Personal Orchestra: Conducting Audio/Video Music Recordings

Jan O. Borchers
Computer Science Dept.
Stanford University
Stanford, CA 94305-9020
borchers@stanford.edu

Wolfgang Samminger
Computer Science Dept.
University of Linz
4040 Linz, Austria
Wolfgang.Samminger@liwest.at

Max Mühlhäuser
Telecooperation Group
Darmstadt University
64283 Darmstadt, Germany
max@informatik.tu-darmstadt.de

ABSTRACT

Personal Orchestra lets anybody conduct an electronic orchestra at a new level of realism: users interact not with a synthetic, but an original audio and video recording of a real orchestra—the Vienna Philharmonic. Nevertheless, they can interactively control not only volume and instrumentation, but also tempo of the orchestra, using natural conducting gestures. A gesture-tracking and -predicting algorithm interprets user input, and a high-fidelity playback algorithm renders audio and video at variable speed without time-stretching artifacts such as pitch changes. The system was designed following a set of user interface design patterns for interactive exhibits. It is being used as the "Virtual Conductor" exhibit in the HOUSE OF MUSIC VIENNA by hundreds of visitors every day.

Keywords

Conducting, orchestra, exhibit, gesture recognition, time-stretching, music, audio, design patterns

INTRODUCTION

Today, much research focuses on devising better ways to deliver electronic music to listeners. Less effort, however, has gone into creating new ways to empower the recipient to interactively influence and control the recorded musical performance. At the same time, experience with commercial systems shows that offering the user creative control over a multimedia playback process generally leads to a much more engaging experience that offers the listener a more intense long-term level of engagement and satisfaction.

Conducting is a well-established professional example of controlling music, and it is often mimicked by amateurs conducting alongside a recorded piece. The Personal Orchestra project aims to provide a new level of realism to this experience, by offering the user an immersive audio and video rendition of the orchestra playing the piece, and by giving the user actual real-time control over the musical performance, using conducting gestures.

Project Environment

The HOUSE OF MUSIC VIENNA is an exhibition and venue center dedicated to presenting the rich musical past, present, and future of Austria's capital. It opened its doors to the public in June 2000. It offers visitors four floors of interactive and traditional exhibits on musical topics, ranging from a basic understanding of human music perception, to a historical tour of Vienna's musical geniuses, to installations of futuristic musical instruments.

When the center was still in its planning stages, we decided to create an interactive exhibit for this environment that would provide visitors with the experience of conducting a philharmonic orchestra.

Constraints From the Orchestra

Thanks to historical ties between the HOUSE OF MUSIC VIENNA and the Vienna Philharmonic Orchestra, we were able to work with this famous orchestra. This implied high demands on video and especially audio fidelity of the final system, since such an orchestra will accept neither any "synthetic" sound generation such as MIDI, nor any audio quality noticeably below CD standards.

Interactive Exhibit Constraints

Interactive exhibits like *Personal Orchestra* pose several design challenges that are less prominent in other types of interactive systems. In particular, they require a near-zero learning curve, because of their special user profile:

One-time users. Users typically encounter the system for the first and (unless they return to the exhibition again and again) also the last time.

Short-time users. Visitors use the system over a rather short period of only a few minutes (see our evaluation for data supporting this claim).

Fuzzy user profile. Apart from the above characteristics, few assumptions about the "typical user" can be made, since there is hardly any demographic limit to the visitors of a museum or exhibition center.

Our problem, then, can be summarized as:

> Create an interactive system to let users without prior knowledge about conducting or technology use a natural interface to conduct a recording of the Vienna Philharmonic Orchestra playing a classical piece, and

Proceedings of the Second International Conference on WEB Delivering of Music (WEDELMUSIC'02)
0-7695-1862-1/02 $17.00 © 2002 IEEE

create an authentic rendition of the orchestra, based on real recorded audio and video, that follows the user's conducting in real time and as realistically as possible.

RELATED WORK

There is a large body of research in systems that follow human conducting. We will only present those efforts most relevant for comparison here.

Max Mathews' *Radio Baton* [11] was among the first systems to provide a conducting experience. It uses the movement of one or more batons emitting radio frequency signals above a metal plate to determine conducting gestures. A MIDI file is played back in sync with these movements. Conducting is restricted to the space above the metal plate.

In the *Virtual Orchestra*, a commercial system by Fred Bianchi and David Smith, two technicians follow the movements of a conductor, adjusting playback parameters of a computer cluster accordingly in real time. The system has been used successfully in many commercial productions, but produces synthesized sound only.

Similarly, the *Digital Orchestra* [13] by Jeff Lazarus and Steve Gabriel offers real-time adaptable music playback, but is also based on synthesized sounds, and does not include a video of the orchestra.

The *WorldBeat* interactive music exhibit [2] contains a *Virtual Baton* feature to let users conduct a classical piece using an infrared baton. It reacts very directly and realistically, using gesture frequency, phase, and size to adjust tempo and dynamics. It detects the upbeat at the start of a piece, and detects syncopic pauses in mid-play. Conducting again controls playback of a MIDI score. The conducting feature is based on earlier work by Guy Garnett et al. [7].

Satoshi Usa's *MultiModal Conducting Simulator* [16] uses Hidden Markov Models and fuzzy logic to track gestures with a high recognition rate of 98.95–99.74%. It plays back a MIDI score, with matching tempo, dynamics, staccato/legato style, and an adjustable coupling of the orchestra to the conducting.

Marrin's *Digital Baton* measures additional parameters beside baton position, such as pressure on parts of its handle, to allow for richer expression [9]. Her *Conductor's Jacket* [10] uses sixteen additional sensors to track muscle tension and respiration, translating gestures based on these inputs to musical expressions. It uses a MIDI-based synthesizer to create the resulting musical performance.

T. Ilmonen's *Virtual Orchestra*, demonstrated at CHI 2000, is one of the few systems that also feature graphical output; however, it renders the orchestra synthetically as 3-D characters. Audio output is again MIDI-based [6].

In SONY's MusicBox project [15], an interactive exhibit in the SONY complex in Berlin allowed visitors to "conduct" the Berlin Philharmonic. It featured a high-quality, large video display of the orchestra, and audio was produced from original recordings of the orchestra. However, the only parameters to control were volume and emphasis of instrument sections. Controlloing tempo—technologically challenging, but one of the most basic aspects of conducting—was not possible. The exhibition was closed after several months.

Thus, all these systems share one or more of the following characteristics, rendering them unsatisfying for us:

- They are mostly designed to interpret professional conducting styles, which does not match the skills of our target user group of museum visitors.
- While many of them focus on optimizing their gesture recognition, they do not pay the same attention to their output quality: using synthesized sound generation such as MIDI playback instead of processing the actual audio recording of a piece makes it virtually impossible to create a system with the unique sound of a specific, renowned orchestra such as the Vienna Philharmonic playing in their Golden Hall.
- These systems mostly do not provide a natural video rendition of the orchestra playing—a critical feature of the experience we wanted to provide.

DESIGN

Personal Orchestra required solving a set of difficult user interface technology challenges. They all developed, however, out of our fundamental goal of enabling the user *experience* outlined in the problem statement above.

Design patterns

Our user interface design was based on a set of *human-computer interaction design patterns* for interactive exhibits [3]. These HCI design patterns capture principles and guidelines of interaction design for this class of systems.

Each pattern is a textual and graphical description of a successful solution to a recurring usability problem in interactive exhibits, and contains the same components: Its *name* is used to refer to the pattern easily and create a vocabulary for the design team. Its *ranking* shows how valid and universal the author considers the pattern, and a *sensitizing example* shows a picture of a real interface to illustrate the idea that the pattern captures. This is followed by a *problem statement* explaining what UI design problem the pattern addresses, and a set of *examples* or other empirical results are then used to show how this problem has been solved in similar ways in different systems.

These examples are generalized into the *solution*, a more reusable design guideline for the problem of this pattern. A *diagram* shows the essential idea of the solution in graphical form. Each pattern also refers to its *context* (when it should be applied) by pointers from other patterns in the language that address larger-scale design issues, and it itself refers in turn to smaller-scale patterns (its *references*) to consider next in order to implement and further unfold the design solution that this pattern suggests.

We have reproduced the overall graph of the pattern language in Fig. 1 to convey an idea of the design patterns that were considered for this project. As an example, the EASY HANDOVER pattern from that language makes the following design recommendation:

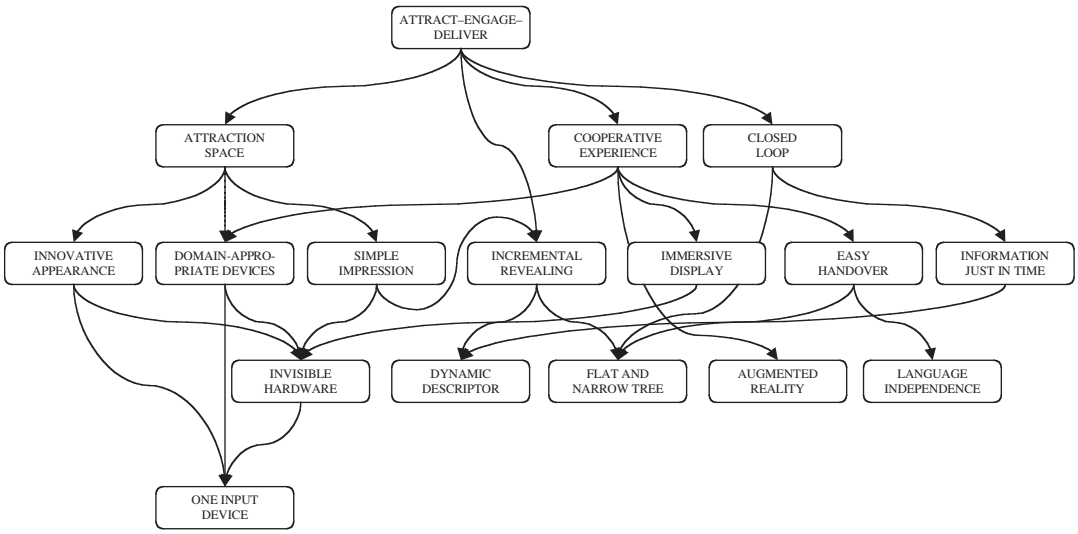

Figure 1: The HCI design pattern language for interactive exhibits [3].

- At interactive exhibits, one user often takes over from the previous one, possibly in the middle of the interaction, and without necessarily having observed or knowing much about the interaction history of his predecessor.
- Therefore, minimize the dialogue history that a new user needs to know to begin using the interactive exhibit. Offer an obvious way to return the system to its initial state. Let users change critical, user-specific parameters (such as language) at any time during the interaction.

Of course, this list is just the essence (the problem and solution statements) of the EASY HANDOVER pattern. The entire pattern consists of three pages of text and graphics, including examples of existing systems using this solution successfully, context and reference pointers to other pattern in the language, and all other pattern constituents listed earlier.

Nevertheless, this excerpt should convey an idea of how these patterns were able to help us design *Personal Orchestra*: For example, we used the above pattern to decide that no special gestures or other actions should be necessary anywhere during the conducting, to stop or otherwise control the exhibit. While these gestures could have been explained in the initial opening screens, there was no guarantee that any particular user would have actually seen those instructions.

HCI design patterns have recently received increasing attention [4]. While it is beyond the scope of this paper to further discuss this approach, it is explained in detail in [3].

User experience

Based on these design patterns, we dedicated an entire room to this exhibit, and worked with an interior design team to create an atmosphere reminiscent of the Golden Hall, the orchestra's home concert hall. Users enter this room, find wall-size facsimiles of the available pieces on wall tapestries, traditional note stands, and a red velvet conductor's podium to conduct. In front of them, a large rear video projection shows the orchestra, softly rehearsing, waiting for a conductor to become active (see Fig. 2).

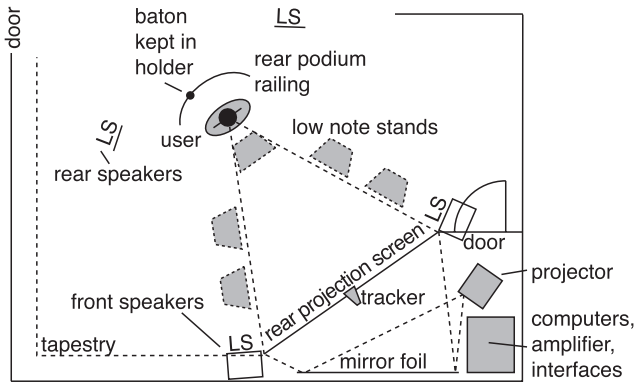

Figure 2: Overview of the *Personal Orchestra* exhibit space.

When a user picks up the infrared baton and presses the button on it (as indicated on the idling orchestra screen), a first screen appears, and by moving the baton up and down, the user controls a highlight on-screen to select one of the available languages. The selection is activated by pressing the button on the baton.

The subsequent screen explains how to conduct, and offers a similar mechanism to select a piece, or to learn more about the exhibit (see Fig. 6).

Once a piece is selected, the orchestra appears on the screen, waiting. When the user begins to conduct, the orchestra starts playing, following the conductor's gestures. The players continue until the piece is over, when they raise to congratulate the conductor with applause from the audience, or until they have decided that the user keeps conducting too badly (see below)...

Conducting gestures

The target user group could not be expected to know professional conducting gestures. The system therefore uses a sim-

ple "up/down" conducting style. Downward turning points of the baton trajectory would identify the conductor's beats. In accordance with traditional conducting, vertical size (amplitude) of the conducting gesture controls overall orchestra volume. Horizontal direction (conducting "towards" certain instrument sections) lets users raise those sections above the rest of the orchestra.

A realistic error message

Users had to be prevented from conducting too slowly or too quickly, for two reasons: technically, time-stretching or time-compressing the orchestra audio with sufficient quality is only possible to a certain extent. "Politically", the Vienna Philharmonic would not have liked the idea that exhibit visitors could make them play arbitrarily fast (and look arbitrarily silly in the process).

Our initial solution, a dialog box informing the user about his mistake, would have ruined the immersive experience. Instead, we invented a more natural and realistic error message: If the user "teases" the orchestra too much by conducting very quickly, slowly, or stopping completely, the orchestra reacts in the most natural way—they stop playing, and one player gets up to complain about the conductor's skills.

The system design was extended to include suitable tolerance rules for the orchestra (currently 8 beats of conducting too quickly or slowly), detect conducting that breaks these rules, and show the corresponding complaint sequence without noticeable interruptions.

Feature requirements

This design required the following major system features:

- a wireless baton-based input device to convey a realistic conducting experience on the device level,
- a gesture-tracking algorithm to determine speed, size, and direction of conducting gestures,
- a time-stretching playback algorithm to play the audio and video of the orchestra in real time, following the conducted tempo, volume, and instrument section emphasis,
- and a software framework to process events for language and piece selection, and to create the appropriate user interface responses and internal reactions.

IMPLEMENTATION

System architecture

Fig. 4 shows the resulting *Personal Orchestra* architecture. The visitor conducts using an infrared baton whose signals are picked up by a tracker and sent to the *POServer* machine. There, tempo, volume, and orchestra section emphasis are determined by gesture recognition and prediction. This "heartbeat" information is sent via our TCP-based *Personal Orchestra Control Protocol (POCP)* to the *POClient* computer, which renders the selected piece accordingly in audio and video, permanently adjusting playback parameters to follow the conducting.

During the initial selection of language and piece, and upon finishing or breaking off a piece, POServer sends similar

Figure 3: The *Personal Orchestra* exhibit in the HOUSE OF MUSIC VIENNA.

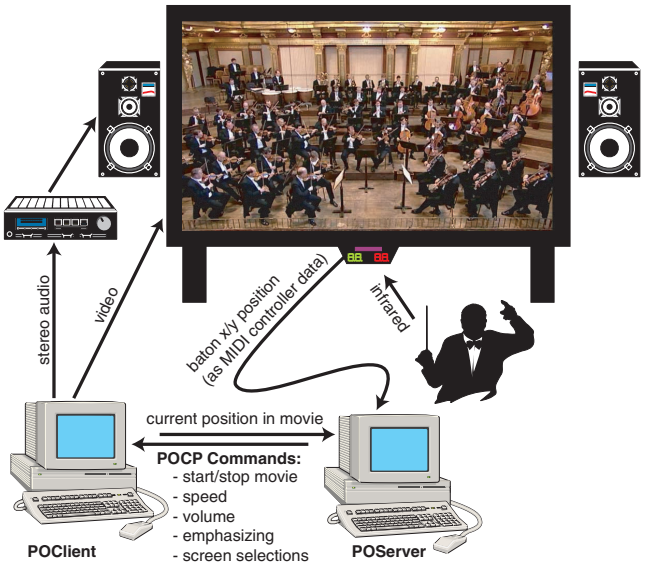

Figure 4: *Personal Orchestra* system architecture.

POCP commands to POClient to display the corresponding screens and movie sequences. POCP is a simple, HTTP-like protocol that sends textual messages about the current speed, volume, instrument emphasis and state from the server to the client, and returns movie positions from the client.

Input technology

We used Don Buchla's *Lightning II* infrared baton system [14]. It translates input from the infrared-emitting, battery-operated baton, received by a tracker mounted below the screen, into MIDI controller signals representing x/y baton coordinates with a resolution of 7 bit each. A third, binary controller signal represents the baton button.

Gesture recognition and prediction

From continuously monitoring the position of the baton, its current x/y position as well as approximations for its first derivatives are known. Every time the system detects a downward turning point in the gesture (negative-to-positive sign change of the first derivative of the y coordinate in the

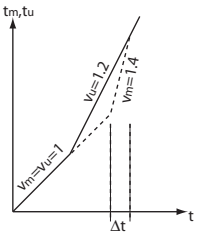

Figure 5: When a conductor speeds up, the orchestra has to play faster than the new tempo to get back in phase.

baton trajectory), it is interpreted as a "downbeat". These downbeats correspond to a series of positions in the movie marked manually as the "beats" in the music, using a simple utility we developed to take time-stamped key press inputs from a user tapping alongside a piece being played back.

User tests with an early prototype showed, however, that the conductor's perception of his conducting actually means that he expects the beat to be played shortly before the baton changes direction at the bottom of the trajectory; we incorporated this through a time offset that becomes added to the downbeat positions in the movie.

The current playback speed is then adjusted so that the orchestra always follows the conductor. There are two major problems with this, however:

First, a conductor may be conducting at the same speed as the orchestra plays, still the two may be out of *phase*—e.g., the orchestra would always play their "beat" half a beat after the conductor's downward beat gesture.

Second, when a conductor, e.g., speeds up, a part of the current beat has not been played yet when the next, first conductor beat gesture at the higher tempo arrives: the orchestra has to "catch up" with the conductor (Fig. 5). To get back in sync with the conductor, we adapted an algorithm known from the area of distributed systems: to synchronize clocks over a network requires changing the speed of the clock to be adjusted, instead of simply jumping ahead in time. The same is true for playback resynchronization: to catch up, playback speed has to be increased above the target (measured) new conducting rate for a while, until movie and conductor are at the same time in their piece, then it has to level back off to the actual new conducting speed, according to the following formulae:

Let b_u be the time of the last, and b'_u the time of the previous beat conducted by the user. Similarly, let b_m be the position ("time") of the last, and b'_m that of the previous conducted beat in the movie. Then the relative velocity (tempo) with which the user is conducting the movie is $v_u = \frac{b_m - b'_m}{b_u - b'_u}$.

Under the realistic assumption that, within a single conducted beat, the conducted tempo does not change dramatically, the current position t_u to which the user has conducted the movie at time t now equals $t_u = b_m + v_u \cdot (t - b_u)$.

Then, if the movie is currently at position t_m, the new relative velocity v_m of the movie ($v_m = 1$ for the originally recorded tempo) to catch up with the conductor within a time window of Δt is $v_m = \frac{\Delta t \cdot v_u}{t_m + \Delta t - t_u}$.

Of course, since this adjustment happens every beat, the catch interval is not used in its entirety if it is longer than one beat; instead, the movie speed will gradually converge back to the new conducted speed, reducing its over-estimate in a series of adjustments.

The larger the time window Δt in which this catching up happens is chosen, the more the orchestra creates the impression of being "slow to catch up"—it does not respond immediately to a tempo change, but rather over time, and it takes the orchestra longer to get back in sync with the conductor. The advantage is that short tempo jitter by inexperienced conductors does get filtered out; the orchestra is more "benign" and tolerant against such errors. We left this parameter as a variable that can be changed before running the system, to simplify adjustments in everyday use.

A similar low-pass filter was implemented for the instrument section emphasis. While it was technically easy to raise a section as soon as the user points to it, many users adopted a swinging conducting style that contained a lot of lateral (x) amplitude. This would have constantly shifted emphasis between instruments, which is a very unnatural behavior. Our final system only begins to react to an emphasis after the average of the conducting direction has remained in that section for a few beats. Again, this slows down reaction but makes the system more tolerant against conducting glitches.

High-quality interactive audio/video time-stretching

A broadcast-quality Digital Betacam video camera fixed to a position resembling the view of the conductor recorded the orchestra playing various pieces without a conductor. Its output was converted to AVID, a computer-compatible digital video format. Microphones throughout the orchestra recorded the various instrument sections onto ADAT digital audio tape. In order to synchronize audio and video simple clappers were used instead of an electronic synchronization technique like SMPTE, as relatively short pieces of audio and video had to be recorded—digitally. The challenge now was to adjust the speed of, or *time-stretch,* the orchestra movie being played back.

Time-stretching video is simple; most multimedia libraries easily handle changes in playback speed by repeating or dropping frames. As long as these variations do not drop below animation frame rates (around 12fps), and as long as there is no extreme movement that would create jerkiness at higher-than-normal speed (which is not the case with an image of an orchestra sitting and playing), the change of video playback speed creates no critical artifacts. While nonstandard playback speed creates unnatural movements (such as with respect to gravity—objects falling at slower than normal speed), this was also uncritical with our scenery.

The audio track, on the other hand, creates a problem since of course, just changing the speed of a PCM recorded wave audio file being played back will also change its pitch. It is relatively easy to avoid this by Fourier-transforming the audio signal: in frequency space, it is possible to change the

duration of a signal without changing its frequency. After inverse Fourier transformation, the resulting audio signal can be played in a time-stretched version at the same pitch. Another way to look at the same process is *granular synthesis*, which essentially cuts the audio signal into small packets of ca. 50ms and then repeats or leaves out packets to adjust tempo.

Unfortunately, these simple methods create noticeable artifacts in the audio signal. Typically, slowed-down Fourier-transformed versions will exhibit a strong reverberation component since all parts of the audio signal will have been prolonged equally. Granular synthesis needs to mix several signals into each other to avoid artifacts at packet borders, and even then fast attack sounds, if they happened to fall into a packet that is repeated, would sound twice.

Still, various algorithms exist that time-stretch audio in real-time. However, while some of these can, for example, speed up ("time-compress") recorded speech [5] with high intelligibility, these algorithms produce insufficient sound quality when applied to polyphonic, musical audio signals.

Essentially, audio data needs to be preexamined, even depending on music type, and time-stretched off-line to take care of these special cases, which takes a multiple of the original playing time to create good audio results. Naturally, it is only a matter of time before affordable hardware can do these computations in real time. However, at the time Personal Orchestra was implemented, the ratio of processing time to signal length was still far away from real-time performance (32:1 on a Pentium III/450 processor).

For that reason, we intended to pre-time-stretch all our audio channels at various speeds, and then, for each channel in parallel, crossfade between its pre-stretched versions to change playback speed. That way, time-stretching would take place only once during development for each channel and speed required, taking as much time as necessary to produce the best possible audio quality. During playback, all to be done would be to determine the new tempo required, and smoothly crossfade all four audio channels from their current tempo track over to their newly selected one within a few milliseconds. (This crossfade is necessary to avoid the audible clicks that the audio waveform discontinuities during an immediate track switch would create.)

However, this would have introduced different time coordinate systems for each audio track. To avoid this, and benefit from the system support of a single movie file with one video and multiple audio files, we *pitch-shifted* the audio between -1 and +1 octave, in half-tone steps of a factor of $\sqrt[12]{2}$. Playing back, for example, an audio recording that has been pitch-shifted down one octave at double speed returns the original pitch at double tempo. This way, we were able to integrate all pitch-shifted audio tracks with the video track, and a tempo change simply meant fading over to the appropriate audio track, and simultaneously changing playback speed of the entire movie to bring that audio track back to its original pitch. We used high-quality pitch-shifting algorithms from a commercial time-stretching software package, Ensoniq's TimeFactory (distributed by Steinberg, Inc.).

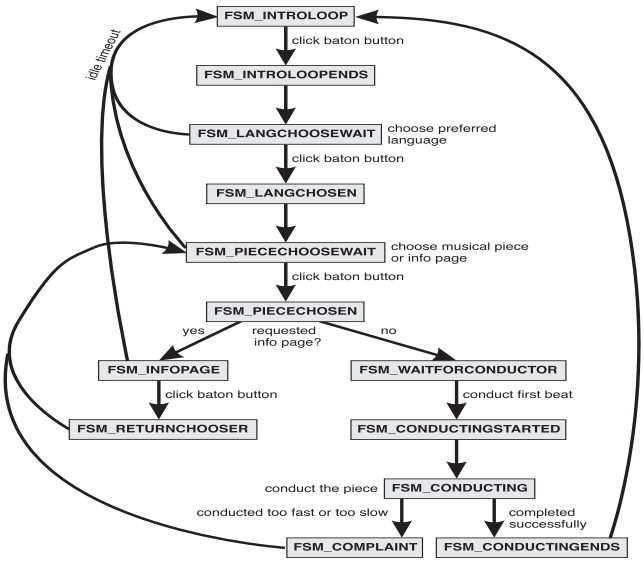

Figure 7: The Finite State Machine of *POServer*.

The emphasis between different instrument sections was simply implemented by pitch-shifting our four recorded and pre-mixed instrument section channels separately, and mixing them according to emphasis during playback in real time.

Navigation

As indicated in the Design section, *Personal Orchestra* plays a movie loop of the orchestra rehearsing until a user picks up the baton and presses the button on it. The orchestra disappears, and the user selects her favorite language and piece by moving the baton up and down and pressing the button.

Fig. 6 (left) shows the initial design of that screen (German versions were done only at that time). User feedback showed that users were missing some basic explanations on how to conduct. Also, while suitable for the musical atmosphere of the exhibit, the design did not look modern enough. The final design took these comments into account (right).

After selecting a piece, the orchestra appears again, waiting for the user to start conducting. The conducting ends either in the orchestra complaining when the conducting is too bad for several beats in a row, or with the end of the piece, with the orchestra raising and a big round of applause from the invisible audience behind the conductor. The state diagram of the system is shown in Fig. 7.

Hardware and software

The client and server software was implemented in Java. After initial experiments with Microsoft Windows and its DirectX/DirectMedia interfaces, we decided to use two Apple Power Macs G4/500 running Mac OS 9 and QuickTime, since they provided a more appropriate multimedia environment for our particular development and exhibition needs. Audio and compressed video are streamed directly off the hard disk. Video is projected via a rear-projector attached to POClient, audio is fed from the same machine into a 2x2 high-end speaker setup with front and rear speakers to en-

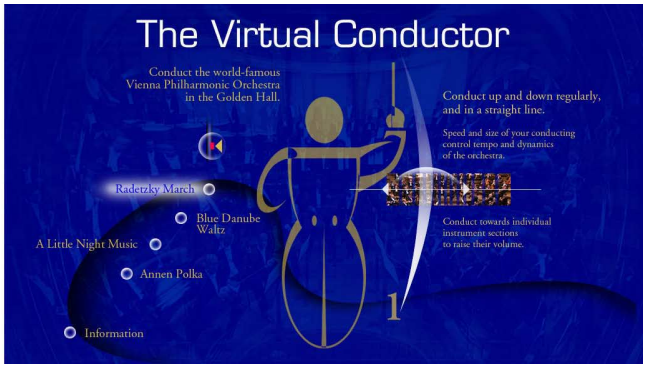

Figure 6: Initial (left) and final design (right) of the screen to select a piece in *Personal Orchestra*.

able sound locating as well as creating an audio ambiance of the orchestra filling the room.

EVALUATION

User observations

In addition to user feedback during the iterative design and prototyping of *Personal Orchestra*, we conducted several user studies of visitors using the exhibit in the first few weeks after the opening of the HOUSE OF MUSIC VIENNA. In an initial round, we did qualitative observations, and noticed that very few users managed to conduct a complete piece successfully. On the other hand, our "error message" of the orchestra complaining turned out to be a major attraction of the system for users, who would frequently try to intentionally provoke a complaint from the orchestra. However, it often happened that the user was trying to get acquainted with the system, but before having finished that phase, the orchestra had already stopped and was complaining. This was very frustrating for users, as well as the other chance of failing shortly before the end of a piece. We therefore increased the tolerance of the orchestra by fine-tuning our parameter sets, and introduced safety zones at the first and last few seconds of each piece to avoid those frustrating error situations. We also discovered that people did not read the signage at the exhibit, and instead just looked at the idle loop, wondering what to do. We therefore rendered a single sentence (in both languages) into the idle loop encouraging the user to pick up the baton and push its button.

After those improvements, we did another study where we observed, and then interviewed 30 random users between 9 and 67 years, with a wide variety of reported educational, musical, and computing backgrounds. The average user tried to conduct 2.4 pieces. Average usage time was 5.9 min. 97% of all users managed to do basic conducting gestures recognized by the system. 93% of all users realized that they could control tempo, 77% that they could control volume, and 37% that they could control emphasis of instrument sections. 60% of all users managed to finish conducting a piece without errors, 27% did so on their first attempt. On a scale of "good", "mediocre", and "bad", 81% judged audio quality to be "good", the remaining 19% "mediocre". Video quality was judged "good" by 75%, "mediocre" by 21% and "bad" by 4% (one user). 93% voted the exhibit to be a top three exhibit in the HOUSE OF MUSIC VIENNA.

Conductor feedback

Although *Personal Orchestra's* target users are museum visitors who generally are not experienced conductors, we made some interesting discoveries when we had two professional conductors use the system. One conductor claimed that whenever the orchestra started lagging behind, and he tried to speed them up, the players would suddenly start playing double-time. What was happening?

In order to get a lagging orchestra back to their desired tempo, professional conductors generally switch to more accentuated, exaggerated conducting movements: they increase the "wrist-flick" component of their downward conducting gesture. This by itself would not pose a problem, since our system would still determine the bottom turning point of the gesture correctly. It turns out, however, that this wrist-flick leads to a subsequent momentary leveling-off (a stopping point in the trajectory) of the baton on its way back up, when the wrist is relaxed again. While this would still be acceptable for our system, the added weight of the electronic baton in comparison to a standard, wooden conductor's baton leads to a second, small ditch in the baton's trajectory—which the gesture recognizer interpreted as a second turning point, i.e., double tempo being conducted...

While it is technically fairly straightforward to filter and correct these misinterpretations, their real value for us lay in the fact that they clearly indicated two things:

1. A serious conducting system needs to be adaptable to the conducting style and experience of the user. As an exhibit, our system is not intended to be tailored to each individual user, but we do provide a set of parameters that can be changed in a configuration file to change the "immediacy" with which the orchestra reacts to changes in the volume, instrumentation, and tempo conducted.

2. Conducting is a much richer form of interaction than is obvious at first sight. Tempo, volume, and instrumentation are the major, but only the basic functions that it communicates. An orchestra can in fact perform a piece flawlessly without a conductor (as seen in our own recordings of the Vienna Philharmonic), but the conductor is

there to give life and personality to a rendition, especially during the rehearsal period. *Personal Orchestra* fulfilled its goal—to provide a basic, understandable conducting experience to a wide variety of users. But it remains a fascinating challenge to capture those intricacies of conducting for more advanced and professional users.

FUTURE WORK

We are currently working on several aspects of a next-generation version of Personal Orchestra. Research directions include improving our tempo following algorithms, improving the usability of the system, designing a rendering engine that implements continuous (as opposed to the current discrete) tempo adjustment, and migrating the code to a faster programming language and a more modern operating system, to be able to run those advanced algorithms and at the same time achieve additional improvements in audio and video quality and realism.

SUMMARY

We designed and built an interactive exhibit that lets users conduct a realistic audio and video rendition of the Vienna Philharmonic Orchestra. Interaction takes place via an infrared baton; natural conducting gestures control not only volume and instrument section emphasis, but also the speed of the orchestra playing—although the orchestra will not tolerate notoriously bad conducting. We used our language of HCI design patterns for interactive exhibits to inform our design, and solved several complex technology issues in order to create the experience we had in mind. The main technical contributions are the methods that make it possible that *Personal Orchestra* provides not synthetic MIDI/VRML, but real audio/video data of this orchestra to interact with, and that it manages to let users influence the speed of this multimedia data stream being rendered in real time with few noticeable artifacts. Its orchestra complaints also offer an intriguing example of turning an error situation in an interaction into a feature adding to the realism of the experience.

Ultimately, *Personal Orchestra* is an example of an *Empowering Interface:* It offers not a gradual improvement in usability or performance for an existing situation, but rather opens up an experience to people that had been completely unreachable for them before. We found this type of interface to be very hard to design correctly, but also very satisfying to build, since its success can be seen daily by watching visitors enjoying, usually for once in their lives, the experience of conducting the Vienna Philharmonic.

ACKNOWLEDGEMENTS

Stefan Seigner, founder and director of the HOUSE OF MUSIC VIENNA, was always open to our project suggestions. The Vienna Philharmonic kindly donated their time and marvellous playing to this project. Many others, in particular Robert Hofferer, Christian Bauer, and Dominik Nimptschke from the HOUSE OF MUSIC VIENNA, and Ingo Grüll, helped turn this idea into reality.

REFERENCES

[1] Jordi Bonada: Automatic technique in frequency domain for near-lossless time-scale modification of audio. *Proc. ICMC 2000*, ICMA, San Francisco, 2000.

[2] Jan Borchers: WorldBeat: Designing a baton-based interface for an interactive music exhibit. *Proc. CHI 1997*, pp. 131–138, ACM, 1997.

[3] Jan Borchers: *A pattern approach to interaction design.* 264 pages, John Wiley & Sons, New York, 2001 (http://www.hcipatterns.org/).

[4] Jan Borchers and John C. Thomas: Patterns—what's in it for HCI? *CHI 2001 Ext. Abstracts*, ACM, 2001.

[5] D. W. Griffin and J. S. Lim: Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing,* ASSP-**32**(2):236-243, April 1984.

[6] T. Ilmonen: The Virtual Orchestra performance. *Proc. CHI 2000*, ACM, 2000.

[7] M. Lee, G. Garnett, and D. Wessel: An Adaptive Conductor Follower. *Proc. ICMC 1992*, ICMA, San Francisco, 1992.

[8] K. MacMillan, M. Droettboom, and I. Fujinaga. Audio Latency Measurements of Desktop Operating Systems. *Proc. ICMC 2001*, ICMA, San Francisco, 2001.

[9] Teresa Marrin: Possibilities for the Digital Baton as a general-purpose gestural interface. *Proc. CHI 1997*, ACM, 1997, pp. 311–312.

[10] Teresa Marrin Nakra: Inside the Conductors Jacket: analysis, interpretation and musical synthesis of expressive gesture. *PhD thesis, Massachusetts Institute of Technology,* 2000.

[11] Max V. Mathews: The Conductor Program and Mechanical Baton, in Max V. Mathews and J. R. Pierce, eds.: *Current Directions in Computer Music Research,* MIT Press, Cambridge, 1991.

[12] Peter S. Maybeck: *Stochastic Models, Estimation, and Control, Volume 1.* Academic Press, New York, 1979, http://www.cs.unc.edu/ ˜welch/kalman/maybeck.html.

[13] David Pogue: The dangers of the Digital Orchestra. *New York Times Direct,* Apr 5, 2001.

[14] R. Rich: Buchla Lightning II. *Electronic Musician,* **12**(8), Cardinal Business Media, Emeryville, CA, August 1996, 118–124.

[15] Music Box (now closed), Sony Center Berlin, 2001, http://www.sony-center.de/sonycenter_eng/-entertainment/music_box/c_index.html.

[16] S. Usa and Y. Mochida: A conducting recognition system on the model of musicians' process. *Journal of the Acoustical Society of Japan*, **19**(4), 1998.