A Reference Architecture and Score Representation for Popular Music Human-Computer Music Performance Systems

Nicolas E. Gold
University College London
Department of Computer Science
Gower St, London, WC1E 6BT, UK
n.gold@ucl.ac.uk

Roger B. Dannenberg
Carnegie Mellon University
School of Computer Science and School of Art
5000, Forbes Ave, Pittsburgh, PA, 15213
roger.dannenberg@cs.cmu.edu

ABSTRACT

Popular music (characterized by improvised instrumental parts, beat and measure-level organization, and steady tempo) poses challenges for human-computer music performance (HCMP). Pieces of music are typically rearrangeable on-the-fly and involve a high degree of variation from ensemble to ensemble, and even between rehearsal and performance. Computer systems aiming to participate in such ensembles must therefore cope with a dynamic high-level structure in addition to the more traditional problems of beat-tracking, score-following, and machine improvisation. There are many approaches to integrating the components required to implement dynamic human-computer music performance systems. This paper presents a reference architecture designed to allow the typical sub-components (e.g. beat-tracking, tempo prediction, improvisation) to be integrated in a consistent way, allowing them to be combined and/or compared systematically. In addition, the paper presents a dynamic score representation particularly suited to the demands of popular music performance by computer.

Keywords

Live Performance, Software Design, Popular Music

1. INTRODUCTION

Popular music (here regarded as music organized around a steady beat, performed live, and with sectional structure determined during performance after [1]) is an important category of music with specific characteristics that provide new challenges to computer participation in performance. Such music includes much (but not all) rock, folk, musical theatre, some jazz [1], country and contemporary (pop) church music, typically (but not always) in 3/4 or 4/4 time. The computer's role is thus no longer about providing expressive accompaniment according to a written score, or freely improvising in response to a stimulus, but instead needs to provide a more dynamic, improvised, and heterogeneous contribution to the music, that accounts for and reacts to the music played by all members of an ensemble. The nature of this genre has been previously characterized [1, 2] thus:

1. Music with steady tempo (not rigidly fixed-tempo but not as expressive in tempo as, for example, romantic period music).
2. Highly complex rhythmic patterns in place of expressive tempo variation.
3. Elaborate improvisation of individual parts leading to changes in rhythm and voicing between rehearsal and performance, and from performance to performance.
4. Tight synchronization including the possibility of playing slightly ahead of or behind the beat for expressive effect and rhythmic "feel."
5. Lack of notation detail as opposed to a full score.
6. Allowance of large structural changes during performance.

Popular music typically uses simple common practice notation, chord lists, or even memorized sections. The structure of a given piece is flexible, determined (and possibly changed) by the performers during planning, rehearsal, and performance. The music is commonly strongly sectionalized and thus can be rearranged, extended through section repetition, reordered, or cut at section boundaries. Ensembles often have mixed ability meaning that computer systems participating in an ensemble must be more tolerant of mistakes, planned substitutions of musical elements (e.g. chords), and ensemble members' absence from rehearsals.

Computer systems designed to participate in popular music ensembles (hereafter termed PM-HCMP systems) must address a range of problems including beat-tracking, tempo-prediction, score-following, ensemble listening, machine improvisation, score-management, and media synchronization. To facilitate the construction of such systems, allow easier integration of both new and existing components, and ultimately compare proposed solutions to problems and sub-problems in PM-HCMP, this paper presents a reference architecture for such systems and proposes an abstract score representation to support the kind of operations required during popular music performance. The primary contributions of this paper are therefore:

1. A reference architecture for human-computer music performance (HCMP) systems.
2. Score representations for managing the highly-structured and dynamic nature of arrangements in popular music.

The rest of this paper is organized as follows: Section 2 introduces requirements for PM-HCMP systems, describing two scenarios from different genres of popular music to illustrate the key points. Section 3 introduces the score representation proposed to support the architecture subsequently presented in Section 4. Section 5 concludes.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

2. REQUIREMENTS FOR PM-HCMP

PM-HCMP systems share a range of common requirements that need to be met. These are presented below, introducing terms for the various components used later in the reference architecture and representations. Such systems need:

1. A way of representing the structure of the written score (or lead sheet or other source material) in a manner appropriate to the goal of performance (for example, elaborated measures, repeats and other notational constructs); in other words, a *static score*.
2. A simple way of representing the ordering of sections of the score without needing to recreate the static score representation in full. A simple representation is required because there is typically insufficient time to fully rewrite scores in performance scenarios, the ensembles concerned may not have the expertise to rearrange music at a fine-grained level, or indeed, some of the music may exist only as memorized blocks. This is termed the *arrangement*.
3. A way in which to transform, combine, and represent the static score and arrangement together to provide lookahead and anticipation for human and computer performers: a *dynamic score*. While the internal structure of the static score sections may remain unchanged during a performance, the dynamic score can be rewritten to account for impromptu performance decisions (e.g. repeating the chorus an additional time). The dynamic score thus begins as a representation of the future unfolding of the static score and gradually becomes a history of how that score was played. The dynamic score is analogous to the execution trace of a computer program.
4. A way in which to communicate the need for changes to the dynamic score to the performers: *cues*.
5. A way in which these representations can be communicated to a range of systems involved in supporting PM-HCMP: a *reference architecture*.

2.1 Performance Scenarios

One of the fundamental problems of PM-HCMP is in reconciling the static structure of a score with the dynamic structure of its performance. Assuming that the possibly many representations of the score have been reconciled (a non-trivial task but outside the scope of this paper to solve), the ensemble has the task of planning an arrangement for performance. This would typically involve arranging the sections of the score, determining the number of repetitions of each and so forth. The arrangement will then be rehearsed, perhaps modified, and is then ready for performance. During the performance it may be that additional repetitions are included or sections are cut at the discretion of the leader.

Much contemporary Christian music falls within the notion of popular music as defined here. One example, a commonly used song in many churches, is Beth and Matt Redman's "Blessed Be Your Name" written in 2001. There are many versions of the sheet music available for this song (for example, see [4]). A typical notated form (*static score)* would contain *Intro|Verse|Link|Chorus|Bridge|Coda* with repeats to indicate the sequential structure. A typical expanded pre-service arrangement (*dynamic score*) of the piece (as written by a band member following rehearsal) would be

Intro|V1|V2|Link|Chorus|V3|V4|Link|
Chorus|Bridge|Bridge|Chorus|Chorus|Coda

In preparing for rehearsal, the music leader creates (or retrieves) a static score labeling the sections as they see fit. They then produce an arrangement with the support of software that subsequently creates a dynamic score in readiness for rehearsal. During rehearsal, cueing systems prepare the computer to play and the beat acquisition systems ensure that computer parts remain in time with the other members of the band. At performance-time, the same systems operate to keep the dynamic score in time with the ensemble. If the leader decides to repeat the bridge twice more, a cue would be issued to modify the dynamic score.

As a second example, jazz performance is often based on "standards," or well-known songs specified by a melody and chord progression. A typical performance consists of an introduction, a statement of the melody, solos, and a repetition of the melody. Each soloist plays one or more "choruses." Sometimes, the last few measures are repeated (a "tag") or a special ending is played. At a gig, the band decides to play "Airegin" and adds it to a set list. In rehearsal, it was decided to have the HCMP system play guitar chords during both the melody and the piano solo and play some pre-composed string backgrounds behind the other soloists. When it comes time for the tune, the HCMP system is ready to play, listens for the count-off, and starts to play as planned. A cue is given when new soloists enter and a separate cue enables the HCMP guitar accompaniment when the piano solo starts. During the last solo, the leader decides to extend the performance by trading fours (a common structure where a musician plays four bars, the drummer plays four bars, a second musician plays four bars, etc.). Someone cues the HCMP system to "play fours," which alters the dynamic score to continue with the chord progression but to disable any music output.

Both of these examples illustrate key aspects of the HCMP problem. At performance time, it is not enough to know the static location in the score; a performer has to know the state of the current performance. For example, is this the first or second time through a repeat? The performer also needs to have a sense of intention in order to improvise successfully (e.g. building towards a final chorus). While the computer's musical material (e.g. a MIDI sequence or audio clip) can be specified statically and attached to a nominal measure, those static elements may actually map to many locations in the full-length audio of the piece (or variations in a long midi sequence). In addition, many media must be coordinated (e.g. midi, audio, electronic music display). The benefits of HCMP in both jazz and church music are the possibility of filling in for missing musicians or to augment the instrumentation of small groups.

3. Score Representation

Since the score representation is fundamental to the operation of the proposed architecture, this is presented first. The static score representation is intended to be easy to encode from a printed score or lead-sheet whilst also being amenable to arrangement and re-arrangement during performance. Since popular music arrangement typically works at the measure-level, the representations presented here operate on measures and groups of measures.

3.1 Static Score

The static score language consists of block declarations (*Decl(a)*) and terminations (*End(a)*), numbered measures (*Mx*), repeat declarations (numbered, un-numbered, dal segno), repeat terminations, and alternative ending declarations and terminations. This language allows the abstract structure of a score to be encoded without being concerned with the lower-level specification of the music material. This is beneficial for the management of multiple media during performance and for the ease of encoding and reconciliation in preparation. The static score thus encodes the score as written at the measure level and attaches sectional labels to groups of measures.

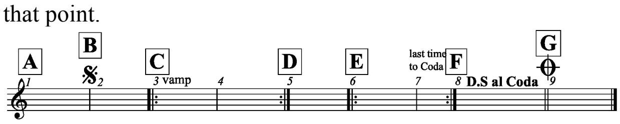
Figure 1 shows a short score fragment that will be used to illustrate the encodings proposed. The rehearsal letters indicate the designation of sections of the piece. The fragment contains a number of structural complexities including a vamp repeat (section C) to be repeated as desired by the performers, a traditional repeat and a D.S. repeat with coda. A static score encoding for this fragment is shown in Figure 1. The score encoding is relatively easy to construct quickly from reading the musical score at preparation time.

Arrangement

The arrangement uses the sectional labels declared by the static score to specify the order of the sections to be performed. This is equivalent to the musicians noting the sectional structure of the song described in Scenario 1. It allows for easy rearrangement during rehearsal and performance, simply by changing the letter ordering and regenerating the dynamic score. An example arrangement is shown in Figure 1.

3.2 Dynamic Score

The dynamic score provides a measure-level unfolding of the static score in accordance with the arrangement. Once an arrangement has been created, the measures to be played can be specified (as Mx where x is the measure number) in readiness for the render systems to schedule their data. Since it is important to be able to navigate through a piece during rehearsal (e.g. to respond to directions such as "let's go from the second time through section E"), each measure is attached to a state vector describing the sectional progress of the piece to that point.



Static Score	Arrangement	Dynamic Score
Decl(A)	A	M1[A1]
M1	B	M2[B1,A1]
End(A)	C	M3[C1_1,B1,A1]
Begin Repeat (DS)	D	M4[C1_1,B1,A1]
Decl(B)	E	M3[C1_2,B1,A1]
M2	F	M4[C1_2,B1,A1]
End(B)	B	M3[C1_3,B1,A1]
Decl(C)	C	M4[C1_3,B1,A1]
Begin Repeat (n)	D	*(CUE)*
M3	E	M5[D1,C1_3,B..]
M4	G	M6[E1,D1,C1_..]
End Repeat		M7[E1,D1,C1_..]
End(C)		M6[E2,E1,D1,..]
Decl(D)		M7[E2,E1,D1,..]
M5		M8[F1,E2,E1,..]
End(D)		M2[B2,F1,E2,..]
Decl(E)		M3[C2_1,B2,F..]
Begin Repeat (2)		M4[C2_1,B2,F..]
M6		M3[C2_2,B2,F..]
M7		M4[C2_2,B2,F..]
End Repeat		*(CUE)*
End(E)		M5[D2,C2_2,B..]
Decl Ending (DS,1)		M6[E3,D2,C2_..]
Decl(F)		M7[E3,D2,C2_..]
M8		M6[E4,D2,C2_..]
End(F)		M7[E4,D2,C2_..]
Decl Ending (DS,2)		M9[G1,E4,D2,..]
Decl(G)		
M9		
End(G)		

Figure 1: Encoding of Example Score Fragment

This captures the notion of the dynamic score being both a prescription of what is to be played and subsequently a history of what has been played. Figure 1 shows a possible dynamic score for the example fragment and arrangement shown in the figure. This is a post-performance dynamic score since pre-performance, the number of iterations of section C (the vamp section) cannot be known and it is only receipt of a cue (shown in brackets in the dynamic score) that causes the remainder of the score to be written as far as possible (until the next vamp is encountered). Unbounded repeats like this are counted during performance to support rehearsal direction (e.g. "twice through the vamp and then on"). In works without non-deterministic repeats, the entire dynamic score could be produced before performance begins.

4. Reference Architecture

Having established a score representation to support the dynamic nature of PM-HCMP, this section presents a reference architecture to capture the necessary key aspects of PM-HCMP systems. The aim of the architecture is to provide a standard organization for the components of a PM-HCMP and to give some expectations as to the type of data transmitted between them without overly constraining the design of such systems in the future. Figure 2 shows the full reference architecture that supports the key aspects of the HCMP problem.

4.1 Real-Time Components

Real-time synchronization aspects are handled by the beat and tempo tracking systems (the Beat Acquisition, Reconciliation and Prediction Modules). These should export time-stamped messages for detected pulses, meter, phase, and measures of confidence. Since there may be many of these systems, a reconciliation system is needed to filter noisy beats and decide which beat tracking source to follow on the basis of confidence and other information. This could adopt a similar approach to that outlined in [3] but accounting for the improvised nature of the music. The output of the reconciled beat data is passed to a tempo prediction system. This exports a beat-time curve to a virtual scheduler.

4.2 Abstract-Time Components

The virtual scheduler and its associated systems are concerned with the abstract time aspects of the system. The virtual scheduler retimes events scheduled on a nominal time curve by warping the curve according to the incoming tempo data from the tempo prediction system. Events are then passed to the actual scheduler for real-time scheduling. This allows the unification of all media and handles the variation of latency between the various media sources in the render system components.

The dynamic beat information (dbeat) is provided by the virtual scheduler to a structural position tracker that maintains the current score state information, mapping the dynamic score and static score and keeping the current measure count. The dbeat is a monotonically increasing beat counter and is thus inappropriate as a direct index to the static score position.

Score management is handled as described in Section 3 by the functional components in the centre of the diagram. These respond to user input (Make Static Score, Make Arrangement), and to input received from cueing systems ((Re)Make Dynamic Score).

4.2.1 Cueing Systems

Cueing systems are required to allow the computer system to react to high-level structural and synchronization changes during performance (e.g. additional repetitions of a chorus). Three types of cues are necessary:

1. *Static Score Position Cue.* This cue is necessary when synchronization with the static score is lost. Issuing it will cause the dynamic score to be re-made accordingly.
2. *Intention Cue.* This cue is needed to inform the computer of the intended direction of the current performance (e.g. exiting a vamp section or adding an additional chorus). Issuing it (e.g. using a MIDI trigger, gesture recognition or other method) will cause the future dynamic score to be remade.

3. *Voicing/Arrangement Cue.* This cue is needed to allow control over the voicing of a section (e.g. it may be desirable to prevent a particular instrumental group from playing on the first time through a repeat but allow them to play on the second). Issuing this type of cue affects only the render system to which it is issued.

4.3 Render Systems

Render systems are responsible for providing multi-media output at the appropriate time. In order to keep the detail of the specific types of media and their output separate from the abstract architecture, each render system is responsible for the management of its own data (e.g. MIDI, audio, score images). In order to link these data elements to their appropriate static score position (and thus to their appropriate scheduling as the dynamic score is played), metadata is required. A standardized format is proposed for this to relate static-score measure-numbers (and where necessary, the repeat-count, in order that the appropriate version to the dynamic score position is used) to the properties of the format concerned. This leaves render systems free to determine whether they need beat-level information or simply use the measure-level data, for example, a score display system might map a measure to image information thus: M1 → x, y, width, height, page, beat. An audio render system might represent audio at the beat level: M1 → b1:0s, b2:05s, b3…

Abstract beat-time information can thus be linked to real-time source material (to allow the correct scheduling of real-time data) while allowing the overall system to remain oblivious to the specific source formats being used. Render systems use a callback interface whereby they schedule events with the scheduling systems. These call the appropriate renderer at the scheduled time, causing synchronized real-time output of media in accordance with the dynamic score and beat tracking information. The proposed metadata representation is defined as:

metadata entry ::= measure || multi-measure || end
measure ::= (measure number, repeat-count, renderer-specific string)
multi-measure ::= (start measure number, repeat-count, count, renderer-specific string)
end ::= renderer-specific string

This generic approach allows all render systems to use the same metadata format but preserve such data as they need within the renderer-specific strings. An "end" element is required to enable renderers to terminate activities (e.g. audio playback).

5. CONCLUSIONS AND FUTURE WORK

This paper has presented a reference architecture and associated score representation for popular music human-computer music performance. The aim is to provide a standardized approach permitting easy integration of sub-systems and comparison between them. Future work will include refinements to the architecture and representations in the light of practical experience of implementing systems based on the architecture. Alternative approaches to modelling the dynamic score will be explored, for example, modelling it as a finite state machine where cue events cause state transitions. We believe that a flexible system based on this architecture will enable us to rapidly explore many variations of sensors, renderers, and interfaces as well as to integrate and share independently developed components.

6. ACKNOWLEDGMENTS

The support for this work by the EPSRC SLIM project (EP/F059442/2) and NSF Grant 0955958 is gratefully acknowledged.

7. REFERENCES

[1] Dannenberg, R.B. *New interfaces for popular music performance.* Proceedings of the 7th International Conference on New Interfaces for Musical Expression - NIME '07, ACM Press (2007), 130-135.

[2] Dannenberg, R. *Computer Coordination With Popular Music: A New Research Agenda.* Proceedings of the Eleventh Biennial Arts and Technology Symposium at Connecticut College, (2008).

[3] Grubb, L. and Dannenberg, R.B. *Automating Ensemble Performance.* Proceedings of International Computer Music Conference (ICMC-94), (1994), 63-69.

[4] Redman, B., Redman, M, *Blessed Be Your Name,* in *Here I Am to Worship*, Hal Leonard Corp, (2004), ISBN 0634079778.

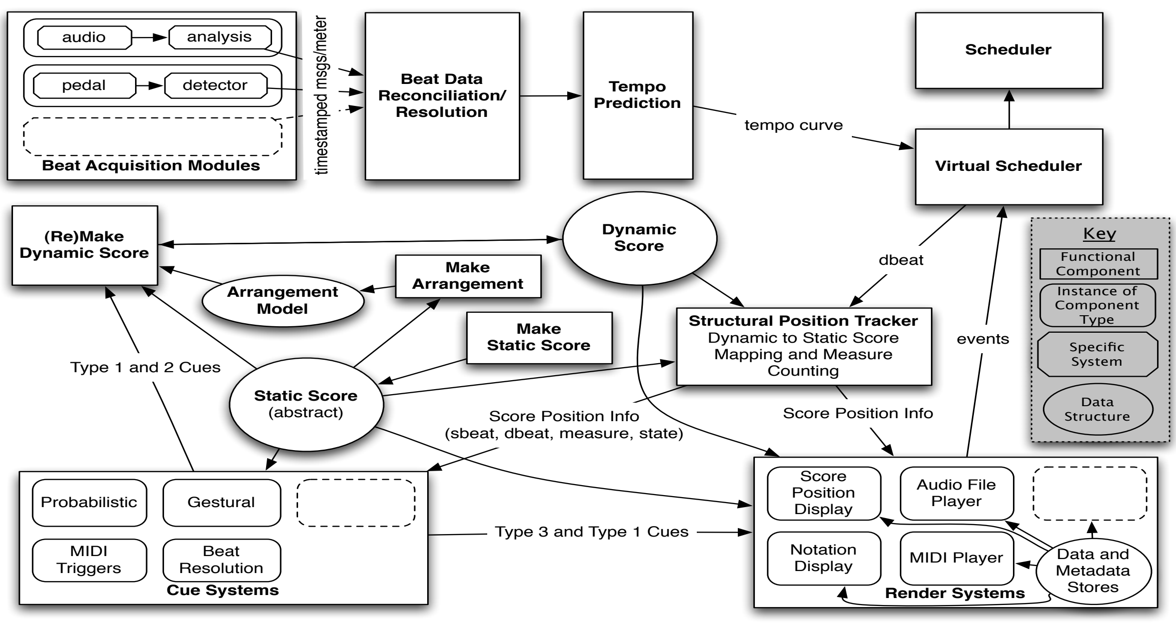


Figure 2: PM-HCMP Reference Architecture