MUSICA

MUSical Interactive Collaborative Agent

Donya Quick, Chris Kim, and David Burke

donyaquick@gmail.com, riskun@gmail.com, davidb@galois.com

Abstract. MUSICA is a system for creating and manipulating jazz music with a novel natural language interface and reactive, generative music capabilities. It is cross-platform and can be run through a web browser on most devices.

Keywords: generative music, natural language, user interfaces, artificial intelligence, knowledge representation

1 Introduction

MUSICA (MUSical Interative Collaborative Agent) is an interactive online system for creating and playing jazz[1]. Users can create short melodies, arrange them into a larger piece, and then improvise with the aid of a computer-generated band. MUSICA challenges the notion of musical software as a passive tool and attempts to provide a more collaborative human-machine experience by featuring a natural language interface for creating and editing scores. The system also features a real-time, generative jazz band can actively listen to the users solos and responds to them in real time. An online demo[2] is available at `demo.musicaresearch.org/`. As MUSICA is an active work in progress, this paper presents only a high-level overview of each system component.

Our system supports multiple interaction modalities: chat (natural language), point and click, and MIDI controller (or a mapping of pitches to keys on a computer keyboard). Users can compose a piece then jam over it, then go back and compose some more. Our user testing during development has shown that people with varying levels of musical knowledge are able to interact effectively with the system to create music.

MUSICA is modular, with each sub-component communicating using similar representations. This has allowed us to easily wrap MUSICA's functionality with different interfaces and apply them in different domains, such as game soundtrack generation (Quick & Burrows, 2020). Our musical representations and data structures are based on the notion of *elementary composable ideas*: simple, atomic concepts that can be combined to form more complex concepts

[1] This work was funded in part by DARPA Grant W911NF-16-1-0567, part of the Communicating with Computers program.

[2] A video demo is also available at `https://youtu.be/rabG6v92JWc`

and relationships (Mandler, 2004). This style of representation has been used in related domains involving natural language, like describing and referring to items and events in visual environments (Amant et al., 2006; Pustejovsky & Krishnaswamy, 2016). Our system uses an adaptation of these concepts to the musical domain (Quick & Morrison, 2017).

2 System Overview

MUSICA focuses on of various styles of jazz using formats similar to lead sheets: a monophonic melody using traditional music notation with chord symbols appearing above the staff. Compositions in our system consist of some number of *phrases*: short sections consisting of only a few measures. MUSICA also provides several computer band styles with three musical rolls: solo/melody, harmony, and bass. Users create music with MUSICA in three steps:

1. **Phrase Composition**: constructing and editing relatively short sections of melody in the format of a jazz lead sheet. Users can preview how each phrase will sound with the computer-generated band. Our system for creating and editing scores uses a chat-style *natural language interface* in addition to some more standard point and click tools.
2. **Organizing**: creating the larger-scale structure of the piece by adding *instances* of composed phrases. We use a drag-and-drop-style interface for creating a list of phrases that can contain repeats. For each phrase instance, users can toggle which instruments are playing. Users may also choose to let the system improvise over the phrase rather than playing the melody as is.
3. **Playback/Improvisation**: non-musicians can listen to the piece performed in a variety of styles, or they can take an active role in the music with their own solos with either a MIDI keyboard or a pitch-mapping to their computer's keyboard.

Each one of these tasks is accomplished by a different module of our system, all of which attempt to "speak the same language" when possible through shared data representations and interfaces. Data representations of user-created or generated music serve as incremental snapshots that allow users to maintain different versions of the same project for future retrieval or error correction. Originally stored in a SQL database, these representations are also transformed into alternative formats for internal module-to-module communication as well as future integration with other external systems.

3 Phrase Composition

MUSICA has a chat-style natural language interface for creating and editing musical scores. Experienced musicians can interact using domain-specific terminology, such as "delete the second C," "add a G4 eighth note," or "transpose measure one up three half steps." Those with little or no musical experience have

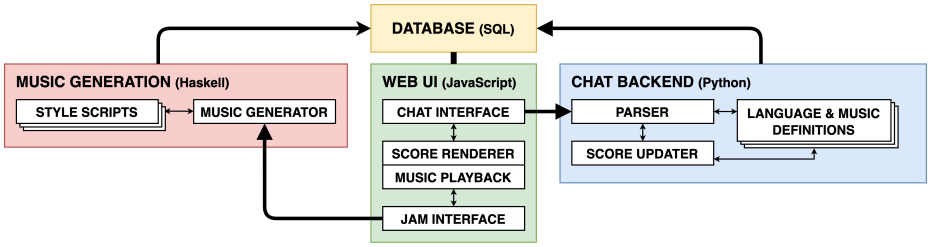

Fig. 1. MUSICA's overall system architecture.

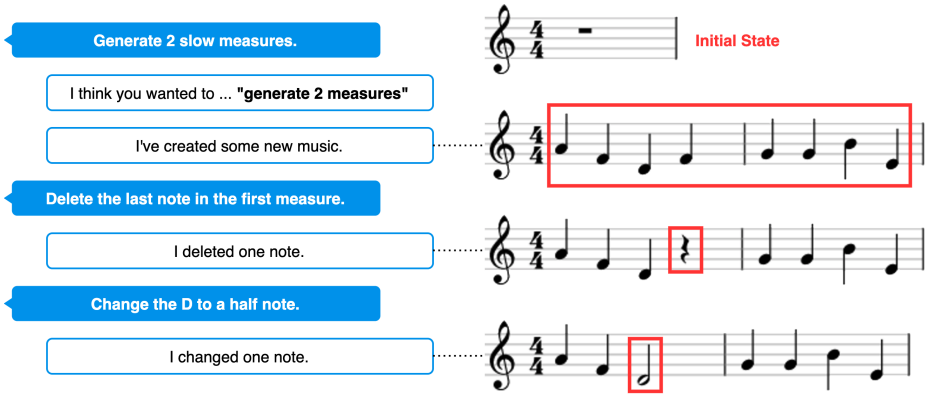

Fig. 2. Example of our "composition by conversation" chat and the resulting score after processing each command. User commands are in blue. With each change made to the score, the computer provides a response confirming what happened.

the option to simply ask the system to create things ("generate slow music") and can ask questions about the score ("what are the notes?") to learn about musical terminology.

Natural language has only been sparsely studied in musical contexts. While there has been work on music-related natural language in areas such as song descriptions and genre classification (Oramas, Espinosa-Anke, Lawlor, Serra, & Saggion, 2016; Tata & Di Eugenio, 2010), language for manipulating musical scores is relatively unexplored. In MUSICA, we address natural language in context of collaboratively making music through conversation. As a foundation for this work, we performed sentence elicitation studies to gather examples of how people talk about changes in traditional musical scores (Quick & Burrows, 2019). Based on this data, we have created our own Python-based parser to handle descriptions of a number of different musical operations. For situations where the natural language interface fails or is simply inefficient, we also allow a range of point-and-click operations on the score.

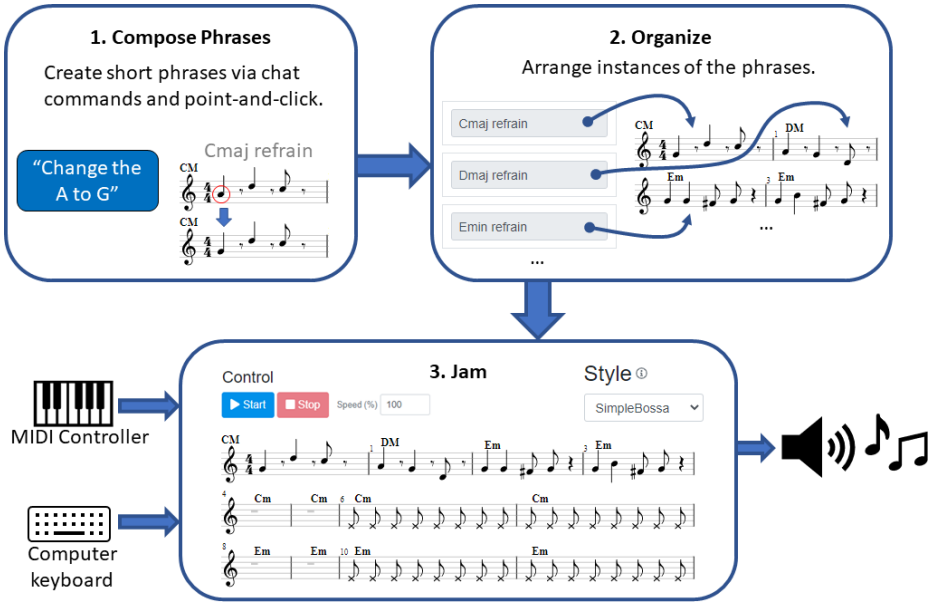

Fig. 3. User experience workflow: creating phrases, organizing phrase instances into a larger piece, and then playing or improvising with the music. In the final score, blank measures indicate areas for user soloing (although the user also is free to play outside these) and those with X-shaped note heads indicate where the computer will improvise. The performance will loop through the piece until the user decides to stop.

4 Composition Organization

Phrases are typically short, on the order of 2 to 8 measures long. Users can create higher level structure by adding instances of phrases to a piece. For each phrase instance, users can elect to either have the phrase performed as-is or treat it as an improvisational section over the same chord progression. Improvisational phrases can be marked for either user or computer soloing.

5 Playback and Improvisation

The playback and improvisational part of our system, "jam," is implemented in a combination of Haskell and JavaScript. It allows the user to either hear a completely computer-generated performance or to improvise interactively. We use just-in-time generative algorithms to fill in short sections with seamless playback. For more detail on this aspect of the system, we refer the reader to our prior work (Quick & Thomas, 2019), which is similar to some methods for generating dynamic game soundtracks (Plans & Morelli, 2012; Robertson, de Quincey, Stapleford, & Wiggins, 1998; Engles, Chan, & Tong, 2015). MUSICA supports a number of different styles for performing its lead sheets and lets the user set the

tempo of the performance. Our system does its own sound synthesis and allows low-latency responses to key presses without special equipment.

MUSICA records the notes users play during sections marked for user soloing and supplies this information to the next computer soloing phrase. This allows our generative algorithms to quickly analyze user input and provide more coherent responses that may reuse features of the user's solos. The degree of user material that appears in computer-generated solos is variable; the computer may decide to strongly echo the user or it may decide to take initiative and create a completely novel melody.

6 Conclusion and Future Work

MUSICA is a system for creating and performing jazz music using a novel natural language interface and interactive generative algorithms. Our chat interface is both an exploration of an uncommon means of musical interaction and provides an interface that will be familiar a wide range of people. The generative framework also accommodates a wide range of users, from non-musicians who just want to listen to those with more advanced musical equipment.

We have periodically done user testing throughout the development of MUSICA. While early feedback about the system expressed frustration at the unusual methods for creating music, as our chat interface has improved over time we have noticed that positive remarks are starting to feature distinctly collaborative language in comments. For example:

- "It made it fun and wanted me to interact with it more."
- "I like the idea behind it, it's a system that if improved can be a full resource for a musician."
- "It was very nice to see how the computer generated samples, and how we could adapt and compose together."

The last comment is particularly encouraging to us, since it indicates that our approach has the potential to serve less as a tool and more like a partner in music creation - perhaps a way to provide new and interesting ways to experience and learn about music for those with limited musical experience.

Our module-driven interface system facilitates easy and rapid prototyping of user studies and applications without the need to develop bespoke systems for individual use cases. In addition to experimenting with various layout permutations to accommodate various user needs, we also recognize the opportunity to build additional modules, such as music visualizations and additional virtual instruments, to further extend system capabilities and facilitate new experiences.

Because our framework is easily extensible, it will be easy continue adding support for new natural language commands, musical score features, and generative styles. We are also currently developing an app to permit easier use of our system on mobile devices while still achieving low latency responses during interactive musical performance. We hope that our efforts in that direction will eventually provide a unique way to interact with music for those who might lack the equipment necessary to use most traditional music software.

References

Amant, R. S., Morrison, C. T., Chang, Y., Mu, W., Cohen, P. R., & Beal, C. R. (2006). An image schema language. In *Proceedings of the international conference on cognitive modeling.*

Engles, S., Chan, F., & Tong, T. (2015). Automatic real-time music generation for games. In *Proceedings of the 11th AAAI conference on artificial intelligence and interactive digital entertainment* (p. 220-222).

Mandler, J. (2004). *The foundations of mind: Origins of conceptual thought.* Oxford University Press.

Oramas, S., Espinosa-Anke, L., Lawlor, A., Serra, X., & Saggion, H. (2016). Exploring customer reviews for music genre classification and evolutionary studies. In *17th international society for music information retrieval conference (ISMIR 2016)* (p. 150-156).

Plans, D., & Morelli, D. (2012). Experience-driven procedural music generation for games. In *IEEE transactions on computational intelligence and AI in games* (Vol. 4, p. 192-198).

Pustejovsky, J., & Krishnaswamy, N. (2016). Voxml: A visualization modeling language. In *Language resources evaluation conference (lrec).*

Quick, D., & Burrows, C. (2019). Evaluating natural language for musical operations. In *Proceedings of the international computer music conference.*

Quick, D., & Burrows, C. (2020). Jazzy beach critters: a demonstration of real-time music generation with application to games. In *Proceedings of the AISB workshop on computational creativity.*

Quick, D., & Morrison, C. T. (2017). Composition by conversation. In *Proceedings of the international computer music conference.*

Quick, D., & Thomas, K. (2019). A functional model of jazz improvisation. In *Proceedings of the 7th ACM SIGPLAN international workshop on functional art, music, modeling, and design* (p. 11–21).

Robertson, J., de Quincey, A., Stapleford, T., & Wiggins, G. (1998). Real-time music generation for a virtual environment. In *ECAI 98 workshop on AI/ALIFE and entertainment.*

Tata, S., & Di Eugenio, B. (2010). Generating fine-grained reviews of songs from album reviews. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 1376–1385).