INMAMUSYS: Intelligent Multiagent Music System

Miguel Delgado, Waldo Fajardo, Miguel Molina-Solana*

Department of Computer Science and Artificial Intelligence, Universidad de Granada, Daniel Saucedo Aranda s/n, 18071 Granada, Spain

Abstract

Music generation is a complex task even for human beings. This paper describes a two level competitive/collaborative multiagent approach for autonomous, non-deterministic, computer music composition. Our aim is to build a high modular system that composes music on its own by using Experts Systems technology and rule-based systems principles. To do that, rules issued from musical knowledge are used and emotional inputs from the users are introduced. In fact, users are not allowed to directly control the composition process. Two main goals are sought after: investigating relationships between computers and emotions and how the latter can be represented into the former, and developing a framework for music composition that can be useful for future experiments. The system has been successfully tested by asking several people to match compositions with suggested emotions.

Key words: Computer music, Music composition, Multiagent systems, Rule-based system

1. Introduction

Surely music is one of the most difficult human disciplines. It requires creativity, specific knowledge and eventually, some manual abilities. It is commonly said that music is something humans do, but something we do not understand. There are several mechanisms involved in music and unfortunately, many of them are still unknown. Others are so complex that we cannot manage them with current computer tools.

On the other hand, Computer Science has suffered a huge evolution in just a small period of time. However, despite all the great applications and problems solved during the last decades, there are many problems that computers are unable to deal with nowadays.

Among others, we can point fuzzy representation of abstract concepts and, in general, dealing with feelings and emotions. In fact, cognitive processes involving reasoning, knowledge and experience are hardly represented in a computer and are current hotspots for Artificial Intelligence.

Music is a good example of applied AI related with those topics. It has been demonstrated that music composition is a hard task even for humans, so using machines appears as a very interesting and fascinating area of research. AI techniques are going to be applied to the musical domain with the aim of understanding human musical abilities. Because there are so many challenges to deal with, a bottom-up approximation is required for solving all of them in a modular way.

This paper is organized as follows. In Section 2, we examine some previous works done in the field. Section 3 deals with the architecture of the proposed system. In Section 4, we discuss design and implementation of system components. Section 5 presents an evaluation method and summarizes some users' impressions about the output; and in the last section, future work to be done is presented and discussed.

2. Background

Since first computers were developed many people have tried to apply them to musical tasks.

There are two main classes in which computer music projects could be classified: analysis and composition. The first one consists on extracting information from the music itself (or the associated data) in order to learn some rules, or go to a model that describes the concrete examples. Because this is not the main field of this paper, so we are not going to go further. However, [1], and [2] could be reviewed for more information.

Composition is about generating new music from the rules. In fact, is doing the process in the other way:

*Corresponding author

Email addresses: mdelgado@ugr.es (Miguel Delgado), aragorn@ugr.es (Waldo Fajardo), miguelmolina@ugr.es (Miguel Molina-Solana)

'from rules to music' instead of 'from music to rules'. According to [3], the final objective of most of the compositional prototypes is to demonstrate that standard musical techniques could be handled by computer programming, and also to validate generative music theories.

Initial approaches in algorithmic composition consisted on randomly selecting notes (mainly pitch and rhythm) with some constraints in order to generate compositions. This vision produced limited results but was a great point of departure for later works.

Experts systems has been widely used to compose music. Rules concerning pitch, duration and volume have tried to apprise the knowledge involved in human composition. [4] introduced an early example of rule system. Many works on computer music generation are based on the approach that the composition rules are specified by the composer. So that, an expert is needed in order to give all the knowledge. The problem here is that Music Theory is not as formal as it should be to be easily represented in a computer. However, [5] proposed an interesting model to generate music using grammars.

Another approach to compose music with computers is to use genetic algorithms. Genetics have been successfully applied to several problems with difficulty in defining the solution process, or where searching a huge solution space is needed. Composition falls into this class of tasks, and some works in this direction can be found in [6] and [7].

Although many advances have been done during last decades in Computer Music in general and in algorithmic composition in particular, it is true that the greatest moment was when [8] presented EMI Project.

Many prototypes, throughout the years, have demonstrated that computers algorithms cannot be compared with human minds. Machines just produce very simple compositions in a quite mechanical manner.

Computer programs that perform music with proper stylistic considerations (like a human expert would) are very scarce and only work for a few examples and in very specific domains may be found. The main problem is that algorithms rarely deal with feelings. And music without emotions is unworthy and not natural.

[9] expressed his conviction that the unique way for creating a machine whose creations transmit something to listeners, should start by simulating emotions in computers. [10] addressed this topic (expression in computers) from a musicological point of view, while [11] proposed a cognitive model for understanding melodies.

This topic is a current hotspot in computer music, and many groups are working on it right now. [12], [13], [14], [7], [15] and [16] can be mentioned as some relevant examples. For further information about composing music with computers, [17] and [18] works should be reviewed.

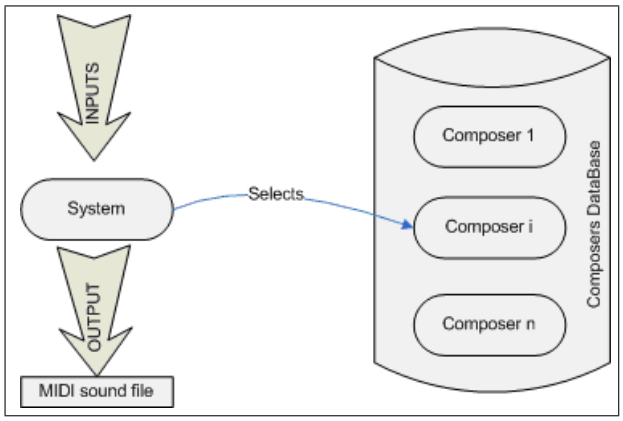

Figure 1: First level architecture

In recent years, agent paradigm has become quite common. Almost everybody is using it for everything. The reason is that agents are a very powerful way of implementing distributed AI. Indeed, they can be combined with others tools, such as rule-based systems, case-based systems or searching algorithms. Actually, Agent Theory just defines interactions between agents, not how they are internally built. Because of that, we can have a multiagent system where agents can be implemented with just an algorithm, using CBR techniques, rules, genetics... Interesting work related with multiagent systems can be found in [19] and [20].

3. Architecture

The architecture we propose in this paper is a two-layer multiagent system. The current application of multiagent systems in real-time environments is an area of increasing interest. In general, multiagent systems are an appropriate approach for solving inherently distributed problems, whereby clearly different and independent processes can be distinguished.

The first level is the competitive one, where agents (called *composers*) compete among themselves to be the one chosen for composing. This layer allows us to make an initial separation between composition styles. We think it does not make any sense to have a one-for-all agent, so a collection of simple agents specialized in some task is proposed. Each composer announces its abilities, and the system, according with user inputs, selects the composer that better fits. See Fig. 1.

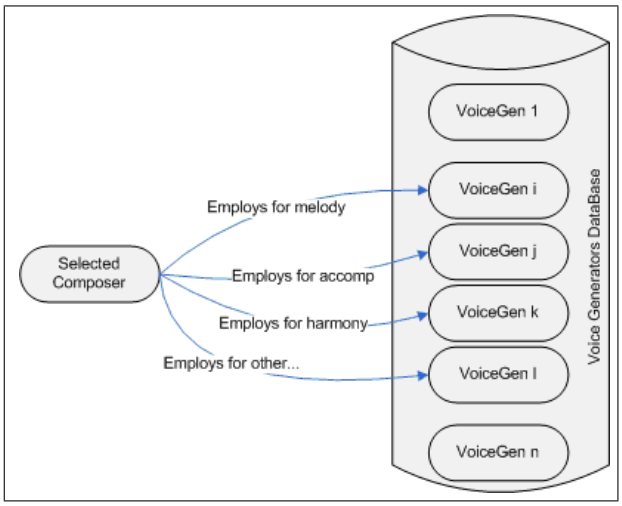

Figure 2: Second level architecture

This agent chooses, between other parameters, rhythm, number of voices, and instruments to be used in the compositions. However, the selected composer only acts as a director, being useless on its own. Going further with the modularization principle, the composer agent finally asks some others agents from the second level for their collaboration in order to get a solution [1].

This second level contains auxiliary agents that collaborate between them, so this layer should be understood as a collaborative level. We call these agents *voice generators*, because mainly that is their job. See Fig. 2.

In general, there are just three voices in normal compositions: melody, accompaniment and harmony. However, the system presents no limitations in relationships between agents. A composer agent can employ as many voices as desired, and in the other hand, several composers could use the same voice generator.

When some intelligence is needed, agents are to be designed as intelligent rule-based systems. Otherwise, they just implement a simple algorithm. At the end, we have a big set of simple agents that together, manage to find a whole solution for the problem.

4. Design

INMAMUSYS current prototype has several composers that generate music in different ways: from just a random composer to more elaborate ones where aesthetic principles are mainly searched.

[1] Solution in this context means just a composition that the system offers as output, without evaluating its quality

An important design principle that is going to guide our system is creating a tool that can be used by everyone. Most prototypes present a very complex interface that only experts can understand, an even for them, it is very annoying to fill a huge amount of data in order to get a composition. In fact, when people speak about music, they do not usually use technical terms such as granularity, rhythm or tonality; they use emotions, feelings and abstract concepts. Actually, they are not speaking in low level, but in a high level, where technical parameters do not exist. Keeping that in mind, our aim is the users ought to be faced with a friendly interface with only a few questions; and not really difficult ones, just questions about the wished music style. In other words, a high level music interface is to be introduced.

In fact, we are integrating into the system itself the knowledge about how to use tonality, rhythm and instruments in order to get, for instance, a sad music.

Until now, this task has been left in the hands of humans. We propose to go further providing the machine with the suitable resources it should use to compose in a certain way. The human does not have to worry about dozens of parameters and how they should be combined to get a certain result.

Java has been chosen as programming language due to its object oriented style and ease of use. *jMusic* library is used for music representation and processing. This library is described by [21].

4.1. Knowledge representation

We have made use of several ways to represent information. Firstly, because most agents are designed as rule-based systems, we need to represent some information through IF-THEN rules. They are implemented in XML in order to achieve an universal way of representation. Fig. 3 shows the XML schema for rules in the DTD language. Basically, a knowledge base is a set of several rules. A rule has two parts: IF and THEN. IF part consists of some attribute-value pairs; and THEN has a list of consequents, that can be viewed as actions to be done. We have decided to employ several rules subsets in order to respect the modularity, and to make the inference engine's job easier. Higher performance is also reached with this approach.

To write code, we use the representation that *jMusic API* offers us. This is an object oriented one, very close to normal Western classical music representation. There are manuals and several examples in jMusic website (http://jmusic.ci.qut.edu.au). As agents are developed for a concrete musical style or concrete task, there is obviously a lot of heuristic information about how to compose coded in each agent. Agents are implemented

```
<!ELEMENT knowledgebase (rule)+>
<!ELEMENT rule (if, then)>
<!ELEMENT if (atrib,value)+>
<!ELEMENT atrib ANY>
<!ELEMENT value ANY>
<!ELEMENT then (consec+)>
<!ELEMENT consec ANY>
```

Figure 3: XML schema in DTD language for representing rules in the system

with certain parameters (that only an expert knows) in order to accomplish its goals. This kind of knowledge should be minimized and moved to a unique and standard repository, in order to obtain an independent knowledge database. However, many times it is not easy to code procedural information in a data format, so we should be aware that much knowledge about the topic is into the composition process itself.

System output is a sound file in MIDI format. This representation is preferred rather than others audio formats such as *wav* or *mp3*, because it is easier for symbolic manipulation. Even more, MIDI could be easily converted into the others two (if needed), but the opposite is not true. Eventually, a music sheet could be generated if needed.

4.2. Composition Agents

Composer agents, as we have previously said, are located in the first layer of the architecture. They compete with others in order to be elected for composing. Basically, all composers are in charge of defining the number of voices the composition will have, instruments to be used, the measure and tonality. They also indicate which second level agents should be used for generating voices. In other words, composers act as directors. In current system there are four implemented composers: Muzak, Dark, Scales and Random.

Muzak Composer generates music in a Muzak[2] way. This kind of music, also called ambient or elevator music, could be described as a soft and quiet one. Brian Eno wrote "Ambient Music must be able to accommodate many levels of listening attention without enforcing one in particular; it must be as ignorable as it is interesting". From this point of view, ambient music does not need to be very elaborate or complex. Its only requisite is to be pleasant and agreeable to the human ear.

Dark composer aim is to compose music that provokes fear in listeners. To do that a set of suitable resources is used. Dissonances and diminished fifth interval (known as tritone) are heavily employed by this agent. This interval is called *diabolus in musica* (the Devil in music) and has been historically avoided. Great distances between voices and deep bass chords are also useful resources that are employed.

Scales Composer is quite simple. It just produces some ascendant and descendent scales in random tonalities and modes. The mixing of several voices doing the same in different moments, velocities and tones produces interesting effects.

Finally, we introduce the Random Composer with its two variants: the *differential*, and the *independent*. In the first one, intervals are randomly generated, so a note pitch depends on previous note. The other option is to randomly generate pitches, so a note is independent of the rest. Durations are also generated with a random generator. As the reader can imagine, a composer acting this way produces chaotic compositions without any internal coherence. So that, chaos is the tag that better defines the music this composer generates.

4.3. Voice generators

Voice Generators are in charge of producing sounds with different volumes, durations and intonations for every melodic line.

Many kinds of voice generators could be implemented, but the main ones fall into any of the following classes: melody, harmony, accompaniment and drums. However, the system architecture is flexible enough to allow any other desired voices. In fact, the architecture does not care about the semantic meaning of a certain melodic line.

4.3.1. Harmony generator

Harmony generator makes use of a set of rules that indicates which tonal movements are allowed. In this way, chord progressions can be generated. With this, we have the skeleton of compositions. We can understand this as a Markov chain model. In Fig. 4 we can see the set of rules in the current implementation, that correspond with Fux's counterpoint rules ([22]).

4.3.2. Melody generator

Melody generators produce the melody of compositions. Current design uses a big set of motives which are one measure long. A motive is randomly selected in each measure and put into the melody voice. However, preliminary tests reveal that acting this way, compositions will not be coherent, in the sense that they will not seem as a unique entity, but rather, like little pieces stuck together.

[2]MUZAK is the name of a company specialized in this kind of music. People often refer to ambient music in this way.

IF actual_degree = I THEN next_degree = I
IF actual_degree = I THEN next_degree = II
IF actual_degree = I THEN next_degree = III
IF actual_degree = I THEN next_degree = IV
IF actual_degree = I THEN next_degree = V
IF actual_degree = I THEN next_degree = VI
IF actual_degree = I THEN next_degree = VII
IF actual_degree = II THEN next_degree = II
IF actual_degree = II THEN next_degree = IV
IF actual_degree = II THEN next_degree = V
IF actual_degree = II THEN next_degree = VI
IF actual_degree = II THEN next_degree = VII
IF actual_degree = III THEN next_degree = II
IF actual_degree = III THEN next_degree = IV
IF actual_degree = III THEN next_degree = V
IF actual_degree = III THEN next_degree = VI
IF actual_degree = IV THEN next_degree = I
IF actual_degree = IV THEN next_degree = II
IF actual_degree = IV THEN next_degree = III
IF actual_degree = IV THEN next_degree = V
IF actual_degree = IV THEN next_degree = VI
IF actual_degree = IV THEN next_degree = VII
IF actual_degree = V THEN next_degree = I
IF actual_degree = V THEN next_degree = III
IF actual_degree = V THEN next_degree = IV
IF actual_degree = V THEN next_degree = VI
IF actual_degree = VI THEN next_degree = II
IF actual_degree = VI THEN next_degree = III
IF actual_degree = VI THEN next_degree = IV
IF actual_degree = VI THEN next_degree = V
IF actual_degree = VII THEN next_degree = I
IF actual_degree = VII THEN next_degree = III
IF actual_degree = VII THEN next_degree = V
IF actual_degree = VII THEN next_degree = VI

Figure 4: Rules used by harmony generator to generate chord progressions

The trick here consists in to randomly select (from the whole set) a subset of motives to be used in each execution. To have a small number of motives for each execution makes the composition appear as a whole, giving the impression of melodic phrases. In fact, there is no structure at all to be followed, but the repetition over and over again of similar motives induces the listener to think so. Employing this simple trick the output seems to be organized in phrases (but as seen there is no more than random selection) and the impression is quite realistic.

Moreover, due to the fact that the subset is randomly selected, it is different for each execution. This is an important point because it produces a great variety between two compositions, even thought when they are generated with same inputs.

4.3.3. Drums generator

Drums generators are in charge of generating a continuous rhythm. There are several composers that can be classified within this class. They implement different rhythms: some offers more density, others are lighter; ones use various sounds (drums, snare, cymbal, hat...), others just use a simple drum. At the end, it is the first level agent (the composer) the one in charge of selecting

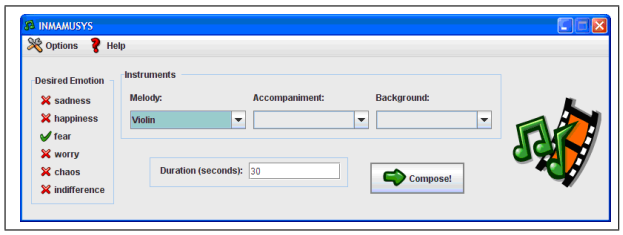

Figure 5: INMAMUSYS's user interface

the appropriate Drum Generator (or even none of them) for each occasion.

4.4. User interface

Our main aim was to develop an easy and simple interface that anyone could use without problems.

Before, we talked about the purpose of implementing an easy-to-use system. The objetive here is presenting to the users an interface that anyone can use, even if they are not experts. To do that, we need to include within the system all the information related to technical parameters. This is a great deal because the matching between emotions and music parameters is not direct, and to define parameters themselves is definitively not a trivial task.

The interface (see Fig. 5) is very dynamic and not fixed because its contents depend on the composers implemented in the system and the capabilities they announced. The form only contains terms that agents have declared in some rule. With this, we avoid unexpected inputs and assure the system will always be able to give us a valid output. Also, it lets us add, replace, or modify agents when any problems appear, making changes easier.

In addition, we have added the possibility of selecting which instruments should be used in each voice. However, if we do not specify an instrument, the system selects one from those it thinks is a better fit. Not all instrument combinations sound good, and even if they do, it is possible that they are not compatible with the style of the melody. These options exist in order to demonstrate the complexity of the knowledge involved. Choosing rhythm, instruments, tonality or granularity are not trivial tasks, because they should be considered all together, and related with some other concepts. By allowing selection of instruments, users can test if they are able to find combinations that really make great compositions, beating system choices.

4.5. The composition process

In this subsection, the composition process of Muzak Composer is described to illustrate how a composer runs

because the process is quite similar for the rest of composers.

When Muzak composer is selected by the Manager Agent, it firstly decides the tempo of the composition attending to user inputs. The mapping between linguistics tags and exact bpm (beats per minute) is done by using a normal random number generator with given mean and variance. The means values for each tag have been statistically obtained from both a set of classical sheet music and experts' feedback.

As said before, composer agents are also in charge of selecting the number of voices to be used, and the second level agents are responsible for their generation.

To begin with, the harmony generator is executed, producing a chord progression which is the skeleton of the composition.

After that, it is the melody generator's turn. For this agent, and in the current implementation, a motive database is decided to be used (as commented before), as well as employing one of its elements in each measure. Even thought there is no kind of structure of phrases, or a grammar to be followed, the global composition seems to have an internal structure. This effect is suggested by frequent motive repetitions.

Next step is to execute the accompaniment generator. Goal here is to complete the harmonization of the composition. Accompaniment generator introduces some new notes and completes chords. It could just be a note every measure, an arpeggio or perhaps something more elaborate.

It is important to indicate that melody and accompaniment are generated always in C major. This is not a handicap and greatly facilitates the task. However, it is necessary to transpose them to the correct position in each measure, according to the tonality and the current grade. Also, the note in the first time of each measure must be accentuated. It is up to voice generators to assure that.

5. Experiments and evaluation

The evaluation of any musical work is a complex task and often comes down to individual subjective opinion. It depends not only on formal aspects but also on some stylistics ones. Because of that, it is hard to empirically evaluate music compositions, and therefore it is difficult to evaluate the effectiveness of a computer music composition system.

Many metrics could be developed, but all of them will fail as at the end, music (understanding the term as much more than just a chain of sounds) cannot be reduced to a number.

Due to this difficulty, many authors will typically conclude their papers with a vague comment such as "compositions generated by the system are quite impressive and very promising" or "sometimes, melody seems to be a bit simple and unelaborated; but many, results are very human like". However, this is unsatisfactory for two reasons: first, evaluating the music produced by the system reveals little about its utility as a compositional tool; and second, qualitative and subjective evaluation by the designers of the system reveals little about the value of the tool to other composers ([3]).

This author affirms that there is an historical malaise in adopting suitable evaluation procedures for judging the degree to which the aims have been satisfied. As we agree with this assertion, we have also proposed an evaluation method to assess compositions in the same way that they are usually appraised: through audience reactions and critical reviews. So that, a short test has been design to carry out that task.

5.1. Experiments

To begin with, we asked some people for listening and evaluating some examples generated by INMAMUSYS. Most of them could not believe that a machine was the real author. For us, this is an important point because capturing the essence of a human's composition method is a great deal. This fact makes us also think that our system could have passed, in some sense, the Turing test.

The second part of the proposed evaluation method was more formal and consists in testing what emotions were induced on listeners when listening to some compositions. The objective was to probe whether compositions generated by the system provokes in listeners emotions such as those that guided the composition process and were given as inputs to the system. This test will give us a measure of how the system is able to compose music that successfully matches user emotional requests.

Four compositions were generated with different input values. Songs A, B, C and D are respectively generated by using the inputs worry, happiness, chaos and worry again (this input is used twice). System mapped these emotions into the use of Dark, Muzak, Random and Dark composers. The four examples were presented to the listeners without giving them information about the origin of the songs, and the listeners were asked to select emotions that better fit what they listen. Users could tick as much tags as they want from the list: sadness, happiness, fear, worry, chaos and indifference. Twenty people, involving a huge range in musical ex-

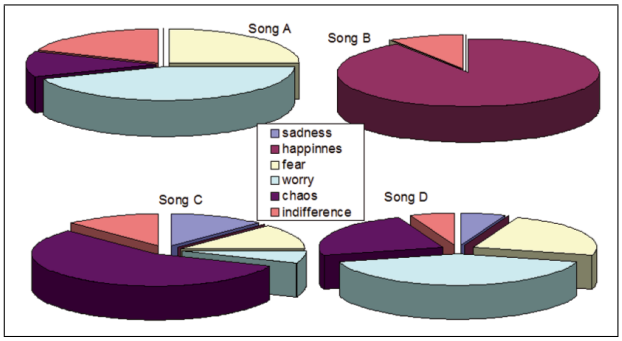

Figure 6: Results from the test: Feelings that the four tested songs provoked in listeners

pertise, participated in our experiment by answering the questions.

Results can be found in Fig. 6. We were really surprised by them because we did not expect such a clear confirmation, even more when the emotion-composer matching mechanism is quite primitive. In all the four cases, the most repeated options is the right one (the one that was given as input to the system). It is true that the list of potential options is small, (we would like to make it bigger the next time), but it is also clear that differences between some tags are very small, so that it would have been easy to get some wrong answer.

It is also noticeable, that many people (12 out of 20) point out that the first song and the last one were very similar. They are definitively different but these answers make us think that Dark composer outputs do not cover the whole area at the solution space that we supposed, but just a little zone. We think this should be corrected in the future, because is important to accomplish the objective that all the compositions from the same inputs fall in the same area, but it is also crucial to maximize this area, not just focusing in a point.

To conclude, we would like to comment that an equilibrated system has been developed: on the one hand, there is a great variety between different executions; on the other, we have managed that each composition sounds as a whole not like several pieces stuck together. As said, the whole solution space for a tag (e.g. fear) is not completely covered, but at least we can say that almost all the system outputs are classified (by humans) under the right tag.

6. Conclusions and future work

Composing music is a very complex process that involves many disciplines and tasks. In this paper we have presented a new approach for composing music using computers. Because there are so many challenges to deal with, a bottom-up approximation is required for solving all of them in a modular way. We are interested in building a framework for successfully composing music that provokes some feelings in listeners.

In order to achieve this goal, we have proposed here a two-level architecture that successfully deals with the complex problem of music composition, so that, a huge problem can be broken into smaller tasks. This approach makes use of several agents and rule-based systems. It also permits that users were provided by an easy-to-use interface that hides all the complexity of music composition. Even more, inputs for this interface are emotional inputs from the users, so that we are able to address the problem of music expressiveness.

Several design decisions that affect the system were taken and we have explained them. We have focused in knowledge representation because it is a main topic. In that section we also describe different agents, as well as the role they play in the whole system. At the moment, just four kinds of composers have been developed: Dark, Muzak, Scales and Random.

Finally, results obtained after evaluating the current system are showed. This evaluation has been carried out with an experiment and in a formal way. Even though the system is in a quite early stage of development, results are promising enough to encourage us to continue working with this framework.

6.1. Future work

The current prototype is quite rigid in the way it deals with user inputs. By now, we are just using classic rule matching for selecting agents, and that is a poor approach for dealing with fuzzy concepts, such as emotions and musical terms. So that, including fuzzy logic in the inference system is probably the first modification to be done. This change would requiere major modifications in the inference engine, as well as the definition of fuzzy domanisn and liguistic tags. However, we think it would be worth the effort.

Secondly, we would like to develop more composers in order to get a bigger collection of these kinds of agents. Our aim is to obtain as much diversity as possible not only to compare differents algorithms and compositional mechanisms, but for implementing and testing new ideas. These ideas could be related with new theories of musical styles or cognitive processes. Even more, as the set of composers growths, we will be able to reproduce a wider range of human emotions.

Another interesting project would be to develop a module to automatically obtain composition rules. In

the current prototype, this knowledge is given by humans, and coded into the agents. It would be very powerful if the system could analyze a music sheet, then to extract some rules, and thus to compose according to them. The system will win a great flexibility with this ability. Not in vain, this one is the key idea in Cope's system, and an interesting hotspot in current data mining. Doing this, the system would be quite complete, in the sense that it would include both composition and analysis tasks.

As seen, there is plenty of room for researching in the computer music area in general and in composition in particular. At our department, we feel computer music is a great opportunity for modern Artificial Intelligence. We have developed this project with the aim of it being used as a framework for future experiments and works.

We also believe that Expert Systems and Agent Theory have many things to say in this field. Composing music is a huge problem that should be divided into several tasks, and it definitively needs intelligence to be done.

Acknowledgements

This research has been partially supported by the Spanish Ministry of Education and Science under the project TIN2006-15041-C04-01.

References

[1] C. Anagnostopoulou, G. Westermann, Classification in music: A computational model for paradigmatic analysis., in: International Computer Music Conference, 1997, pp. 125–128.

[2] M. Balaban, The music structures approach in knowledge representation for music processing, Computer Music Journal 20(2) (1996) 96–111.

[3] M. T. Pearce, D. Meredith, G. A. Wiggins, Motivations and methodologies for automation of the compositions process, Musicae Scientiae 6 (2) (2002) 119–147.

[4] A. Friberg, Generative rules for music performance: A formal description of a rule system, Computer Music Journal 15 (2) (1991) 56–71.

[5] F. Lerdahl, R. Jackendoff, A generative theory of tonal music, Cambridge, Mass: MIT Press, 1983.

[6] M. Marques, V. Oliveira, A. Rosa, Music composition using genetic evolutionary, in: 2000 Congress on Evolutionary Computation, 2000, pp. 714–719.

[7] E. R. Miranda, At the crossroads of evolutionary computation and music: Self-programming synthesizers, swarm orchestras and the origins of melody, Evolutionary Computation 12 (2) (2004) 137–158.

[8] D. Cope, Experiments in music intelligence, Madison, WI: A-R Editions, 1996.

[9] M. Minsky, A conversation with Marvin Minsky. Understanding musical activities: Readings in AI and music, Menlo Park: AAAI Press, 1991.

[10] L. Meyer, Emotion and meaning in music, Chicago, IL: University of Chicago Press, 1956.

[11] E. Narmour, The analysis and cognition of basic melodic structures: The Implicacion-Realization model, Chicago, IL: University of Chicago Press, 1990.

[12] H. Kiendl, T. Kiseliova, Y. Rambinintsoa, Use of fuzzy strategies for interpretation of music, Journal of Multiple-Valued Logic and Soft Computing 12 (1) (2006) 149–169.

[13] R. L. de Mantaras, J. L. Arcos, Ai and music. from composition to expressive performance., AI Magazine Fall (2002) 43–58.

[14] Q. Zhang, E. R. Miranda, Towards and evolution model of expressive music performance., in: Sixth International Conference on Intelligent Systems Design and Applications, ISDA'06, 2006, pp. 1189–1194.

[15] G. Widmer, The musica expression project: A challenge for machine learning and knowledge discovery, in: 12th European Conference on Machine Learning, EMCL 2001, 2001, pp. 603–614.

[16] A. Wieczorkowska, Towards extracting emotions from music, in: Intelligent media technology for communicative intelligence, IMTCI, 2004, pp. 228–238.

[17] D. Cope, Computers models of musical creativity, The MIT Press, 2005.

[18] E. R. Miranda, Composing music with computers, Oxford, UK: Focal Press, 2001.

[19] P. M. Todd, G. M. Werner, Frankensteinian methods for evolutionary music composition. Musical networks: Parallel distributed perception and performance., Cambridge, MA: MIT Press/Bradford Books, 1999.

[20] R. D. Wulfhorst, L. Nakayama, R. M. Vicari, A multiagent approach for musical interactive systems, in: Second International Joint Conference on Autonomous Agents and Multiagents Systems, 2003, pp. 584–591.

[21] A. R. Brown, A. C. Sorensen, Introducing jmusic, in: Australasian computer music conference, 2000, pp. 68–76.

[22] W. Piston, Counterpoint, W.W. Norton, 1947.