An Open Architecture for a Musical Multi-Agent System

Rodolfo Daniel Wulfhorst, Luciano Vargas Flores, Lauro Nakayama
Cecília Dias Flores, Luis Otávio Campos Alvares, Rosa Maria Vicari

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{wulfhors,lvf,naka,dflores,alvares,rosa}@inf.*ufrgs.br*

Abstract. *The present work proposes a model for a Multi-Agent System capable to deal with music. We believe that many Computer Music problems could be approached through such a Musical Multi-Agent System. Our proposal is based on a community of agents that interact through musical events (MIDI), simulating the behavior of a musical group. As a case study we have implemented a rhythmic accompaniment system. "Listening" to each other, the agents were able to play their instruments in synchronism.*

1. Introduction

Considerable research has been made focusing the interaction between music and computers. However, few works were published applying the concept of agents to this subject. In such few cases, agents were application-dependent (Aeken 2000; Trajano 2000). The aim of this proposal is the analysis, design and implementation of an open model for the communication between musical agents. This model will be validated through the implementation of a rhythmic accompaniment application.

The underlying motivation is the hypothesis that many musical problems may have a computational solution through a community of agents that would communicate using music as their language, like in an ensemble. Thus, the overall idea is to propose an adequate model for inter-agent communication through musical events or signs. To implement this communication we chose the MIDI protocol, since it is used to transmit musical information as musical events (MIDI Manufacturers Association, 2000).

Finally, we will present a case study involving rhythmic accompaniment. Unfairly this problem has been treated as secondary in many researches of Artificial Intelligence applied to Musical Perception, since it is the most basic component of music. Our model is based on the algorithm proposed in (Dannenberg 1984) for rhythmic accompaniment.

2. MMAS – An Open Musical MAS

To guarantee its working, a Musical Multi-Agent System must agree to some minimum requirements. Perhaps the most important of them may be the *delay time*. Like in an ensemble, a certain time is needed to propagate sound information that has been sent from one agent to the group. It must be guaranteed that a message does not take more than a determined time to reach its destination. If not, synchronism between the agents will be compromised. Thus, we chose a *blackboard* architecture to implement our communication mechanism, in which agents share the musical events they generate. We opted for this solution in detriment of its alternative, which would be message exchanging, because of such communication model's intrinsic overhead.

Another requirement concerns *processing power*. Since our system is a real-time application, each message handling process must be fast and economical. To simplify the problem, we took the *MIDI protocol* as inter-agent communication language. The MIDI protocol was defined considering timing aspects of musical execution, even in situations where processing and communication resources are limited. Moreover, this represents a proper alternative for the implementation of Musical Perception models of more abstract levels, as rhythm, harmony, melody, performance and expression. Since our system consists of human and artificial agents, its important that all agents have the same expression possibilities. Taking into account that the human agent is limited to a MIDI instrument, we discarded the choice for any higher-level language. So the agent should itself be capable of abstracting higher-level information according to its needs.

Our proposal is to make possible the implementation and research of musical algorithms without a conceptual separation between human and computer. Our intentions, in conceiving the MMAS (Musical Multi-Agent System), is to abstract the notion of who is acting in the system, and simply create a *community of musical agents*. Conceptually, in such architecture, programming won't be directly focused on the solution, by a computer, of a human's problem. Instead, it will be focused on the adequate interaction among the agents in the community. In this way, it is expected from the *emergent behavior* phenomenon, typical of Multi-Agent Systems, to represent itself the solution for the musical problem. Hence, in the context of our proposal, the human being will not more be seen as external to the system and requesting solutions from this one, but as another member of the community, acting inside of it. Thus, the solution of musical problems will be next to the one that happens in reality, for example in cases as of musicians playing in a musical group.

2.1. MMAS' Architecture

To satisfy the already mentioned requirements, our model is comprised by a group of agents and a blackboard (see Figure 1). The amount of agents present in the community, as well as their parameters' configuration (autonomous or interface behavior, and priority), are freely defined. Such freedom contributes for the open quality of the proposed architecture.

Agent distribution occurs between processes on the same computer. Since it is a real-time application (delays must seem worthless to the ear), agents will run locally. Process distribution on the same processing unit represents, however, another challenge. Requested times for each agent to execute its processing cycle have to be minimum, so that they are not perceptible. Resulting consequences of this delay will be part of the system's evaluation. Compensation mechanisms will be studied and implemented in future works. Currently, we adopt a blackboard architecture, where the agents communicate through memory sharing. Such procedure yields reduced, although not disregardable, delays, which shall be further more deeply investigated.

Communication among agents is made through the blackboard, which has methods defined for this purpose. To implement the agent's life cycle methods were for connection, activation, transmission of MIDI events and information request about other agents (priority, status, abilities, instruments, list of registered agents, etc.) MIDI events flow occurs, thus (as a means of communication among agents), by passing them as parameters to the MIDI transmission methods of the blackboard.

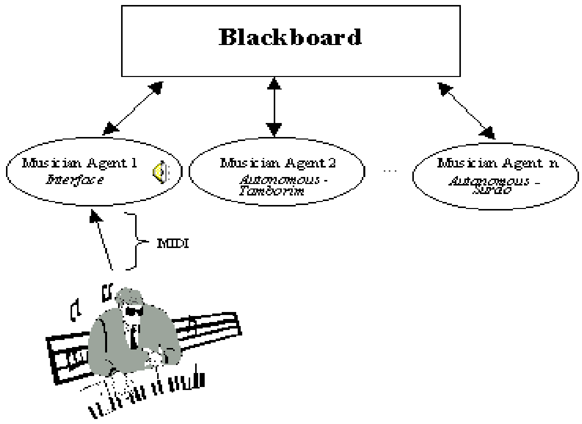

Figure 1: Main view of the Musical MAS

3. Case Study: Rhythmic Accompaniment

As a case study, we applied the described model to build a musical accompaniment system. We implemented agents capable of following with rhythms a musical execution. Each agent is responsible for the execution of its own instrument. Their main role is to remain in synchronism with the other agents, "hearing" and following the tempo of the agents of higher priority.

3.1. Rhythmic Perception

The term "rhythm" refers to the distribution of events in time such that, matched in groups, provoke a metric sensation. (Cooper & Meyer, 1960 apud Large, 1995). Early models were based on the generatives theories of metric structures. However they had failed when explaining the metric perception in musical performances.

Desain & Honing (Desain 1989) developed a connectionist model to discover the intentions of the performer. Their model discovers which between-notes interval the performer intended to express in his performance. This was made by a process called quantization.

Other connectionist models, based on synchronization to a perceived pulse, had good results in real-time performance systems. Scaborough, Miller and Jones (Scarborough 1992) described a model of metric perception known by BeatNet. It's based on a set of oscillators with different periods, synchronized by musical events. This project, however, does not treat the time variability problem.

A handling for this problem is given by the process called beat-tracking (Dannenberg 1990). Dannenberg assumes that the pulse is not static. It must be adapted during the performance, when the interpreter speeds up or down. He concludes that this task is surprisingly difficult. His program uses a static tolerance window, where each note played in this interval is considered as a note played at the "strong beat". The other notes, considered as subdivisions of the time, are ignored. When a "strong beat" note is played before the expected moment, the program automatically starts to proportionally decrease the period of the beat.

We implemented this model with some modifications. More recent and complex models (Large 1994; Toiviainen 1998) implement conceptually similar models. However, they use continuous oscillators, therefore preventing abrupt (discontinuous) changes in the phase. We alleviated this problem through the adaptation of our oscillator's input, preventing abrupt cuts in the output. So we achieved a normalization of the period in relation to the received signal, as related in (Wulfhorst 2000).

Implicit priority definition in a musical ensemble allows certain instruments to have more importance in defining beat and rhythm than others. In the same way, the external agent (human) in our case study has maximum priority, in order to lead the rest of the musical set (remembering that the definition of priorities is free, and may be different for the solution of other problems). However, with maximum priority, he has the responsibility to make his musical intentions clear to the other agents.

Another important issue beside the question "how to adapt" is the decision of "when to adapt". Two distinct moments are defined where an artificial agent ignores the playing from another agent. In a Multi-Agent system, it does not imply that the agent stops its adaptation, but just ignores events coming from agents in one of the following situations.

When an musician begins to play while other are already playing together, he has firstly to adapt his pulse to the rest of the ensemble. In this moments, the other musicians have to be concerned in maintaining their rhythmic intentions, without being affected by the new member. This is an implicit rule in musical groups and has been shown to be a very simple and efficient implementation in our system.

The second situation occurs when an agent is playing fast. This is a restriction imposed by this kind of adaptation mechanisms mentioned in (Trajano 2000). This modification was necessary to increase system's stability.

3.2. Playing in Rhythm

The system allows each agent to adjust its own beat based on the musical events read from the blackboard, depending on their priorities.

In this implementation each agent has a defined rhythmic pattern. The goal of an agent is to play his instrument in synchronism with the others. With the rhythm perception technique previously described, it checks whether it has or has not to play his instrument in that moment, in agreement to the definition of its rhythmic pattern.

4. Results

The results of our case study have been positive, since system behavior was in conformity with our expectation. A musician, admitted in the community of agents, executed musical pieces with tempo alterations, being satisfactorily followed by the rhythm-playing agents. These, in the same way, kept the synchronism among themselves. However, we discovered some situations that caused abnormal behavior:

- Abrupt tempo variations by an agent may compromise the synchronization of the others;

- Different agents mutually adapting their beats, induce a continuous retardation of it. This occurs because there is a natural delay between the sending and the reception of a musical event. Thus, the receiver agent adapts to a period a little bigger than the real one. The resulting delay propagates itself due to the continuity of this process, since all the agents are senders and receivers of events.

The case study proved the viability of a Musical Multi-Agent System's construction. The requirements for a Rhythmic Accompaniment System, implemented by a community of agents, mentioned in section 2, were satisfied by the proposed architecture.

5. Future Work

The present work aims to make possible the research, implementation, validation and integration of agents who communicate musically, supporting the continuity of our research. Thus, our next researches will include:

- Studying the interaction between agents with distinct functionalities and/or implementations (heterogeneous).

- Investigating the behavior of a musician in its most diverse aspects, since we intend that the MMAS incorporates libraries that reflect such behaviors.

- Studying dynamic agent communities: building musical agents capable to learn, hoping to improve their cooperation level.

- The current case study will be the basis for a more robust Rhythmic Perception Model. We consider this as fundamental to advance in other areas, such as Harmonic and Melodic Perception. More subjective concepts, as the analysis of an agent's musical intentions, will make possible a more intelligent interaction of the community.

- Implementing a graphical interface as an agent itself.

We expect that through a broader and more powerful library for Artificial Musical Perception, associated with learning capabilities, it will be possible do develop agents with a much more efficient collaborative behavior.

6. Conclusions

Through the implementation of the MMAS architecture, we verified that instrument execution and rhythmic accompaniment generate beat perception effects between senders and receivers, that influence directly the synchronism between such performing agents.

The Multi-Agent approach has proved to be valid for the development of musical real-time interactive systems. It's greatest advantage is to provide a model that's more similar to a real situation, as like musicians playing in an ensemble. The traditional separation between the user and the computer is transcended. The system is therefore considered as a community of agents (that it includes the user), what constitutes one of this work's contributions.

7. References

Aeken, F. & Alvares, L. O. (2000) Synthesising Applause Using Multi-Agent Systems. In: 3rd Ibero-American Workshop on DAI & MAS, Nov. 2000, Atibaia - Brazil. Proceedings... p.123-135.

Dannenberg, R. B. (1984) An On-line Algorithm for Real-time Accompaniment. Proceedings of the 1984 International Computer Music Conference. Computer Music Association.

Dannenberg, R. B. (1990) Tracking musical beats in real time. S. Arnold and G. Hair, eds. Proceedings of the 1990 International Computer Music Conference. San Francisco: International Computer Music Association. pp. 140-143.

Desain, P. & Honing, H. (1989) The Quantization of Musical Time. Computer Music Journal, Vol. 13, 1989.

Large, E. W. & Kolen, J. F. (1994) Resonance and the Perception of Musical Meter. Connection Science, 6 (1), 1994. p.177-208.

MIDI Manufacturers Association. (2000) The Official MIDI Specification. Available through HTTP at www.midi.org (last accessed in sep/2000).

Scarborough, D. L.; Miller, B. O. & Jones, J. (1989) Conectionist Models for Tonal Analysis. Computer Music Journal, Vol.13.

Toiviainen, P. (1998) An Interactive MIDI Accompanist. Computer Music Journal, 22 (4), 1998. p.63-75.

Trajano, E.; Guigue, D. & Ferneda, E. (2000) Automatic Segmentation of Musical Flows: a Rational Agents Approach. Brazilian Symposium on Computer Music, 7., 2000, Curitiba, Brazil. Electronic Proceedings of the XX Congresso Nacional da Sociedade Brasileira de Computação. Curitiba, Brazil: Champagnat.

Wulfhorst, R & Vicari, R. (2000) Percepção Musical Artificial: Uma Revisão Bibliográfica de Técnicas. Technical Report. Porto Alegre, Brazil: PPGC / UFRGS.