Musically Intelligent Agent for Composition and Interactive Performance

Keiji Hirata
NTT Communication Science Laboratories
hirata@brl.ntt.co.jp

Tatsuya Aoyagi
Tsuda College
aoyagi@tsuda.ac.jp

This paper discusses an approach to realizing intelligent musical systems that can enhance a musician's inspiration and his/her creativity. The authors believe the main obstacle to achieving such a system is a lack of colorful personality and introduce a collaborator agent. In our framework, the diversity of the collaborator's personality is realized by the diversity of inference mechanism and that of knowledge base. The planning block within a collaborator agent carries out inference and is in charge of the inference strategy. The musician's model within it represents the musical knowledge that the musician possesses. The integration of agent technology, case-based reasoning (CBR) and deductive object-oriented database (DOOD) technique can contribute to realizing a collaborator agent. Prototype system, P^3, is presented. In P^3, the planning process of a collaborator agent is implemented in CBR and its musical knowledge is managed by DOOD.

1 Introduction

Developers of composition systems and interactive performance systems face two common problems. One is building a musical system that can enhance a musician's inspiration and creativity. The other is finding the technologies that can support the activities required for such a system. Roughly, there are two design policies; one is aimed at a system with a low level interface like an assembly language, and the other an intelligent system with a high level interface. However, many of the latter system are convenient tools at most, and can hardly stimulate musician's creativity. The authors believe this problem is caused by a lack of colorful personality.

Section 2 discusses the problem in more details and proposes that agent technology [7] [1] [2] can greatly contribute to resolving the problem. Section 3 describes a jazz composition collaborator system called P^3 that we are prototyping to examine our framework. Section 4 gives concluding remarks.

2 Musical System Design

When developing a musical system, there are two levels at which the system is aimed. One is a *tool* level and the other is a *collaborator* level (Fig. 1). Here, we regard a collaborator as a high-level tool.

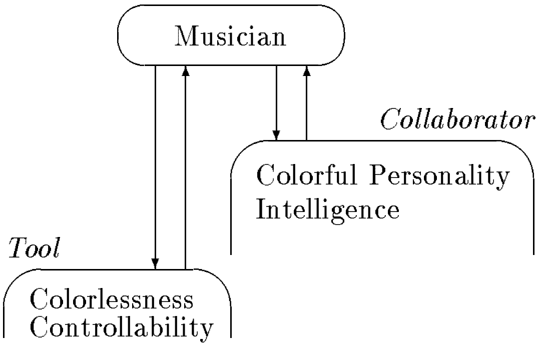


Figure 1: Tool and Collaborator

2.1 Tool

A tool is a general-purpose, user-controlled system that gives a feedback to the user; MAX and Csound are examples of such systems. A good tool has to sufficiently convey the user's intention to a target object and make its reaction tangible comprehensively. In general, this can be considered controllability or programmability. To ensure controllability, it is important to explicitly represent and manipulate the structure of a target object.

It has been claimed that a musical tool should be colorless and neutral to specific musical concepts; a colorless tool does not have ad hoc musical knowledge nor heuristics [3]. Thus, such a tool is not likely to color the works composed using it, and can realize a clear causality of cause and effect (action and reaction). However, a colorless tool is merely a passive slave for a musician, it can hardly inspire or stimulate his/her creativity.

If we reconcile controllability and colorlessness, the interface level of a tool becomes lower and the tool will have more control parameters. This may lead to complication of the target object structure, too. In general, it is difficult to control altogether the many parameters consistently.

2.2 Collaborator

A collaborator is an task-oriented intelligent system that autonomously exchanges messages with a musician. Ideally, a collaborator should be able to grasp the musician's disposition, suggest new perspectives that could trigger aesthetic revelations and fertilize the musician's own knowledge and creativity. EMI and Cypher can be viewed as collaborator systems in some sense.

Then the authors believe that a collaborator must have colorful personality and intelligence, since the diversity of the collaborator's personality can stimulate creativity on various aspects of a musician (Fig. 1). For instance, there are personalities of obedience and perversity, and there are a noisy collaborator and a quiescent one.

In order to perceive musician's ambiguous requests through the interaction, adapt to changes in the working environment and react to them, a collaborator must have the intelligence of understanding the musician's intention and that of planning.

If the collaborator judges that it has useful information, it should import the information to the musician. On the other hand, a collaborator must realize a working environment in which a musician can concentrate on the musical task itself. In this case, during composition and performance, misunderstandings of the musician's intention and the articulation, the exchange of trivial information between the musician and a collaborator, and their engaging in unessential work are undesirable. What attitude a collaborator takes depends on its personality, and the intelligence also supports such an working environment.

2.3 Agent Technology

Agent technology [7] [1] [2] contributes to realizing of a collaborator that can inspire and stimulate creativity.

Although there are many definitions of an agent, an agent is generally defined as a virtual, intelligent autonomous process that acts as a substitute for a human. This is called personification, too. In some cases, mobility is included in the definition. Usually, the user's model is embedded in an agent to realize this definition.

Further, when many agents are well-organized and communicate with each other, they can perform a complicated cooperative task.

2.4 Collaborator as Agent

The authors propose a *collaborator-as-agent* approach (Fig. 2). The collaborator agent communicates with a musician and the other agents, while it manages to control many parameters of a tool. Since we suppose that a tool provides a proper abstraction level, we build a collaborator on the tool level.

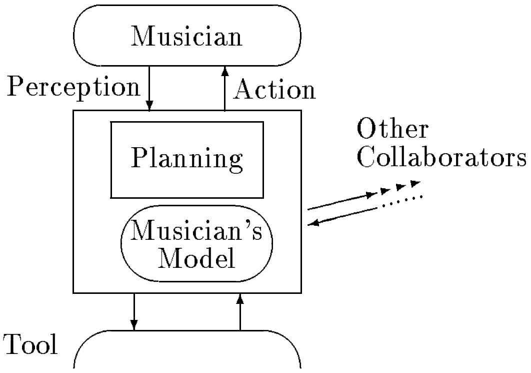


Figure 2: Collaborator agent

The collaborator agent works as follows: it shares the same goal, intention and context with a musician, perceives the musician's behavior, performs reactive planning, and acts to the outside world (Fig. 2). Here, the context means the flow of a tune and the situation in which a collaborator is working.

The source of colorful personality and intelligence is the planning block and the musician's model. The planning block carries out inference and is in charge of the inference strategy. The musician's model means the musical knowledge that the musician possesses. The planning block and the musician's model may be updated by learning and/or replacement.

Furthermore, agent technology may even open an alternative art universe. For example, since agents can work not only as collaborators but also as avatars of musicians themselves and are mobile on the internet, agents can collect music materials for the musician and cooperate with each other through the internet.

3 The P^3 System

To verify the collaborator-as-agent approach, the authors are prototyping a composition tool for jazz musicians called P^3. The basic function of this system is the generation of a jazz piano performance from a simple chord progression (e.g. D_{m7}–G_7–C_{M7}). This function can be considered reharmonization.

Key aspects of P^3 are

(a) a reharmonization mechanism that fits the musician's generic intuition,
(b) an interface that gives a musician a clear view of the model and manipulates the model as the musician intends, and
(c) availability on various platforms.

For (a), the reharmonization mechanism corresponds to the planning block and structurally represents and manipulates a chord progression and its associated voicings. For (b), the interface gives the abstraction of many tool control parameters. For (c), the various platforms are needed for the communication function.

As the very first step towards P^3, the authors have designed a musical knowledge representation and manipulation method and implemented it in the collaborator-as-agent approach. In P^3, the musical knowledge means chord, chord progression, cadence tree (described in Sec. 3.2), voicing and voice leading.

3.1 System Organization

Fig. 3 shows the system organization of P^3. Context Editor in Fig. 3 is a part of the collaborator agent and has the GUI shown in Fig. 4. In the figure, one can see "`1,1 AbM7 136 dur=144 key=Ab`⋯" at the top of the list in the central white box; these lines with indentation represent a cadence tree. Since Context Editor is implemented as a Java applet invoked through a web browser, it is available almost anywhere. Java's remote method invocation (RMI) connects Context Editor to Inference Engine. Inference Engine works as a collaborator agent; the planning process is implemented in case-based reasoning (CBR) and the musical knowledge is managed by deductive object-oriented database (DOOD) technique.

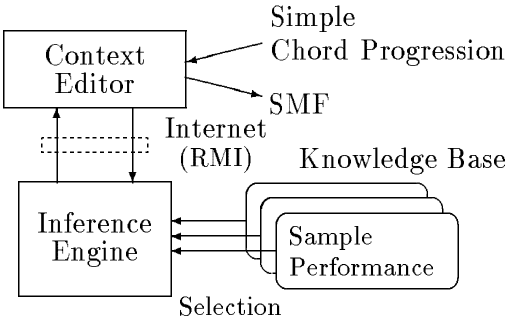
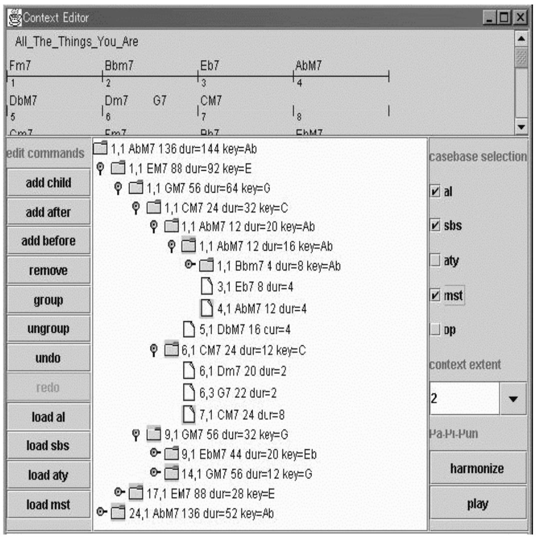


Figure 3: System organization of P^3



```
1,1 AbM7 136 dur=144 key=Ab
  1,1 EM7 88 dur=92 key=E
    1,1 GM7 56 dur=64 key=G
      1,1 CM7 24 dur=32 key=C
        1,1 AbM7 12 dur=20 key=Ab
          1,1 AbM7 12 dur=16 key=Ab
            1,1 Bbm7 4 dur=8 key=Ab
              3,1 Eb7 8 dur=4
              4,1 AbM7 12 dur=4
            5,1 DbM7 16 cur=4
        6,1 CM7 24 dur=12 key=C
            6,1 Dm7 20 dur=2
            6,3 G7 22 dur=2
            7,1 CM7 24 dur=8
      9,1 GM7 56 dur=32 key=G
        9,1 EbM7 44 dur=20 key=Eb
        14,1 GM7 56 dur=12 key=G
    17,1 EM7 88 dur=28 key=E
  24,1 AbM7 136 dur=52 key=Ab
```

☑ al
☑ sbs
☐ aty
☑ mst
☐ op

context extent: 2

Figure 4: Context Editor window

The outline of system operation is as follows. First, Inference Engine

1. loads all contexts occurring in sample performances designated by a musician.

Then, Context Editor

2. constructs a cadence tree from an input chord progression by using Context Editor, and
3. composes a query consisting of the cadence tree made at step 2 and indicates a subtree to be reharmonized.

Then, Inference Engine

4. retrieves cases that are most similar to the context in the query,

5. fetches the associated voicing of the indicated subtree in the context found at step 4, and
6. generates a standard MIDI file of a new performance.

The shallow iteration of steps 2 and 3 and the deep iteration of steps 2 through 6 render a performance planning stage, during which a musician can communicate with the system and determine an outline of the performance for real; a musician can rearrange the cadence tree displayed on Context Editor.

3.2 Cadence Tree

In order for CBR and DOOD work well, at first, a rational data structure to represent chord progression and a harmonic context is required.

We define a cadence as a sequence of chords giving the feeling of conclusion, resolution and/or termination. Usually, a cadence is made up of two to five chords. A cadence tree is a tree structure in which a cadence works as a node or a leaf; it is regarded as a representation of a harmonic context.

Fig. 5 shows a cadence tree of chord progression $\mathrm{E}^{\flat}_{M7}$–$\mathrm{B}^{\flat}_{m7}$–$\mathrm{E}^{\flat}_{7}$–$\mathrm{A}^{\flat}_{M7}$ with a certain interpretation. Within a cadence tree, the root pitches are represented relatively to the key. Each trapezoid stands for a context corresponding to a cadence; (1) is the context made of $\mathrm{E}^{\flat}_{M7}$–$\mathrm{A}^{\flat}_{M7}$ in the key of $\mathrm{E}^{\flat}$, and (2) $\mathrm{B}^{\flat}_{m7}$–$\mathrm{E}^{\flat}_{7}$ –$\mathrm{A}^{\flat}_{M7}$ in the key of $\mathrm{A}^{\flat}$. The cadence tree can be regarded as a chord progression version of the time-span reduction tree in GTTM [6]. For example, the first line in (1), $\mathbf{I}_{M7}$, is the prominent chord in the cadence I_{M7}–$\mathbf{I}_{M7}$. As such, a cadence tree represents the chord progression of an entire tune. The expression d = 5 in (2) means the difference of key $\mathrm{A}^{\flat}$ to key $\mathrm{E}^{\flat}$ is the fourth degree (five notes in the chromatic scale). Please see [5] and [4] for the technical details.

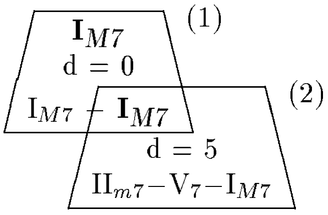


Figure 5: Example of cadence tree

Many jazz standards consists of 32 bars or so. Generally, a tune has 30~40 chords and 20~40 cadences, which means there are 20~40 nodes and leaves in one cadence tree.

A musician indicates a subtree to be reharmonized by selecting in Context Editor a box representing the root node of the subtree.

4 Concluding Remarks

The authors are now implementing the collaborator agent, and future work will be

- to improve and extend the functions of a collaborator agent,
- to explore what is the diversity of personality required for a collaborator,
- to establish an evaluation methodology for a music collaboration system like P^3 from an engineering point of view.

Our poster presentation will include a demonstration video of the current working system.

References

[1] *Comm. ACM*, Special Issue on Intelligent Agents, Vol.37, No.7 (1994).

[2] *Comm. ACM*, Special Issue on Multi Agent Systems on the Net, Vol.42, No.3 (1999).

[3] R. Rowe and B. Garton, Editor's Note, *CMJ*, Vol.17, No.2 (1993).

[4] Hirata, K.: Representation of Jazz Piano Knowledge Using a Deductive Object-Oriented Approach, *Proc. of ICMC*, pp. 244–247 (1996).

[5] K. Hirata and T. Aoyagi, Pa-Pi-Pun: Intelligent jazz harmony creation system that can be interactively used by anyone anywhere, In *IPSJ SIG Note*, 99-MUS-31, Info. Processing Soc. of Japan, 1999 (In Japanese).

[6] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*, The MIT Press, 1983.

[7] M. Wooldridge and N.R. Jennings, *Intelligence Agents: Theory and Practice*, Cambridge University Press, 1995.