BoB: an Interactive Improvisational Music Companion

Belinda Thom
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15207
bthom@cs.cmu.edu

ABSTRACT

This paper introduces a new domain for believable agents (BA) and presents novel methods for dealing with the unique challenges that arise therein. The domain is providing improvisational companionship to a specific musician/user, trading real-time solos with them in the jazz/blues setting. The ways in which this domain both conflicts with and benefits from traditional BA and interactive computer music system approaches are discussed. *Band-out-of-the-Box* (BoB), an agent built for this domain, is also presented, most novel in that unsupervised machine learning techniques are used to automatically configure BoB's aesthetic musical sense to that of its specific user/musician.

1. INTRODUCTION

Imagine an agent who is your own personal musical companion: someone who can play music with you, trade licks with you, improvise with you. This companion gets to know you and your musical personality, listening and interacting with you in such a way that, to paraphrase the drummer Keith Copeland,

> the agent answers you halfway through your phrase, bringing you to the point where you can actually sing what its going to play next, and then, instead of playing that, it'll play something against it which compliments what you're singing in your head. [5]

The goal of this research is to build an agent like this, someone that is actually *fun* to play with, to build a *believable* improvisation music companion (IMC).

This paper describes the experience of building an IMC that automatically configures its aesthetic musical sense to that of its specific user/musician. The agent's task is to trade "musically appropriate" short solos with its improvising musician in the jazz/blues setting. The activity is improvising melody – spontaneously generating a succession of notes, i.e., a *note-sequence* – like the kind one might play on a saxophone.

This agent's goal, to engage the user in an evolving musical conversation between two soloists, is called *Band-OUT-of-the-Box* (BoB) in tribute to the commercial software product *Band-in-a-Box* (BiB) [16]. Whereas BiB is non-interactive – during playback it does not listen to nor respond to what its user plays in real-time – BoB's central focus is user-specific live melodic interaction.

IMCs are a new application for believable agents (BA) and for interactive computer music systems research. Earlier systems developed in both fields did not address the special challenges that arise when creating a musical *companion.* For example, BA typically focus on interactive story telling, where success depends on how well artists can author their character's personalities within the confines of a particular BA architecture, e.g., [15]. Musical improvisation, an elusive and poorly-understood creative process, cannot be authored in this controlled way.

Similarly, most interactive computer music systems fall into one of two camps [24]. In the first, the composer is the author of the program, e.g., [18], and it is assumed that the composer will successfully author an appropriate musical aesthetic. In the second, the author attempts to build an aesthetically neutral system that can be tailored by the end-user: a composer and/or performer, e.g., *Max* [17]. While generality means that users can realize their own aesthetic ideas, to do so requires programming, or at best, the hand-tuning of many parameters. Altering the interactive task to one of musical companionship makes it paramount that the agent automatically configure itself so as to reflect its user's particular and momentary style. Simply put, composition – the notion of setting down fixed ideas in advance – makes less sense.

From a BA point of view, a main contribution of this work is to use machine learning to automate (rather than interfere with) the authoring process. Specifically, BoB watches the user improvise and uses what it sees to configure itself in a musically appropriate manner. Musically and creatively speaking, it is the mechanisms for automated configuration that are of interest. Specifically, BoB learns to perceive its user in musically appropriate ways. A key aspect of this research is to then reverse this process, enabling this

To appear in Fourth International Conference on Autonomous Agents (Agents-2000).

customized perception to control the agent's generation of melodic response.

The remainder of this paper examines the IMC domain and addresses the ways in which this application both conflicts with and benefits from traditional BA and interactive computer music system approaches. Next, design principles for building an IMC agent are formulated, and their realization in the architecture of BoB are examined. BoB's performance and limitations are also described. This work is presented in conjunction with a live real-time demonstration of BoB.

2. BELIEVABLE AGENTS

The concept of a BA, originally developed by the Oz Project, focused on building interactive virtual worlds whose characters and story-line were believable [4]. While the nature of BA are debated, a central theme is creating interactive, autonomous agents that exhibit rich, emotional personalities [15] [10] [11] [7]. IMCs both combine and extend current BA and interactive computer music systems research. This section addresses the ways in which BA methodology maps and does not map into the IMC domain.

2.1 Compatibilities

A fundamental reason why the BA approach maps into the IMC domain has to do with its audience and artist-centric viewpoint, whereby the "building a new kind of cultural experience: engaging, compelling, and hopefully beautiful and profound" [15] is embraced. Of enormous benefit here is the belief that artistic goals are at least as valuable as purely technical ones, the idea being that good art (transformed into a computational setting) will breed good science (but that the converse is not necessarily true).

In creating autonomous interactive characters that have the same rich personalities as those created by artists, many BA researchers attempt to understand the reflections of practicing character artists, and to integrate these insights into their design approach. In building BoB, the reflections of practicing improvisors have proven similarly useful [2][5][19].

BA research is also an inherently artistic pursuit. Emphasis is on *building* interactive characters. As such, a common goal is to provide character artists with the same level of artistic control that, for example, animators have. With BoB, a similar goal takes precedence. Artistic control involves coupling BoB tightly enough to the improvisor so as to make it seem like a musical conversation is taking place, while at the same time keeping BoB sufficiently decoupled so as to allow the generation of sufficiently novel new material.

Other frequently employed principles in BA research that map directly into the IMC domain include:

1. **Stress specificity.** Whereas traditional AI systems seek to capture *the* general theory of some domain, IMCs seek to capture notable aspects of a specific musician's improvisations. Bailey further emphasizes this point [2]: "There is no general widely held theory of improvisation and I would have thought it self-evident that improvisation has no existence outside of its practice." Since rule/grammar based jazz systems, e.g., [14], usually derive from *someone's* theory of jazz, this paradigmatic shift is a significant departure.
2. **Success is measured in terms of audience perception.** Whereas classical AI systems seek some form of objective measure, with IMCs one must first build the agent and then observe how *engaging* the experience is that it provides to its improvisor.
3. **Have strong affinities to behavioral AI.** In particular, prefer the broad, shallow capturing of musical capabilities. Fortunately, this makes the IMC domain a more feasible one. Whereas classical AI-based music research attempts to build a musical expert, we merely want an agent that provides a modicum of musical appropriateness in the solo trading context.
4. **Assume the audience is willing to suspend their disbelief.** This assumption also makes the task of building BA more feasible. In BoB, this assumption falls directly out of the fact that it is going to be used by musicians who *want* the computer to facilitate a musically engaging conversation with them. As a result, the user is more likely to overlook an agent's shortcomings (such as performing a solo without expressive dynamics or articulation). BiB's success is also in part due to this fact; many of its accompaniment styles are somewhat canned, yet the experience of playing with this "cheesy" band, versus playing with no band, is a powerful one.

2.2 Distinctions

That some BA philosophies do not map into the IMC domain are a direct result of the fact that activities involving interactive story telling and/or character development are fundamentally different from those involving interactive musical improvisation. "The difference between composition and improvisation is that in composition you have all the time you want to decide what to say in fifteen seconds, while in improvisation you have fifteen seconds" – jazz saxophonist Steve Lacy [2]. The important point is that since improvisation is not composed, what does it mean to *author* it?

While the specific details of an interactive story/character are also not decided until "the last fifteen seconds," their literal meaning enables more constraints to be specified upfront (e.g., a frightened woggle always quivers, except perhaps when the fear is so great that it causes paralysis). Another critical difference between BA and IMCs is that the "author," that is to say she who decides what is musically appropriate, *is* the audience/user.

Because musical conversation does not have the literal meaning and interpretable context that language does, notions of personality and character either break down or become nearly impossible to operationalize. Musicians that do speak of personality usually do so in frustratingly vague terms:

> "I would like ideally to express my, *I don't know*, personality *or whatever*, musically, to the limits of my ability. [... to ...] play *in such a way* that it would be recognizable as me, and it would express *something* to people about the way I feel

Figure 1: Trading Fours. Each staff shows a different player's spontaneous performance. The accompaniest's chords are shown on top. The first four bars of the musician's solo are shown in the middle ("|" divides bars), at which point the agent is listening. The agent then responds by playing another four bars while the musician listens (bottom).

> about things" – Jazzist Ronnie Scott [2], emphasis added.

It is also not clear that an improvisational experience makes sense outside of the physical context within which it is played, making the task of evaluating IMCs even more elusive and subjective. For example, while it may make sense to focus on a user's engagement with a personality post-facto – similar to the way that it makes sense to ask a reader to summarize a movie's plot after viewing it – it is not clear that anything but an improvisor's transitory, immediate musical experience is worthy of attention. On this point, *Yes* guitarist Steven Hicks noted that, with respect to the success or failure of his improvisations [2]: "It's either good or bad but if you listen to an improvisation over and over again it just gets worse. [...] It's something that should be heard, enjoyed or otherwise, and then completely forgotten."

Ultimately, while Bailey acknowledges that improvisors may get to know each other well enough for a common language to have come into being, he also points out that [2]:

> "...the improvisors I spoke to, in almost all cases, did not find any sort of technical description adequate. It wasn't that they weren't interested in technical matters. They just did not find them suitable for illuminating improvisation. They finally chose to describe it in so-called 'abstract' terms. And it became clear that, whatever its deficiencies, this is the best method available."

In other words, while there is inherent structure to be had in spontaneous melodic conversation, it is unlikely to expect that the burden of specifying this process can be successfully pawned off onto the improvisor.

For this reason, perhaps the best we can do to consort with the improvisor is to let an IMC attempt to automatically configure itself based on the musical appropriateness that it finds embedded within the user's improvisational example. Given the infeasible nature of explicit technical specification, BoB uses unsupervised learning techniques to do this task.

3. RELATED WORK

While I am unaware of any other interactive computer music system that focuses on both believable and companionable improvisation, much related work has been done. The systems outlined below were chosen because they address the specific issue of generating sequences of notes/chords (in contrast to other issues being researched, like expressive performance).

Walker's *Improvisation Builder* (IB) uses techniques from conversation analysis in order to build a system that trades solos and/or accompanies its user [24]. Group interaction is not specified up-front. IB is most notable in that it constantly infers its current role in the evolving conversation – to solo and/or to accompany. While mechanisms do exist for parsing and generating solos, Walker's papers do not detail these aspects. It appears that an expert-systems style approach has been used to configure the system, specifying what musical transformations are acceptable, how to derive new solos from what the user plays in real-time via cut-and-paste methods, etc. The agent-like aspects of IB are not discussed.

The goal of Bryson's interactive system is to provide real-time accompaniment to a folk melodist without the benefit of a score [8]. This work is most agent-like in that Brook's subsumption architecture was used. Most relevant here is that machine learning was used to configure the system's chord generator. In particular, a neural network was trained on a set of accompaniment examples, learning to predict what *chord* should be played given a recent window of acoustic melodic input. Lower-level issues, like how to realize a particular musically appropriate, user-specific note-sequence for a given chord/context, were not addressed.

Rowe's *Cypher* system [18], notable for its strong musical foundation, is essentially in the composer-is-author camp. Aesthetics are specified by the composer. A fundamental goal was to make this authoring as straightforward as possible by providing musically meaningful (hand-tuned) feature detectors, which a composer can then configure in whatever way they want. Cypher uses a society-of-minds based approach.

In terms of IMC, *GenJam* [6] is the most relevant system in that it uses a *specific* listener's preferences in order build an improvised melody generator that is tuned to its listener's aesthetic musical sense. In particular, a genetic-algorithm searches for a good melody generator, guided by the following fitness function: how well does the user like the melodies it generates? While *GenJam* has demonstrated some ability to encode a listener's musical preferences, this approach is ill-suited for the IMC domain because: 1) listener rating is invasive, interrupting the physical, transitory nature of solo-trading; 2) the bottleneck is the listener's ability to accurately rate many improvisations; 3) *GenJam* is neither interactive nor real-time.

4. BOB'S DOMAIN: SOLO TRADING

The task of building a complete IMC is enormous. In this section, the restrictions made in order to transform BoB into a manageable system are discussed.

The key to IMC success is to have the technology to, in a musically-appropriate manner, perceive and generate improvised melody. As such, BoB's explicit focus is *melodic content*, the succession of $< pitch, duration >$ pairs that comprise a note-sequence. Information related to expression (e.g., timber, articulation, dynamics) is ignored, justified by the following observation: who cares about expressive performance if the agent has nothing to express in the first place? Furthermore, case-based reasoning techniques for expressive audio realization of note-sequences in jazz already exist, i.e., *Saxex* [1].

4.1 Restrictions

Difficulties associated with real-time tempo tracking and expressive performance are avoided by limiting the solo-trading task as follows:

1. Only two *interactive* soloists are considered, the musician and BoB. There is also a *non-interactive* computer accompanist, whose job is to provide canned, fixed-tempo background music, guiding the soloists through time.
2. Tempo, harmonic structure and group interaction (who solos where and when) are fixed up-front. This information (called the Lead Sheet) is the road map for the conversation and is readily available to all.
3. Players take turns *trading fours* – soloing on top of four bars of accompaniment (Figure 1).
4. Solos are converted into audio via MIDI (continuous control is turned off).

4.2 Musical Appropriateness

At best, one can talk about an IMC responding to its user in such vague terms as a *musically reasonable* or *appropriate* manner, which depends upon harmonic context, the most recently performed improvisations, higher-level trends that emerge as the conversation evolves, the soloists and how familiar they are with one another, etc. So, what is musically appropriate behavior?

Musician's generally agree that successful solo-trading is accompanied by the ability to [5]:

- Extend the music contour gracefully between parts.
- Imitate or transform, to varying degrees, precise features of a previous player's ideas.
- Combine operations like those listed above with the development of ones own ideas.

Musicians also talk of becoming musically familiar with one another's melodic concepts, of having internalized a recurring vocabulary of patterns. Perhaps most important is the ability to *stimulate* conversation, as alluded to by trumpeter Tommy Turrentine [2]: "The other night, a drummer was just playing 'tit-a-tang, tit-a-tang, tit-a-tang, tit-a-tang,' all night long. Now, what in the world can *that* generate?" Simply put, the agent must be able to create novel ideas within the constraints of the evolving vocabulary – to paraphrase Jazzist John McNeil [5], "to listen to me, but not to play my stuff back to me."

4.3 Fundamental Requirements

These necessarily vague solo-trading abilities need to be transformed into an operational computational strategy. The approach taken here is based upon the hypothesis that a successful companion must rely on non-invasive data collection techniques – merely watch the user do what they do *best*: improvise – and that a musician's ill-defined abstractions – the operationalizing of which are essential for an engaging solo exchange – can be learned from the unlabelled data that has been collected in this way.

In order for this approach to succeed in the companionable fours-trading domain, we argue that the following are needed:

1. A melody's low-level surface structure must be encoded by multiple viewpoints. In particular, the pitch, intervallic and directional features described in [9] should be used,[1] for while elusive to precise technical quantification, each viewpoint is readily perceived by humans [13] and failure to attend to any one of them may defy a user's musical common sense, which in turn makes it even more difficult for them to suspend their disbelief.
2. The ability to learn more abstract "modes of playing" from the *local* trends exhibited by these low-level viewpoints. For example, the agent needs to be able to unearth the different ways in which a musician's prefers certain pitches, which in turn relates to the subconscious activity of selecting a certain scale and employing it in a particular context.
3. The ability to reverse this musical abstraction, to be able to generate new note-sequences that exhibit a particular playing mode. What this really means is that we need a probabilistic learning approach, one for which we not only learn to cluster (abstract) but also learn the parameters that best account for the data the musician has generated.
4. The ability to perceive and generate in real-time with *per-bar* granularity. With these skills, the agent has the ability to perceive the changes made during the musician's solo and generate a response that exhibits dynamically-similar behavior.
5. Responding appropriately should improve as the user and agent become more familiar with one another. Provided that a larger training set improves learning, this property is a given.

[1]The first and second relate to a solo's tonality; the second and third relate its melodic contour.

5. THE BOB ARCHITECTURE

BoB's architecture is shown in Figure 2. BoB has been implemented on the NT platform using *Visual* C^{++}. What follows is a high-level description of BoB's primary components, what they do and how they interact with one another.

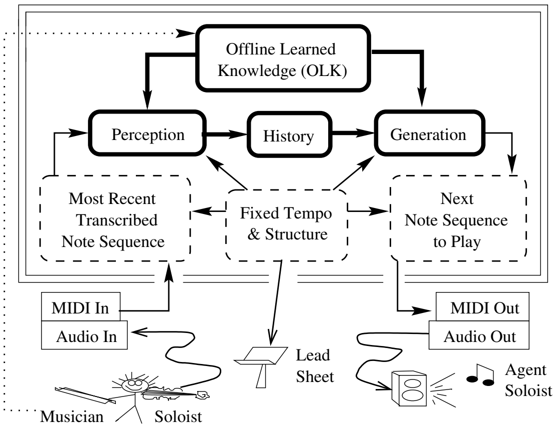

Figure 2: The BoB Architecture. BoB's components are inside the double-lined box; the environment and off-the-shelf interfaces are outside. Those components that handle real-time are dashed. Our focus is on the bold components, which are responsible for IMC-related intelligence. The dotted-line serves to remind us that OLK is musician-specific.

5.1 Representing Solos

BoB uses a novel representation for solos, namely a variable-length tree (VLT) encoding, augmented by more abstract features that attempt to summarize the structure found in the tree's leaves. An overview of the complete representation described in [21] is provided here in part to demonstrate the depth of musical knowledge that is embedded in BoB, a point that is easily obscured in the presence of specific learning details.

VLTs are perceived and generated in real-time on a per-bar basis and handle note-sequences of *different lengths.* These trees were designed to encode and constrain melody as follows. Rhythm is specified by the structure of a tree's internal nodes, each having 2 or 3 children. Pitch content – defined by both octave and pitch-class[2] – is specified in the tree's leaves. As such, melodic contour is determined by the leaves' ordering. While VLTs are suitable for encoding rhythm (e.g., it makes sense to obtain a novel, yet similar, rhythm by growing and/or collapsing various parts of a tree), the in-order melodic representation is a simple "string of notes" approach, a common computer music representation.

BoB moves beyond this typical approach in an attempt to capture deeper structure than is available on the immediate surface. This new scheme is based on the belief that the most salient aspect of melodic improvisation is not the trajectory of individual notes, but the average summary of local contexts of notes. Specifically, BoB's abstract representation involves converting a tree's in-order-walk into histograms which record the frequency with which various low-level trends occur. In particular, the multiple viewpoints outlined in Section 4.3 motivate the use of three per-bar histograms: Pitch Class (PC), which records the frequency with which various pitch-classes are used, Intervallic Motion (INT), which records the frequency of successive *absolute* intervals between neighboring pitches, and Melodic Direction (DIR), which records how many successive intervals retain the same sign.

Clearly, the generation of a musically plausible note-sequence *does* depend on the temporal ordering of pitch. So, why should one believe that a generation scheme that is based upon histograms (which, by definition, ignore temporal ordering) can produce an adequate response? Fortunately, each histogram contains some temporal information (PC contains none, INT is based on successive note pairs, and DIR is based on even longer sequences). By simultaneously considering all three of these viewpoints during generation, some level of per-bar temporal coherence is maintained.

5.2 Offline Learned Knowledge

The Offline Learned Knowledge Component (OLK) performs the task of learning *prior* to real-time solo trading. To learn, the computer first watches the musician improvise, recording what they play while "warming up" on top of a desired harmony. An unsupervised learning algorithm then uses this training data to construct an abstract probabilistic model of its user by arranging the corresponding histograms into different clusters in such a way that these clusters are most likely to have explained how the data was generated (detailed in Section 6.1).

When BoB then goes back online to trade solos, this learned model provides a crucial, customized arsenal of musical skills: abstract perception (what cluster or class is a histogram associated with?) and generation (generate a new histogram for a specific cluster).[3] During real-time operation, the History Component serves to define the current musical environment. Specifically, as shown in Figure 4, history keeps a scratch space of what cluster y_i and likelihood l_i are perceived for VLT x_i. For a 4-bar solo, this corresponds to tuples $< x_1, y_1, l_1 >$ through $< x_4, y_4, l_4 >$.

5.3 The Perception Component

As shown in Figure 3, the Perception Component determines what underlying mode (class) the musician is most likely to be using in their generation of a bar. Because a probabilistic model of the user has been built, this process involves not only an estimate of the most likely mode $y \in \{1, 2, \ldots, C\}$ that the musician is drawing their creative energies from, but also identifies how likely it is l that this estimate is correct. This latter information is of particular musical interest because it can be interpreted as a measure of how *un*surprising (expected, average) the musician's current mode use was.

[2]Pitch-class identifies which tone from the chromatic scale corresponds to a particular note.

[3]While this approach assumes that the distribution over the musician's musical behavior is identical during warmup and solo trading, this assumption can be relaxed if incremental

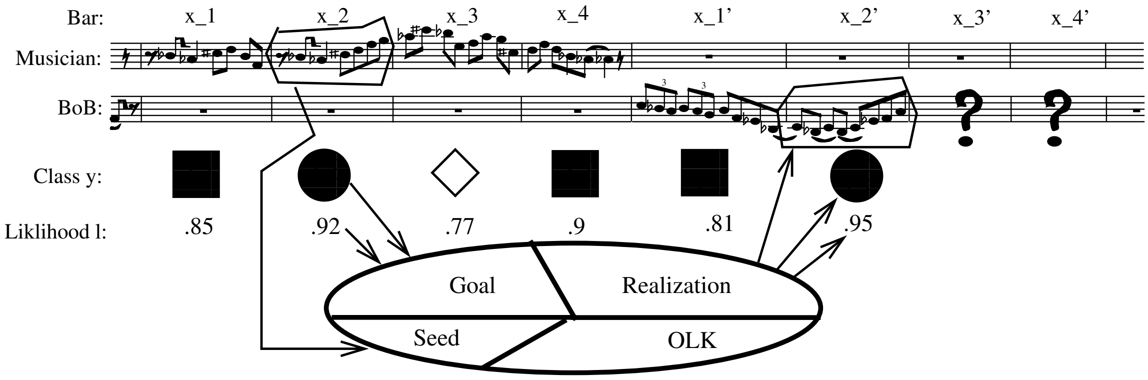

Figure 4: Generating Response in Real-time. The bold ellipse in this figure corresponds to BoB's Generative Component. The incoming information is provided by the History Component. Generation proceeds from left-to-right, producing the outgoing information. The second bar of BoB's response has just been determined.

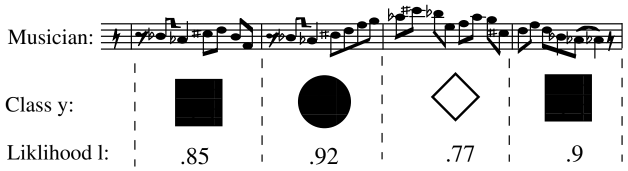

Figure 3: Perceiving a Musician's Fours

In restricting ourselves to a small number of clusters (≈ 5 per viewpoint), this mapping from note-sequences to mode is many-to-one; thus, generalization is a given. That this generalization captures the salient aspects of an improvisor's art form entirely depends upon how suitably both the representation scheme for encoding solo and the learning algorithm's bias are.

5.4 The Generation Component

The Generator is responsible for finding an appropriately constrained, yet novel and stimulating, solo response. As shown in Figure 4, Generation (the bold ellipse) is divided into four sub-components. The important point to note here is that the search for BoB's solo response, VLTs x'_1 through x'_4, is constrained by both the musician's most recent solo and BoB's customized abstract perception of that solo. The Generator uses this information to control its search through the space of possible pitch-sequences by using the goals $y'_j = y_j$ and $l'_j \approx l_j$ (note the relationship between bars j and j' in this figure. The rhythm chosen for this pitch-sequence is closely related to the seed rhythm x_j.

Generation *is* search because there is no trivial method for inventing a pitch-sequence with a specific l_j. Real-time issues require that the search is tightly coupled to this goal (per-bar, ≈ 2 seconds of computation are available). This abstract-driven search amounts to inverting the many-to-one perceptual mapping that was learned. BoB performs this inversion by introducing constrained randomness. Specifically, improvised melodic response is modelled as a spontaneous, highly constrained process – a random search through the space of possible solos that committed practice builds [12]. In BoB, committed practice is embodied in the probabilistic user model.

online techniques are incorporated.

6. LEARNING MUSICAL ABSTRACTION

We now address the most crucial aspect of IMC – how to learn an appropriate model of user-specific musical abstraction from these unlabelled histograms. We first detail the unsupervised learning algorithm that is used to infer a generative model for this data, which allows both abstract perception (which cluster does this histogram belong to?) and generation (generate a new histogram from this cluster). We then outline a method for generating a new solo (VLT) for a given abstract cluster and likelihood.

6.1 Unsupervised Learning

We developed a mixture model of multinomials[4] (vMn) whose parameters were determined by maximizing the likelihood of observing the training set's local trends in surface structure (per-bar PC, INT and DIR histograms). Note that these histograms are necessarily sparse; there are simply fewer counts than there are bins, the price paid for per-bar resolution. For complete details regarding the algorithm and its application to sparse histograms, see [23].

Briefly, vMn assumes that each histogram was *independently* generated by one of C different component distributions (abstract playing modes), where each component controls the likelihood of seeing counts in certain bins. Because the training set's "true" labelling is unknown, the optimization of likelihood is non-linear; multiple local optima exist. A variant of the expectation-maximization algorithm was developed to estimate both a likely clustering for the histograms and estimates for the model's parameters. For each viewpoint PC, INT and DIR, a separate model was built (which assumes there is no correlation between the counts in different views).

This algorithm was applied to the improvisations of bebop

[4]The multinomial is probability distribution for modelling discrete, nominal random variables.

Step	Subcomponents Involved	Action
1	Seed	VLT x_j is used to seed the search and contains a corresponding number of leaves n. For simplicity, assume that x'_j is an exact copy (or slight mutation) of x_j's rhythm.
2	OLK/Goal	Construct markov chain $\mathcal{M}$ for which component y_j's PC, INT and DIR and parameters are used to define the probabilities along edges.
3	Goal	Construct pitch-sequences by taking n-sized random walks through $\mathcal{M}$.
4	OLK/Goal	Select that pitch-sequence for which l'_j and l_j are closest.
5	Realization	Assign this pitch-sequence to the leaves of x'_j.

Table 1: A Conflict Free Generative Algorithm.

saxophonist Charlie Parker [21] in order to demonstrate the promise of our approach. Namely, very reasonable musical behavior was embedded within both the model's parameter estimates and the histogram clusterings. For example, per bar, Parker tends to select notes that correspond to relevant bebop scales, which is impressive when one considers the system was told nothing about bebop, especially given the fact that these scales were invented in part to explain his playing style [3]. Observing the abstract clustering of PC versus bar, it is also evident that the system perceived Parker as having employed different, musically appropriate scales in different parts of the twelve-bar blues progression. To cite another example, an analysis of INT and DIR reveals that Parker tends to play either straight runs of distinct pitched notes (his trademark "runs of eighth's") or note-sequences with repeated (often syncopated) note-pitches.

However most important is the component generative parameter estimates themselves. Consider for a moment the example concerning the bebop scales. Learning did not simply say "scale s was used in bar so-and-so." Rather, scale s is embedded in the more complex representation that describes *how much* individual pitch-classes were preferred by *Charlie Parker* in that scalar mode. In fact, it was the researcher who interpreted this complex representation as the (simpler) corresponding scale in order to demonstrate the power with which this approach learns musical abstraction.

6.2 Abstract-Driven Generation

A component's estimated parameters provide important clues that can be used in generation, namely what type of local surface-structure a particular abstraction is expected to display. The difficulty that remains is to develop a search procedure that quickly finds a note-sequence that realizes *each* aspect of these multiple viewpoints.

In the past, ad-hoc random hill-climbing methods were employed to encourage pitch-sequences to have certain properties [20]. Unfortunately, when stochastic change to a pitch was made, there was no explicit attempt to reconcile the conflicts that arose from the multiple, often conflicting histogram viewpoints. For example, whereas interval-based events depend on successive note *pairs,* pitch class depends on *single* notes. While one stochastic choice for pitch may improve intervallic likelihood, it might also decrease pitch-class likelihood.

We are now developing a generation algorithm for which these conflicts are resolved. In particular, as long as we are willing to assume that the parameters that control one viewpoint do not affect the others, we can directly sample pitch-sequences from a distribution that, on average, produces sequences that exhibit all of the desired tendencies. This sampling scheme merely requires a particular set of PC, DIR and INT generative parameters, and it is efficient (computation time is proportional to sequence length).

We only have space here to briefly highlight the components of this generative algorithm, shown in Table 1 and detailed in [22]. In this table, the closeness of l_j and l'_j is used to estimate the "fitness" of a given pitch-sequence. We then assume that a constant number of random walks through $\mathcal{M}$ (≈ 100) provides an acceptable range of likelihoods from which to choose l'_j.

Figure 5 is provided to give insight into how this algorithm reverses the many-to-one perceptual mapping. While there are many note-sequences that belong to a specific mode, by restricting oneself to a specific rhythm, one narrows the range of possibilities. Restricting oneself to pitch-sequences within a certain range of likelihoods narrows the range even further.

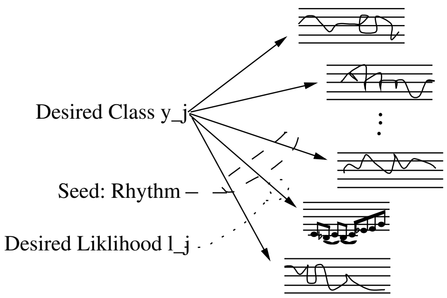

Figure 5: Abstract Generation: Search

6.2.1 *Evaluation*

When this algorithm is complete, the interesting empirical work will begin. Specifically, attempts will be made to quantify the degree to which BoB *appears* to realize the different playing modes employed by it's user. For example, this pursuit could involve: measuring the degree to which the user can identify the different modes in BoB's improvisations; observing the effect that C has on on identifiability and novelty; etc.

6.2.2 Limitations

One assumption made by this approach is that an adequately stimulating response will result when BoB *reuses* the musician's most recent abstractions to guide the search for response. Creatively speaking, this is BoB's biggest limitation. After all, is not the agent's role reduced to one of imitating, rather than complimenting, the musician?

Yes and no. On the one hand, perhaps accommodating the *musician's intent,* even if they are not a particularly good or interesting improvisor, may provide them with a more believable and engaging experience than they would have had had BoB chosen to ignore their behavior, instead responding like some virtuoso. On the other hand, aside from generating previously unheard of note-sequences for abstraction y, this algorithm does *not* initiate its own distinct aesthetic agenda. A straight forward extension to OLK may ameliorate this drawback: add an additional layer of learning. For example, learn to predict a new sequence of goals y'_1 through y'_4 from the previous sequence of goals y_1 through y_4.

7. CONCLUSION

In this paper, a new domain for agents, that of improvisational melodic companionship – in particular, interactively trading four bar solos – has been presented. Because success in this domain is measured by how fun the agent is to play with, the agenda in BA research is immediately relevant. However, as music personality does not support operational authoring, methods for inferring musical personality from a user's unlabelled musical examples were developed.

Most important, we have built a *real* agent, BoB, who shows promising ability to aesthetically configure itself to its musician in an unsupervised manner. Soon BoB will be complete – all loops between the musician, perception and generation closed – at which point the most interesting part of this experiment begins. With a working model, those aspects of the architecture which best provide an illusion of musical companionship can be investigated. Ultimately, BoB's measure of success will be how well it plays with (rather than for) its user. Or, *is the band really OUT of the box?*

8. ACKNOWLEDGMENTS

Thanks to Manuela Veloso, Roger Dannenberg, Phoebe Sengers and Michael Mateas for their valuable insight and to the Computer Science Department for their support.

9. REFERENCES

[1] J. Arcos, R. L. de Mantaras, and X. Serra. Saxex: A case-based reasoning system for generating expressive musical performances. *Journal of New Music Research*, 27(3):194–210, 1998.

[2] D. Bailey. *Improvisation, Its Nature & Practice in Music*. Da Capo Press, 1992.

[3] D. Baker. *Jazz improvisation : a comprehensive method of study for all players*. Frangipani Press, 1983.

[4] J. Bates. The nature of character in interactive worlds. Technical Report CMU-CS-92-200, School of Computer Science, Carnegie Mellon University, 1992.

[5] P. F. Berliner. *Thinking in Jazz, The Infinite Art of Improvisation*. University of Chicago Press, 1994.

[6] J. Biles. Genjam: A genetic algorithm for generating jazz solos. In *Proceedings of the 1997 ICMC*, 1994.

[7] B. Blumberg. *Old Tricks, New Dogs: Ethology & Interactive Creatures*. PhD thesis, Media Lab, MIT, 1996.

[8] J. J. Bryson. The subsumption strategy development of a music modelling system. Master's thesis, Dept of Artificial Intelligence, University of Edinburgh, 1992.

[9] E. Cambouropoulos, T. Crawford, and C. S. Iliopoulos. Pattern processing in melodic sequences: Challenges, caveats & prospects. In *Proceedings from the AISB'99 Symposium on Musical Creativity*, 1999.

[10] C. Elliot and J. Brzezinski. Autonomous agents as synthetic characters. *AI Magazine*, pages 13–30, 1998.

[11] A. Frank, A. Stern, and B. Resner. Socially intelligent virtual petz. In K. Dautenhahn, editor, *Proceedings of the 1997 AAAI Fall Symposium on Socially Intelligent Agents*. AAAI Press, 1997.

[12] P. N. Johnson-Laird. Jazz improvisation: A theory at the computational level. In P. Howell, editor, *Representing Musical Structure*, pages 291–325. Academic Press, Inc., 1991.

[13] C. M. Krumhansl. *Cognitive Foundations of Musical Pitch*. Oxford University Press, 1990.

[14] D. Levitt. A melody description system for jazz improvisation. Master's thesis, MIT, 1981.

[15] M. Mateas. An oz-centric review of interactive drama & believable agents. In M. V. M.J. Wooldridge, editor, *Lecture Notes in Artificial Intelligence*, volume 1600, pages 297–329. Springer-Verlag, 1999.

[16] PGMusic. Band-in-a-box, 1999. http://www.pgmusic.com.

[17] M. Puckette and D. Zicarelli. Max development package manual. Opcode Systems, Inc, 1991.

[18] R. Rowe. *Interactive Music Systems : Machine Listening & Composing*. MIT Press, 1993.

[19] D. Sudnow. *Ways of the Hand, the Organization of Improvised Conduct*. Harvard University Press, 1978.

[20] B. Thom. Interactive, customized generation of jazz improvisation: Believable music companions. Thesis Proposal, Submitted to School of Computer Science, CMU, 1997.

[21] B. Thom. Learning melodic models for interactive melodic improvisation. In *Proceedings of the 1999 ICMC*, 1999.

[22] B. Thom. Generating musician-specific melodic improvisational response in real-time. Submitted to the International Computer Music Conference, 2000.

[23] B. Thom. Unsupervised learning and interactive jazz/blues improvisation. Submitted to AAAI, 2000.

[24] W. Walker. A computer participant in musical improvisation. In *CHI 97 Electronic Publications*, 1997.