# The Role of Time in Engineering Computer Music Systems

Eric Lee and Jan Borchers
Media Computing Group
RWTH Aachen University
52056 Aachen, Germany
{eric, borchers}@cs.rwth-aachen.de

## ABSTRACT

Discussion of time in interactive computer music systems engineering has been largely limited to data acquisition rates and latency. Since music is an inherently time-based medium, we believe that time plays a more important role in both the usability and implementation of these systems. In this paper, we present a time design space, which we use to expose some of the challenges of developing computer music systems with time-based interaction. We describe and analyze the time-related issues we encountered whilst designing and building a series of interactive music exhibits that fall into this design space. These issues often occur because of the varying and sometimes conflicting conceptual models of time in the three domains of user, application (music), and engineering. We present some of our latest work in conducting gesture interpretation and frameworks for digital audio, which attempt to analyze and address these conflicts in temporal conceptual models.

## Keywords

time design, conceptual models of time, design spaces, interactive music exhibits, engineering music systems

## 1. INTRODUCTION

Recent advances in technology have enabled increasingly complex computer music systems, many of which incorporate research from a variety of disciplines, such as computer vision for motion tracking, artificial neural networks for gesture recognition, and digital signal processing for audio and video rendering. The concept of time in engineering these complex systems has also become increasingly important, although discussion has been largely limited to issues such as finding techniques for acquiring input data at sufficiently high sampling rates to avoid aliasing [11], or maintaining low-latency, real time performance [2]. From a design perspective, discussion of time in computer music systems is largely limited to a musical perspective [12, 3].

Our work aims at exploring non-standard ways of interacting with multimedia. Music, as an inherently time-based medium, is a natural choice for us to explore these various interaction techniques, and some of our work has resulted in a series of interactive music exhibits. We have found, through this work, that the mapping of temporal parameters from the user domain to the application (in this case, music) domain, or from the music domain to the engineering domain, is often overlooked, and thus, becomes a barrier to designing good interactive computer music systems.

In this paper, we present a time design space for interactive music systems with time-based interaction. We will then use this conceptual framework to describe and analyze the issues with time we have encountered when designing computer music systems. We will provide specific examples from our past and current work; in particular, we will discuss how our discoveries in relating conducting gestures (user domain) to music beat (music domain) can affect the usability of interactive conducting systems, and how the varying conceptual models of time between beats (music domain) and audio samples (engineering domain) affect the adoption of digitally sampled audio in computer music systems.



## 2. RELATED WORK

Domains of time have been proposed before for film [5], where the distinction between the "story" time and "real" time is important for capturing the viewer's attention and providing an entertaining experience. In his work on *Media Streams*, Davis [7] also makes this distinction, and Media Streams aims to provide a metadata framework that enables easier and better media reuse for cinema. The design space we propose in this paper, on the other hand, is provided to better understand the challenges of designing and engineering time-based interactions for music and multimedia systems.

A similar design space was jointly proposed during the Time Design Workshop at the CHI Conference for Human Factors in Computing Systems in 2004 [1]. This design space, which the authors had an active role in creating, consisted of four domains: interaction, user (the person doing the interaction), object (the target of the interaction), and context (the environment in which the interaction is taking place). Taking the specific example of an interactive conducting system such as *Personal Orchestra* [4], or *You're the Conductor* [9], the temporal aspects of each domain are:

- **Interaction:** The system interprets the user's conducting gestures.
- **User:** The user imposes his own tempo through gestures.
- **Object:** The music has a base tempo and temporal structure as described in the score.
- **Context:** The user's friends want to leave, thus encouraging the user to hurry up and get to the end of the interaction sequence.

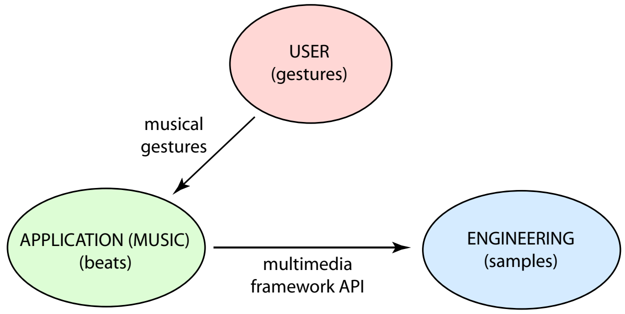


**Figure 1: A time design space for computer music systems, consisting of three domains: user, application (music), and engineering. Interactive computer music system often connect these domains: musical gestures (e.g., conducting, instrument playing) connect the user to the music, and a programming interface connects the music to the underlying technology (engineering).**

Our time design space was created as a tool for describing and analyzing design issues for multimedia systems with time-based interaction. Thus, the domains and links between them are slightly different; these will be described in more detail in the next section.

## 3. TIME DESIGN SPACE

Figure 1 shows our time design space[1]. We have divided this space into three domains:

- **User:** The user domain represents time as perceived by the systems' users.
- **Application (music):** The application domain represents the temporal properties of the medium. In this paper, we limit this medium to music, although it can include other forms of multimedia, such as speech or cinema.
- **Engineering:** Finally, the engineering domain represents the time model of the underlying computer technology used to implement the medium; MIDI, in the form of note events, and digital audio in the form of samples are two examples for music.

Computer music systems often span these three domains, and the links between these domains are a particular type of interaction. For example, when a user conducts an electronic orchestra or plays an electronic violin, they impose their personal time on the music, which usually has an inherent temporal structure written in the score. In the case of an interactive conducting system, the interpretation of conducting gestures which map user time to music time is not necessarily straightforward, as we will discuss in the next section.

Another type of interaction between domains occurs from the music to engineering domains; this interaction takes the form of a programming interface that a person uses to develop a computer music application. This interface can take the form of a visual programming language such as Max/MSP (http://cycling74.com), or in the form of an *application program interface* (API) such as QuickTime or Core Audio (http://apple.com). In our work with using digital audio in interactive conducting systems, we have found that the differing time models between the music and and engineering domains creates difficulties for the application developer. These difficulties and our proposed solution and our proposed solution will be discussed in subsequent section.

[1] Our "time design" space should not be confused with other, more well-known, "design spaces" in human-computer interaction (HCI), such as the one presented by Card *et al*. for input devices [6].

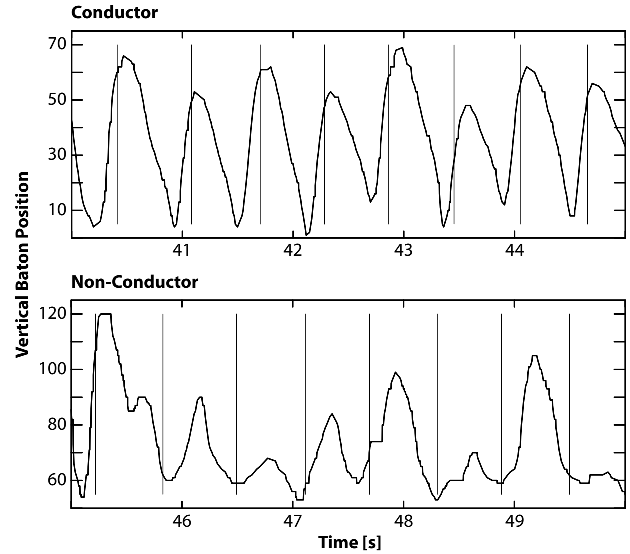


**Figure 2: Sample $y$ vs. $t$ plot of a conductor and a non-conductor. Conductors conduct more consistently than non-conductors. The vertical lines mark the actual beats of the music.**

## 4. TIME IN GESTURE INTERPRETATION

One of the challenges of designing an interactive conducting system is how to properly map the user's conducting gestures with a baton to the music beat. Usa and Mochida [13], for example, observed that conductors often do not feel "satisfied" with a conducting system that follows their beats *too* closely. We often observed a related usability issue where non-conductors would enter a "spiral of death" while conducting a system that follows their beat: they follow the music beat rather than lead it, resulting in a slowdown of the music tempo, which results in a further slowdown of their gestures, and so on.

To study the above issues in more detail, we recently performed a series of user studies comparing the temporal characteristics of conducting gestures amongst conductors and non-conductors [10]. The aim of this work was twofold: to determine a method of systematically distinguishing conductors from non-conductors, and to better understand the various mental models of conducting. Our results confirmed that conductors, unsurprisingly, place their beats ahead of the music beat, and with little variance. Non-conductors, however, also place their beats slightly ahead of the beat on average, but vary them significantly more (see Figure 2). By analyzing beat placement, we were able to uncover a variety of differing conceptual models of conducting. Some users, for example, unconsciously conduct to the *rhythm* (musical pattern formed by the dominant melody/percussion) rather than to the *beat* (consistently spaced intervals to count time, see Figure 3). Others synchronize their beats to the upwards turning point of a simple up-down gesture, rather than the downwards turning point.

These results have interesting implications for the design of interactive conducting systems. Let us reconsider the mapping from

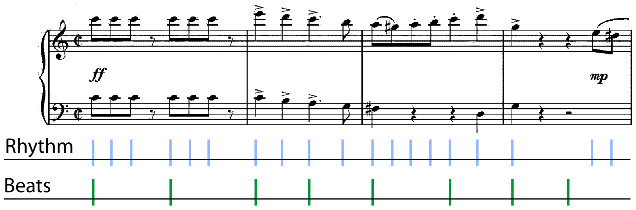


**Figure 3: The first four bars of *Radetzky March* by Johann Strauss. Also shown is the rhythm pattern and the beat pattern.**

user gestures to music beat. A system that attempts to match the music beat with a conductor's beat with minimal latency would actually be undesirable, as Usa and Mochida observed, since conductors *expect* to lead the orchestra by some fixed time interval (e.g., 150 ms, or $\frac{1}{4}$ of a beat, for the particular rendition of Radetzky March that we used in our studies). However, conductors, with their precise timing, would expect the orchestra to respond quickly to *changes* in their beat placement pattern. Non-conductors, on the other hand, conduct more unpredictably, and rather than consistently leading the beat, will sometimes follow it. A system like *Personal Orchestra*, which expects the user to consistently lead the orchestra, would then behave unnaturally, resulting in phenomena such as the "spiral of death" described above. Moreover, since, as we found, some users unconsciously conduct to the music rhythm rather than the beat, a tight coupling between their gestures and the music beat would result in erratic and unexpected changes to the music tempo. In this case, a more sluggish, "benevolently forgiving" response to changes in their movements is actually desirable.

# 5. TIME IN COMPUTER MUSIC SYSTEMS WITH DIGITAL AUDIO

Synthesized music, such as MIDI, continues to be the medium of choice for many of today's computer music systems. While synthesized music has a number of advantages over digitally sampled audio, namely in the explicit control it allows over beats, notes and voicing, digital audio has the advantage of increased realism. Today's synthesizing technology is still unable to reproduce, for example, the unique character of the Vienna Philharmonic playing in their Golden Hall of Vienna's Musikverein. For this reason, we chose to use digital audio in our recent conducting systems, and we continue to work on some of the issues that encumber a more widespread adoption of digital audio in computer music systems. We will describe some of this work in the following sections.

## 5.1 Malleability of time

Explicit control over time is a vital component of musical expression, and as a musician, this control over time is often taken for granted. In our previous work on conducting systems, where users have control over the music speed, volume and instrument emphasis, users most easily identified the interaction with music tempo: in an particular evaluation session where we silently observed users interacting with Personal Orchestra and then interviewed them, 93% of the users realized that they could control tempo by moving the baton faster or slower, 77% realized that they could control volume by making larger or smaller gestures with the baton, and 37% realized that they could control the instrument emphasis by conducting to different sections of the orchestra shown on the large display [4].

Unfortunately, extending this malleability over time to the computer music domain is not always straightforward. While this control translates well to event-based schema such as MIDI, manipulating the temporal properties of digital audio is much more challenging, since the naïve approach of resampling audio has the side-effect of changing the pitch as well. While some artists, such as DJs, will use this characteristic to their advantage, a majority of musicians will consider this an undesirable side effect.

Our current research includes developing better techniques for time-stretching algorithms based on the phase vocoder algorithm [8]; these techniques have been incorporated into systems such as *You're the Conductor*, which performs time-stretching of audio in real time without pitch-shifting artifacts.

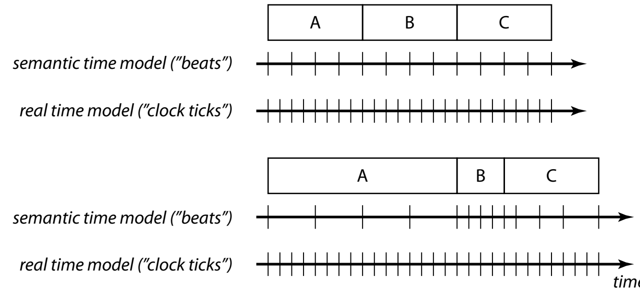


**Figure 4: Difference between the semantic time model and the real time model. The top figure shows an audio clip divided into three segments, and the two models of time: beats and clock ticks. The bottom figure shows segment A time-expanded and segment B time-compressed, resulting in a non-linear relationship between the two models.**

## 5.2 Conceptual models of time in music and engineering

Another problem that we have encountered whilst engineering our interactive conducting systems is a conflict between models of time in music and in engineering. In music, the most natural conceptual model of time is based on its beats and notes, which we call the "semantic time model". Engineers working with digitally sampled audio, however, consider time in terms of regularly spaced clock ticks in real time, the "real time model"; this clock drives, for example, audio sampled at 44.1 kHz. For applications that do not manipulate time, these two models are compatible, since movie time can be described as a linear function of real time. Consider, however, a more complex example of an audio clip divided into three segments, where the first segment is time expanded, the second is time compressed, and the third remains unchanged (see Figure 4). The relationship between semantic time and real time, which is now non-linear, can no longer be described as easily.

While this distinction between semantic and real time has been discussed before [5], modern multimedia frameworks such as QuickTime, DirectShow/DirectSound (http://microsoft.com) or Max/MSP continue to use the real time model for digitally sampled audio and video. Users of the framework are left with the challenges of bridging the gap between these two models on their own, which makes even a simple task such as synchronized playback of time-stretched audio and video unnecessarily difficult; the fact that very few such systems exist today is evidence of this.

To help address these challenges, we are currently working on a new multimedia framework that allows application developers to work directly in semantic time. One of the benefits of our "semantic time framework" is that it allows the user to work in time units of their choosing; for music systems, this can be beats. Since the tempo of a musical performance varies throughout the piece, the beats may not be evenly spaced; nonetheless, beats is often a more convenient unit of time to work with, and we have successfully used this framework to simplify and modularize the design of our interactive conducting systems.

A beat-based system also enables more interesting time-based manipulation of audio data. One can imagine time-stretching to be a semantically simple operation on the temporal axis, similar to scaling the brightness of an image. With the explicit knowledge of beat information provided by the framework, more complex effects such as equalizing the spacing between beats, or adding a "swing" effect to the music would be easier to implement.

Finally, one could imagine other time-based media where semantic time could be applied. For example, for speech, the semantic time units could be in the words. This would facilitate non-linear time-stretching of speech where one would want to change the speed of the speech but not the spaces between them. Alternatively, synchronization of speech with subtitles often found in karaoke machines would be made simpler.

## 6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a time design space for examining and understanding the issues with designing computer music systems with time-based interaction. We then described specific challenges with time design that we encountered in our previous work with interactive conducting exhibits in the context of this design space. We discussed some of the problems when mapping time from the user domain to the music domain through conducting gestures, and when mapping time from the music domain to the engineering domain through digital audio.

Our goal with this design space is to create a conceptual tool to better communicate the time design issues of developing computer music systems with time-based interaction. We believe that this design space is not limited to computer music systems – for example, the application domain could be speech instead of music. The temporal model could then be based on words or syllables rather than the model of notes and beats that we have used in this paper.

As we continue our work with designing interactive multimedia systems, it is our goal to further refine this design space. For example, does a link exist between the user and engineering domains? Are there other time design challenges which exist outside the scope of this current space, and if so, is it possible to extend the space to include these challenges? As we continue to develop this design space, we hope that it will serve as a useful tool for both computer music enthusiasts as well as engineers developing innovative interfaces for computer music.

## 7. ACKNOWLEDGEMENTS

The authors would like to thank the participants of the CHI 2004 Time Design Workshop for their insightful discussion of time design issues.

## 8. REFERENCES

[1] ACM. *Time Design Workshop*, CHI 2004 Conference on Human Factors in Computing Systems, Vienna, April 2004. http://timedsn.net/.

[2] T. Beamish, K. Maclean, and S. Fels. Manipulating music: multimodal interaction for DJs. In *Proceedings of the CHI 2004 Conference on Human Factors in Computing Systems*, pages 327–334, Vienna, April 2004. ACM.

[3] J. Borchers. WorldBeat: designing a baton-based interface for an interactive music exhibit. In *Proceedings of the CHI 1997 Conference on Human Factors in Computing Systems*, pages 131–138, Atlanta, 1997. ACM.

[4] J. Borchers, E. Lee, W. Samminger, and M. Mühlhäuser. Personal orchestra: A real-time audio/video system for interactive conducting. *ACM Multimedia Systems Journal Special Issue on Multimedia Software Engineering*, 9(5):458–465, March 2004. Errata published in next issue.

[5] D. Bordwell and K. Thompson. *Film Art: An Introduction*. McGraw-Hill, New York, 2003.

[6] S. K. Card, J. D. Mackinlay, and G. G. Robertson. A morphological analysis of the design space of input devices. *ACM Transactions on Information Systems*, 9(2):99–122, April 1991.

[7] M. Davis. Media streams: An iconic visual language for video annotation. *Telektronikk*, 4(93):59–71, 1993.

[8] J. Laroche and M. Dolson. Improved phase vocoder time-scale modification of audio. *IEEE Transactions on Speech and Audio Processing*, 7(3):323–332, May 1999.

[9] E. Lee, T. M. Nakra, and J. Borchers. You're the conductor: A realistic interactive conducting system for children. In *Proceedings of the NIME 2004 Conference on New Interfaces for Musical Expression*, pages 68–73, Hamamatsu, Japan, June 2004.

[10] E. Lee, M. Wolf, and J. Borchers. Improving orchestral conducting systems in public spaces: Examining the temporal characteristics and conceptual models of conducting gestures. In *Proceedings of the CHI 2005 Conference on Human Factors in Computing Systems*, Portland, April 2005. Accepted for publication.

[11] T. Marrin Nakra. *Inside the Conductor's Jacket: Analysis, interpretation and musical synthesis of expressive gesture*. PhD thesis, Massachusetts Institute of Technology, 2000.

[12] M. V. Mathews and F. R. Moore. GROOVE – a program to compose, store, and edit functions of time. *Communications of the ACM*, 13(12):715–721, 1970.

[13] S. Usa and Y. Mochida. A multi-modal conducting simulator. In *Proceedings of the ICMC 1998 International Computer Music Conference*, pages 25–32, Ann Arbor, 1998. ICMA.