semiquaver, Demisemiquaver, Hemidemisemiquaver*).

The functional dimension from the PMMSG that we intend in this work, is to keep in mind some characteristics from the Markov chains. In general, a human composition melody presents several distinct moments that is a kind of musical nuances and contrasts. A statistical analysis of this music will generate a transition probability matrix where these different moments (nuances, contrasts,...) would be reduced to a global statistic, which could very often mutilate the local characteristics of the composition (moments with own characteristics). In this way, even that a new music produced by those specific transition probability matrices could preserve the statistical global characteristics of the initial music, it is very unlikely also to preserve the lower level local characteristics. Some works [Oliveira 2003] have shown us that the music produced stochastically in the traditional ways, from a unique model of probabilistic state transition, can easily become monotonous (without those distinct moments). The use of PMMSGs in this work aims to by-pass the undesirable characteristics described above. In this way, through the concept of PMMSG, the generated melody is based on a set of melodic segments (partial regions, fragments of the melody) keeping its own local statistical characteristics. The pitches and durations of musical notes are always produced in function of a PMMSG during the process of music stochastic generation. Through the use of n PMMSG in the process of melodic composition, the global melody will be the result of the interaction among (potentially) n local regions with their own melodic characteristics.

To build the PMMSGs, the user has to choose among those probability distribution of the state transition matrices built in the previous step (see Figure 4). The interface shows an area to select transition probability matrices (for pitches and durations) and also shows the content of the selected matrix (inferior area on Figure 4). In this visualization, the probabilities are represented by a set of shades of gray, which aids the user in the selection task. The shades of gray range from black, indicating a probability of 0%, to white, indicating a probability of 100%. To define a PMMSG, the user must to choose a pair of matrices, combining a transition table of pitches with a transition table of durations.



Transition probability matrices

Pitch transition probability matrices

- Matrix 0
- Matrix 1
- Matrix 2
- Matrix 3
- Matrix 4
- Matrix 5
- Matrix 6

Duration transition probability matrices

- Matrix 18
- Matrix 19
- Matrix 20
- Matrix 21
- Matrix 22
- Matrix 23
- Matrix 24
- Matrix 25

Created Models

- Model [1 , 4]
- Model [4 , 6]
- Model [6 , 10]
- Model [2 , 23]
- Model [30 , 23]

Selected transition probability matrix

0,00	0,00	0,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,00	0,00	0,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,00	0,00	0,00	36,36	63,64	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,00	0,00	13,33	6,67	6,67	73,33	0,00	0,00	0,00	0,00	0,00	0,00
0,00	12,50	41,67	0,00	0,00	33,33	12,50	0,00	0,00	0,00	0,00	0,00
21,82	20,91	12,73	0,00	0,00	0,00	10,91	0,00	5,45	9,09	10,00	9,09
13,18	36,56	26,49	12,79	0,39	1,29	1,94	2,33	1,16	1,29	1,29	1,29
14,15	23,68	28,22	22,06	9,04	0,44	0,44	0,92	0,26	0,00	0,31	0,48
7,87	23,82	19,99	17,12	20,25	7,21	0,25	0,22	0,84	0,87	0,84	0,73
9,91	18,58	18,27	17,23	14,70	15,88	2,30	0,80	1,01	0,55	0,29	0,49
12,06	28,02	19,96	12,32	10,31	12,25	2,32	1,09	1,00	0,84	0,79	1,04

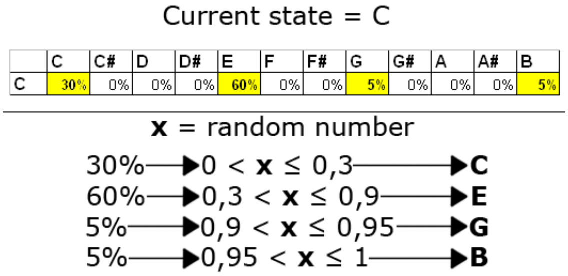
Figure 4. Graphical interface for user PMMSG definitions.

2.3. The Melodic Stochastic Synthesis Layer

In this layer, two Markov chains are generated containing N generated states, one of these chains defines the succession $\mathcal{SC}$ of pitches ($\mathcal{SC}^P$) and the another defines the succession of durations ($\mathcal{SC}^D$).

The generation of the two sets ($\mathcal{SC}^P(r)$) and ($\mathcal{SC}^D(r)$) above take the following parameters: a chosen PMMSG from the set of the user-defined PMMSGs, an initial pitch and duration values, a number N of notes to be generated and a MIDI instrument timbre. Thus, an iterative process is triggered in order to generate each one of the N states through the next follow steps:

1. to identify which row in the transition probability matrix that is related to the current select pitch/duration in order to define the next potential pitches/duration;
2. to define sub-intervals in the $]0, 1]$ interval which are proportional to probabilities generated in the previous step. Each one of these subintervals represented a possible pitch/duration to be generated from a current pitch/duration. This step is ilustred in Figure 5;
3. to generate a random number x from the $]0, 1]$ interval. In order to define the next pitch/duration to be generated, one should to check which sub-interval (defined in step 2) the value x belongs to.



Current state = C

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
C	30%	0%	0%	0%	60%	0%	0%	5%	0%	0%	0%	5%

x = random number

30% → 0 < **x** ≤ 0,3 → **C**
60% → 0,3 < **x** ≤ 0,9 → **E**
5% → 0,9 < **x** ≤ 0,95 → **G**
5% → 0,95 < **x** ≤ 1 → **B**

Figure 5. Example of the process of generation of a new pitch, from the current one and a transition proability distribution

At the end of this iterative process, we get a set of N pitches and a set of N durations. When combining the $i-th$ element from the set of resultant pitches to the $i-th$ element from the set of resultant durations, we obtain the $i-th$ musical note of the melody. So, going through the sets linearly (position to position) we obtain the final melody with N musical notes.

During the iterative process several PMMSG are used in order to avoid some monotony in the final melody. In order to perform the selection of the PMMSGs during each iteration, an utility cycle is define to each PMMSG. An utility cycle (uc) to some PMMSG Pg (uc^{Pg}) determines how many times this PMMSG is used into the iterations. At the beginning of each iteration a value to uc^{Pg} is defined from a random number between 1 and a "maximal utility cycle".

Definition A *maximal utility cycle* (muc) of a PMMSG Pg (muc^{Pg}) determines a maximal limit of iterations:

$$muc^{Pg} = \frac{N}{M} \tag{2}$$

where N is a total number of musical notes to be generated (informed by the user) and M is the total number of PMMSGs defined at the previous layer.

When a PMMSG reaches the end of his utility cycle, a new PMMSG is selected. This selection is carried out by observing which are the last states (pitch and duration) produced. The new PMMSG must have transition probabilities defined for these last states. In this way, we can analyze which PMMSG satisfies these conditions in order to chose randomly one of them to substitute the previous PMMSG.

3. Experiments

The use of a reactive multiagent model aims to produce a perturbation parameter to the system in order to dissociate the resultant melody from any user cognitive bias or aesthetic preferences.

The main goal of the experiment is to demonstrate that the model proposed here is able to generate a global melody with distinct intermediary and local characteristics, through the user interaction with a behavioural-interactive metaphor, rather than making use of the user musical knowledge. In these experiments, each running is unique and irreproducible, due the behavioural-interactive nature of the transition probability matrices generation and the stochastic nature of the process of melodic synthesis.

The multi-agent environment requires some parameters in order to build the structure of the environments and to initialize the composition process. Some examples of these parameters include:

- *Width and height of the environment*, define the dimensions of the environments.
- *Lines and columns*, define how much co-occurrence matrices the environments will be divided.
- *Number of agents* that will interact in each environment.
- *Minimal distance* that each agent have to keep from other agents.
- *Repulsion Multiplier*, represents the strength of repulsion, relative to the intensity with which the agents calculate his evasive behaviour.

Figure 6 illustrates a fragment of the interface showing the multiagent environments and some paths built by the agents during an interaction session.

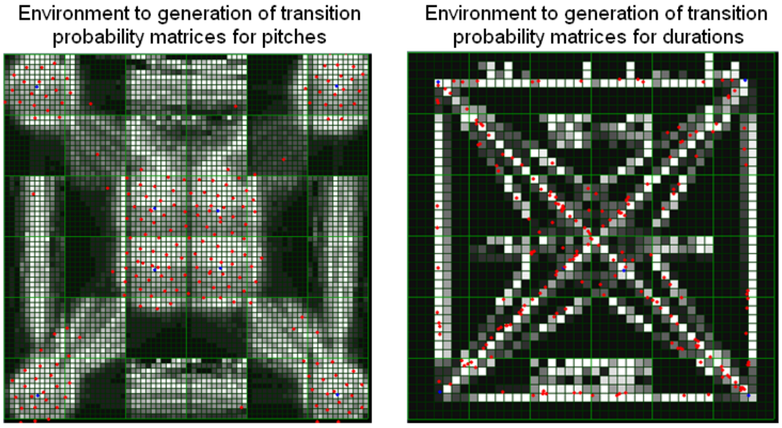
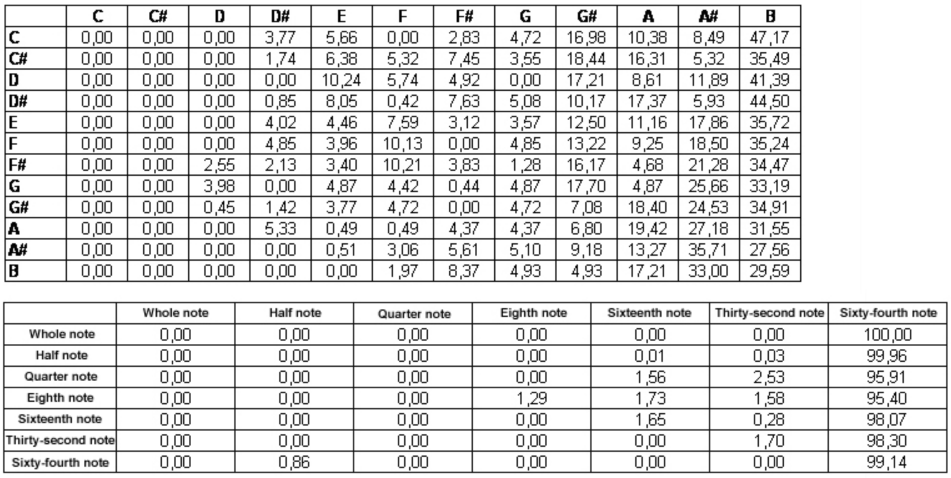


Figure 6. Example of the multiagent environment showing some resultant paths during the agents interactions. At left we have the transition probability matrix for pitches and at right, the transition probability matrix for durations. In each environment, each red dot represents an agent and each blue dot represents a target-point. The distinct shades of gray represent the frequency with which a cell is visited by some agent during the interactions within the environment

The value of some parameters and the number of Target-points defined by the user may produce a great density of Target-points and influence the agent global behaviour. This density will influence straightly the probability distributions in the environments and change "qualitatively" the emerging state transition matrices. Therefore, the stochastic nature of this process do not guarantee the quality of the generated probability transition matrices and do not suffer possible influence from the composer's aesthetic desires.

After the definition of the 36 transition probability matrices of pitches and 36 transition probability matrices of durations, we have defined some metrics related to melodic regions that present distinctive characteristics. For instance, we were interested in the concentration of high/low pitches and long/short durations. One of the generated PMMSG (Figure 7) has presented a great concentration of transition probabilities for high pitches and short durations, on privileging the generation of high notes of short duration.

Others PMMSGs (Figure 8) from the experiments, has presented a greater concentration of transition probabilities for low pitches and long durations, on privileging the generation of low notes of long duration.



	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
C	0,00	0,00	0,00	3,77	5,66	0,00	2,83	4,72	16,98	10,38	8,49	47,17
C#	0,00	0,00	0,00	1,74	6,38	5,32	7,45	3,55	18,44	16,31	5,32	35,49
D	0,00	0,00	0,00	0,00	10,24	5,74	4,92	0,00	17,21	8,61	11,89	41,39
D#	0,00	0,00	0,00	0,85	8,05	0,42	7,63	5,08	10,17	17,37	5,93	44,50
E	0,00	0,00	0,00	4,02	4,46	7,59	3,12	3,57	12,50	11,16	17,86	35,72
F	0,00	0,00	0,00	4,85	3,96	10,13	0,00	4,85	13,22	9,25	18,50	35,24
F#	0,00	0,00	2,55	2,13	3,40	10,21	3,83	1,28	16,17	4,68	21,28	34,47
G	0,00	0,00	3,98	0,00	4,87	4,42	0,44	4,87	17,70	4,87	25,66	33,19
G#	0,00	0,00	0,45	1,42	3,77	4,72	0,00	4,72	7,08	18,40	24,53	34,91
A	0,00	0,00	0,00	5,33	0,49	0,49	4,37	4,37	6,80	19,42	27,18	31,55
A#	0,00	0,00	0,00	0,00	0,51	3,06	5,61	5,10	9,18	13,27	35,71	27,56
B	0,00	0,00	0,00	0,00	0,00	1,97	8,37	4,93	4,93	17,21	33,00	29,59

	Whole note	Half note	Quarter note	Eighth note	Sixteenth note	Thirty-second note	Sixty-fourth note
Whole note	0,00	0,00	0,00	0,00	0,00	0,00	100,00
Half note	0,00	0,00	0,00	0,00	0,01	0,03	99,96
Quarter note	0,00	0,00	0,00	0,00	1,56	2,53	95,91
Eighth note	0,00	0,00	0,00	1,29	1,73	1,58	95,40
Sixteenth note	0,00	0,00	0,00	0,00	1,65	0,28	98,07
Thirty-second note	0,00	0,00	0,00	0,00	0,00	1,70	98,30
Sixty-fourth note	0,00	0,86	0,00	0,00	0,00	0,00	99,14

Figure 7. Example (1) of the PMMSG formed by the probability transition matrix of pitches (the top table), and the probability transition matrix of durations (the bottom table).

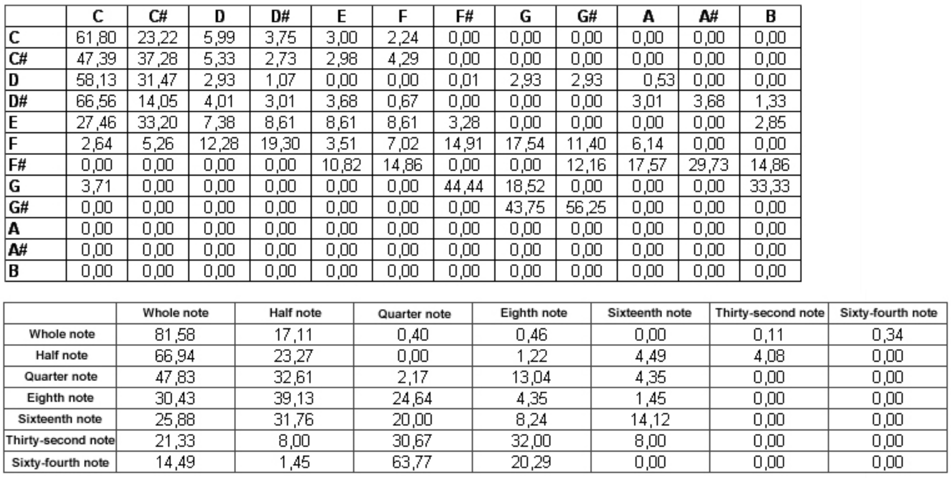
With these generated PMMSGs we supply the entry to the process of stochastic synthesis in order to obtain a melody in which it can be observed regions with those characteristics described above. The final melody is showed at Figure 9 in a "piano-roll" view (where the musical notes are represented by rectangles which length denote the duration and which height represents the pitch) of the resultant melody where these peculiarities are pointed.

Through this scenario was possible to realize that the choice of the probability tables with which the user builds the PMMSG, from which the final melody will be produced, influences but not define the final melody. So, due to the behavioural-interactive characteristics from the process of probability transition matrices generation and due to the stochastic characteristics in the melodic synthesis process, the final melody is disconnected from cognitive processes related to the composers musical domain.

4. Discussion and future work

In this paper, we have presented a model to automatically generate a melody from the organization of musical elements related to probabilistic spaces which are produced through the interaction of autonomous agents in a process inspired by behavioural-interactive metaphors. In a distinctive way from traditional approaches, in this model the composer does not think in terms of musical parameters to compose music, but in terms of behavioural-interactive phenomena and probability distributions. Also, this behavioural-interactive aspect allied to the multiple parameterizing possibilities is able to make many different melodic compositions arise (see [Behrends 1999]).

With this work we got some insights about the application of a model for musical synthesis under three different levels: *a global level*, defined through the PMMSG distribution along the synthesis process; *an intermediary level* (where melodic segments may present particular aspects), defined by the probabilistic characteristics instantiated into the PMMSGs; and, *a local level* (characteristics of each note), defined in function of the transition from the last generated note. These insights are related to the hierarchy



	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
C	61,80	23,22	5,99	3,75	3,00	2,24	0,00	0,00	0,00	0,00	0,00	0,00
C#	47,39	37,28	5,33	2,73	2,98	4,29	0,00	0,00	0,00	0,00	0,00	0,00
D	58,13	31,47	2,93	1,07	0,00	0,00	0,01	2,93	2,93	0,53	0,00	0,00
D#	66,56	14,05	4,01	3,01	3,68	0,67	0,00	0,00	0,00	3,01	3,68	1,33
E	27,46	33,20	7,38	8,61	8,61	8,61	3,28	0,00	0,00	0,00	0,00	2,85
F	2,64	5,26	12,28	19,30	3,51	7,02	14,91	17,54	11,40	6,14	0,00	0,00
F#	0,00	0,00	0,00	0,00	10,82	14,86	0,00	0,00	12,16	17,57	29,73	14,86
G	3,71	0,00	0,00	0,00	0,00	0,00	44,44	18,52	0,00	0,00	0,00	33,33
G#	0,00	0,00	0,00	0,00	0,00	0,00	0,00	43,75	56,25	0,00	0,00	0,00
A	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
A#	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
B	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00

	Whole note	Half note	Quarter note	Eighth note	Sixteenth note	Thirty-second note	Sixty-fourth note
Whole note	81,58	17,11	0,40	0,46	0,00	0,11	0,34
Half note	66,94	23,27	0,00	1,22	4,49	4,08	0,00
Quarter note	47,83	32,61	2,17	13,04	4,35	0,00	0,00
Eighth note	30,43	39,13	24,64	4,35	1,45	0,00	0,00
Sixteenth note	25,88	31,76	20,00	8,24	14,12	0,00	0,00
Thirty-second note	21,33	8,00	30,67	32,00	8,00	0,00	0,00
Sixty-fourth note	14,49	1,45	63,77	20,29	0,00	0,00	0,00

Figure 8. Example (2) of the PMMSG formed by the probability transition matrix of pitches (the top table), and the probability transition matrix of durations (the bottom table).

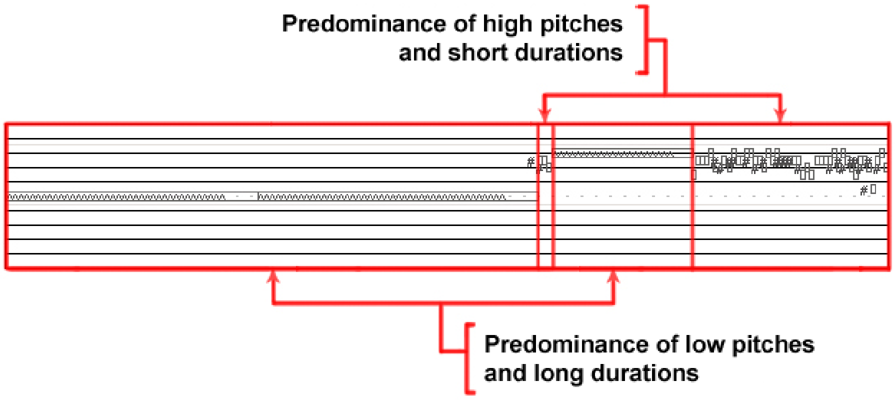


Figure 9. Example of a resultant melody (represented in a "piano roll" view), in which we can identify some melodic regions with particular characteristics.

of the melody global design (comprising the vision of Xenakis about the contemporary music [Xenakis 1992]). In this way, we intend to explore these global/local approaches through new multi-agent interaction models in order to experiment other configurations and organizational multi-agent structures. We can explore in deep our multiple layers of representation and to explore stochastic generation through higher order models of Markov chains in the process of melodic synthesis.

In future work we intend to evaluate the system proposed here as an environment of support for musical creativity, that is, not as an approach for synthesis of musical pieces, but as an environment capable of instigating the composer's creativity and generate creative material that can be worked by the user, within his/her own creative process. Since the melody generated in this environment is an indirect result of the interaction between the composer and a behavioral-interactive metaphor, we take as hypothesis that the environment promotes the emergence of creative insights, performing a dissociation between the melodic result and the user's musical knowledge. Thus, in new studies we also intend to investigate a way to evaluate this hypothesis, checking whether the environment is able to generate interesting melodic structures that would not be conceived by the composer, given his/her musical background.

References

Behrends, E. (1999). Música: Estrutura e Acaso. *Colóquio/Ciencias*, (24):17–24.

Brooks, R. A. (1986). A Robust Layered Control System for a Mobile Robot. *IEEE - Journal of Robotics and Automation*, 2(1):14–23.

Bryan-Kinns, N. (2004). Daisyphone: The Design and Impact of a Novel Environment-for Remote Group Music Improvisation. *ACM Symposium onDesigning Interactive Systems*, pages .135–144.

Burraston, D. and Edmond, E. (2005). Cellular Automata in Generative Electronic Music and Sonic Art: a Historical and Technical Review. *Digital Creativity*, 16:165–185.

Cruz, M. A. S. (2001). Técnicas de Computação Sônica Aplicadas ao Design de Software Musical. Master's thesis, State University of Campinas.

Farbood, M. and Schoner, B. (2001). Analysis and synthesis of palestrina-style counterpoint using markov chains. In *Proceedings of International Computer Music Conference*, Havana, Cuba.

Fischman, R. (2003). Clouds, pyramids, and diamonds: Applying schrödinger's equation to granular synthesis and compositional structure. *Computer Music Journal*, 27(2):47–69.

MacGee, D. and Schaffer, J. W. (1997). *Knowledge-Based Programming for Music Research*. A-R Editions, Inc.

McCormack, J. (1996). Grammar-based music composition. *Complex International*, 3.

Meyn, S. P. and Tweedie, R. L. (1993). *Markov chains and stochastic stability*. Springer-Verlag.

Mezencio, R. (2002). Implementação do Método de Campos Potenciais para Navegação de Robôs Móveis Baseada em Computação Reconfigaráve. Master's thesis, Departamente de Ciências de computação e estatística - Instituto de Ciências matemáticas e de computação - universidade de São Paulo.

Miranda, E. R. (2000). The art of rendering sounds from emergent behaviour: Cellular automata granular synthesis. *euromicro*, 02:2350.

Oliveira, L. F. d. (2003). As contribuições da ciência cognitiva à composição musical. Master's thesis, Faculdade de Filosofia e Ciências da Universidade Estadual Paulista Júlio de Mesquita Filho.

Papadopoulos, G. and Wiggins, G. (1999). AI methods for algorithmic composition: A survey, a critical view and future prospects. In *Proc. AISB'99 Symp. Musical Creativity*, pages 110–117.

Xenakis, I. (1992). *Formalized Music: Thought and Mathematics in Music*. Pendragon, revised edition.