

# Arrangements: Flexibly Adapting Music Data for Live Performance

Roger B. Dannenberg  
Carnegie Mellon University  
School of Computer Science  
5000 Forbes Ave, Pittsburgh, PA  
rbd@cs.cmu.edu

Andrew Russell  
Carnegie Mellon University  
5000 Forbes Ave, Pittsburgh, PA  
andrewru@andrew.cmu.edu

## ABSTRACT

Human-Computer Music Performance for popular music – where musical structure is important, but where musicians often decide on the spur of the moment exactly what the musical form will be – presents many challenges to make computer systems that are flexible and adaptable to human musicians. One particular challenge is that humans easily follow scores and chord charts, adapt these to new performance plans, and understand media locations in musical terms (beats and measures), while computer music systems often use rigid and even numerical representations that are difficult to work with. We present new formalisms and representations, and a corresponding implementation, where musical material in various media is synchronized, where musicians can quickly alter the performance order by specifying (re-)arrangements of the material, and where interfaces are supported in a natural way by music notation.

## Author Keywords

Interactive, Music, Representation, Human-Computer Music Performance, Music Notation, Arrangement.

## ACM Classification

H.5.1 [Information Interfaces and Presentation] User Interfaces -- Interaction styles (e.g., commands, menus, forms, direct manipulation)

H.5.5 [Information Interfaces and Presentation] Sound and Music Computing

## 1. INTRODUCTION

Interactive computer music systems tend to deal with musical structure in two ways: Either (1) structure is ignored as in improvisational systems where there is no overall musical plan or (2) structure is rigid as in fixed media or score following systems [2][9] or fixed media music. We are interested in computer music systems that can participate in fairly conventional music performances – what we call Human-Computer Music Performance for popular music [4][5] – where musical structure is important, but where musicians often decide on the spur of the moment exactly what the musical form will be.

Our work aims to:

- Represent plans for music performances in a formal way that is compatible with computers;
- Create computer systems that can perform music with humans where the music has a clear metrical structure and higher level form that must be followed;
- Provide for quick alterations to plans, which might change

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. *NIME'15*, May 31-June 3, 2015, Louisiana State University, Baton Rouge, LA. Copyright remains with the author(s).

immediately before the performance begins or even during the performance itself.

- Enable the coordination of modules rather than assume a monolithic, multi-media system.

Our approach coordinates multiple “players” that manage different music representations, including music notation, pre-recorded digital audio, MIDI data and other representations. We assume that every medium can be indexed by beat number, and we use mappings to express how to arrange and synchronize different media.

We use mappings to represent the meaning of “control structures” in music notation, such as repeats, endings, and *D.S. al Coda* structures. We also use mappings to represent “arrangements,” such as when the band leader says to play a particular form: “8 bar intro, play the form, a 32-bar solo, and then the last 16 bars.” Mappings are also used to express the relationship between different media. For example, a MIDI file might begin with an “empty” measure, thus the measure numbers do not correspond to measures in other media.

The use of mappings generalizes what might otherwise be a large set of ad-hoc rules and special cases. Mappings give us great flexibility to express not only conventional forms but special cases and odd problems that arise in music practice.

## 2. RELATED WORK

Although one might expect that popular music would receive a lot of attention from researchers, the idea of integrating computers into live performance of highly structured music has not been pursued by many. Rosselet and Renaud [8] describe a number of collaborative systems for popular music, but few offer much more than sequencers. Even Ableton Live [1], known for its support for interactive live performance, has little support for synchronization to live musicians, and the facility for making arrangements is not extensible to other applications or music notation. Andrew Roberts has created software to automatically control Live in order to cue audio tracks and synchronize them to the time and tempo of live rock musicians [7]. Our work can be considered a further extension to handle multiple media in a more flexible and modular fashion, and to use mappings to quickly specify arrangements.

## 3. MUSIC STRUCTURE AS MAPPINGS

Let us begin with a musical score containing repeats, first and second endings, and other control structures. Conventionally, measures are numbered from beginning to end, ignoring the control structures, which means that even a canonical performance of the score will not follow the measure numbering of the notation. Imagine a new score, generated from the original, in which the control notation such as repeats is removed, and copies of measures are inserted wherever repeats are called for. We call this a “flattened” score because we effectively “unroll” the loops and repeats [6].

More formally, we express the flattened score as a mapping  $\mathcal{F}$  from flattened measures to originally notated score measures:

$$s = \mathcal{F}(f) \quad (1)$$

where  $s$  is the location in beats according to the score,  $f$  is the location in beats in the flattened score, and  $\mathcal{F}$  is the mapping from flattened score to original score.

In order to coordinate different media such as notated music, MIDI, and audio, we need a common reference. For now, we will use the flattened score as the reference:

$$r = f \quad (2)$$

where  $r$  represents a reference beat.

Next, we consider that media such as audio and MIDI may not completely correspond to the flattened score. For example, a MIDI file might have one or two measures of silence at the beginning, or an audio file might represent a string section accompanying only the bridge in a ballad. We represent these differences with more mappings:

$$m_i = \mathcal{M}_i(r) \quad (3)$$

where  $m_i$  is the location in beats within medium  $i$ ,  $r$  is a reference beat, and  $\mathcal{M}_i$  is a mapping.

During a performance, it is convenient to count beats that may not match the reference beats. E.g. imagine the drummer sets the tempo by tapping sticks together, beat tracking software picks up the beat and starts counting, and a few beats later the band leader counts in the tune – “1 – 2 – 3 – 4 – .” At this point, the reference beat might be zero, but the internal Performance beat will be 7 or so. Then, during the performance, the singer enters a bar late and the band adjusts by repeating a bar. Now the performance beat is 11 or so greater than the reference beat. We model this as another mapping:

$$r = \mathcal{P}(p) \quad (4)$$

where  $p$  is a performance beat and  $\mathcal{P}$  is the mapping to a reference beat  $r$ .

Now we can give a formal description of *arrangements*, a reordering or restructuring of the normal score. An arrangement is a mapping from arrangement beats to reference beats. When playing an arrangement, we map from performance beat to arrangement beat, letting

$$a = \mathcal{P}(p) \quad (5)$$

Then we map from arrangement beat  $a$  to reference beat  $r$  using an arrangement mapping  $\mathcal{A}$ :

$$r = \mathcal{A}(a) \quad (6)$$

Putting these together, we can easily derive:

- the score position is  $\mathcal{F}(\mathcal{A}(\mathcal{P}(p)))$ , and
- the  $i^{\text{th}}$  media location is  $\mathcal{M}_i(\mathcal{A}(\mathcal{P}(p)))$ .

#### 4. IMPLEMENTATION

Our present system is implemented using separate processes for display, conducting, and playing MIDI and audio. In our implementation, the “Conductor” process coordinates “Player” objects, allowing any number of players to “plug in” at run time. This approach allows for multiple music displays, different musical instruments managed by different computers and/or human musicians, and different implementation languages. We use Java for music notation display and Serpent [3], a real-time scripting language, for other processes. Processes communicate via network messages.

Mappings in our system have a simple representation because in practice, they are not arbitrary real-valued functions. Instead, they map ranges of consecutive integer values to ranges of consecutive integer values. Thus, a mapping can be represented as a list of tuples consisting of the domain value (input), the range value (output), the duration or number of consecutive beats, and a section label for the cases where the range represents a labeled section of the score. In Python, an arrangement that plays the first 8 bars twice might look like `[[0, 0, 32, “A”], [32, 0, 32, “A”]]`.

The Conductor component executes a clock synchronization algorithm so that all players share a common time, as shown in Figure 1. The Conductor also transmits mappings from real time to reference beat, which players use to compute the current reference beat.



**Figure 1. The MIDI Player and the Live Score Display sharing a common time set by the Conductor. At left, time increases from bottom to top and the current position is the top of the keyboard. At right, the current position is shown by a small dot, currently below the first measure. Both representations are positioned at the first bar and remain synchronized during performances.**

#### 5. USER INTERFACE

Score image files are used to build a music notation-based interface without entering notation note-by-note. Images are annotated to indicate systems, barlines, repeat signs, section letters, etc. The user can then create an arrangement by typing a list of section names into a text area. A “Save Arrangement” button sends the arrangement to the conductor, which, in turn, sends the arrangement to all of the connected players.

Questions for the future include how to best present mappings to the user and how the user will create them. We hope to explore the use of music notation in interfaces to construct and visualize mappings. We are especially interested in the use of music notation on touch sensitive displays during live performance.

#### 6. REFERENCES

- [1] Ableton. Ableton reference manual (version 8). (Online) <http://www.ableton.com/pages/downloads/manuals> (Accessed 2011).
- [2] R. Dannenberg. An On-Line Algorithm for Real-Time Accompaniment. *Proceedings of the International Computer Music Conference*, (1984), 193-198.
- [3] R. Dannenberg, A Language for Interactive Audio Applications. *Proceedings of the 2002 International Computer Music Conference*, (San Francisco, USA, 2002).
- [4] R. Dannenberg, N. Gold, D. Liang, and G. Xia. Methods and prospects for human-computer music performance of popular music, *Computer Music Journal*, 38, 2 (2014).
- [5] R. Dannenberg, N. Gold, D. Liang, and G. Xia, Active scores: Representation and synchronization in human-computer music performance of popular music,” *Computer Music Journal*, 38, 2 (2014).
- [6] Z. Jin, and R. Dannenberg. Formal Semantics for Music Control Flow. *Proceedings of the 2013 International Computer Music Conference*, (Aug. 2013), 85-92.
- [7] A. Robertson and M. Plumley. B-Keeper: a beat-tracker for live performance. *Proceedings of the 7th international conference on new interfaces for musical expression*, ACM, New York, NY, USA, (2007), 234-237.
- [8] U. Rosselet and A. Renaud. Jam On: a new interface for web-based collective music performance. *Proceedings of the International Conference on New Interfaces for Musical Expression*, Graduate School of Culture Technology, KAIST (2013).
- [9] B. Vercoe. The Synthetic Performer in the Context of Live Performance. *Proceedings of the International Computer Music Conference*, (1984), 199-200.