Musical Design Patterns: An Example of a Human-Centered Model of Interactive Multimedia

Jan Borchers, Max Mühlhäuser
Telecooperation Research Group
Dpt. of Computer Science, Linz University
Altenberger Str. 69, 4040 Linz, Austria
{jan,max}@tk.uni-linz.ac.at
http://www.tk.uni-linz.ac.at/˜{jan,max}

Abstract

Most multimedia data today is sampled and quantized from analog sources. Even with sophisticated recognition and indexing techniques it remains difficult to associate human-centered, semantic content and structural information with it. This prohibits many desirable, more advanced interaction metaphors with the data than simple replay.

We propose that a human-centered model of both multimedia data and suitable metaphors to directly and instantaneously interact with it is crucial to the design of more interactive and "multimedia-aware" system and application architectures.

We show how we implemented such a model for the media type 'music' in the WorldBeat *system, a highly successful interactive computer-based music exhibit in the Ars Electronica Center in Linz, Austria. The system uses a high-level semantic concept of musical information called "Musical Design Patterns". Interaction with this representation, like spontaneous, computer-supported improvisation, is supported in a novel way using just a pair of infrared batons to control the entire exhibit. The system and its semantic model offer a technologically and artistically innovative approach that should be of interest to multimedia researchers as well as educators, artists and performers.*

Keywords

multimedia semantics, interactive exhibit, music, improvisation, user interfaces, batons

1. Introduction

Have you ever tried to locate a certain MPEG movie on your hard disk when all you have are filenames like "bldng013.mpg"? We call this the *multimedia semantics problem*: While traditional, text-based data can usually at least be full-text-searched (i.e., on a syntactic level), there is hardly any way to get to the "contents", or semantics, of traditional, let alone multimedia data. Multimedia contents are brought into the system by sampling and quantization processes that are highly optimized on a syntactical level – encoding, data compression, etc. –, but that act quite "brainless" on the semantic level: What is inside a video clip, audio file, or other multimedia data object remains unknown.

The digitizing processes are not to blame, though: What is missing is a model on a higher, semantic level that represents some information about the contents of the data recorded. This information may be derived from syntactical analysis (e.g., pattern matching) on the digitized data, or it may incorporate external semantic knowledge about the data into the system. Such a model would offer more sophisticated ways to query the data, and to interact with it in real-time, as appropriate to the medium.

Many ambitious research projects, e.g., [8], currently reconstruct multimedia semantics through syntactic analysis. Results confirm, however, that this task is mostly inappropriately harder and less satisfactory than storing content information (semantics) at creation time.

Therefore, we will focus on a model that mainly uses information made available when multimedia data is created, supported by structural knowledge derived from data analysis. We will describe such a model for musical data, with suitable interaction metaphors to work with it. We will show how we turned our model into a working system, the interactive *WorldBeat* exhibit about computers and music.

2. Background

To understand our approach, it will be helpful to know something about the environment and its external constraints that influenced the design of our *WorldBeat* exhibit.

2.1. The Ars Electronica Center (AEC)

The *AEC* in Linz, Austria, is a "technology museum of the future" [11] discussing how technology will influence our lives in the next century. Each floor deals with different aspects of modern information technology, from new Internet applications to 3-D virtual environments. It opened in September 1996.

2.2. The Knowledge Net Floor

Our research group designed the second floor of the AEC, called *Knowledge Net*. It shows how computers and networks may change learning and working environments.

Half of this floor contains our *Conference/Classroom of the Future (CCF)* with networked workstations on flexibly configurable desks, interactive whiteboards, and equipment for computer-supported cooperative work, video conferencing, etc. It is not only an exhibition area, but also a usable environment for meetings, seminars and other (tele-)collaborative activities.

The other half contains individual interactive exhibits, demonstration stations dedicated to specific subject fields like new media, new user interfaces, and new learning approaches. *WorldBeat* is one of these stations.

Our primary goal when designing this floor was to deliver the following key *messages* to the visitor:

- "Life-long learning" on the one hand, and the ideal of the independent learner on the other hand, make learning and working converge into a process that we call "cooperative information processing".
- This process can become more *cooperative, active,* and *motivating* if computers support it sensibly.

More about Knowledge Net design and implementation issues can be found in a different paper [15].

3. State of the Art

As computer scientists, we had one strong constraint on our musical semantics model: It had to be exact and computable enough to build a system on top of it. Moreover, for an interactive exhibit, it had to be efficient enough for real-time processing and interaction.

3.1. Multimedia in User Interfaces

With our emphasis on a connecting a semantic model of multimedia and interaction techniques to work with it, it is useful to first give an overview of how multimedia and user interfaces have related in the past.

The last ten years have brought about an extreme excitement about *multimedia systems* on the one hand, and *multimodal interfaces* on the other hand. But while multimedia capabilities of systems available today have increased drastically, multi-modal interfaces have developed much less noticeable. This has lead to systems that can process and present multimedia data on a basic level, but that do not offer the user a semantic model and interface to interact with this data in fundamentally new and meaningful ways. Also, only few research projects actually work on encompassing semantic models of multimedia data that include useful interaction metaphors.

Of course, quite a number of excellent user interfaces have been built which apply new media (for a good introduction to multimedia user interfaces see, e.g., [1]). However, many of these systems represent expensive custom developments, often related to the design of entire computer systems (hand-helds, info terminals, etc.). And if a system-level multimodal user interface is design-centered around the specifics of a certain "device" type, it does *by nature* neglect specifics of the media used in an application.

Another long-lasting deficiency in multimedia user interfaces is the fact that software engineering has largely neglected the shift from function-centered, algorithm-driven software to software that is supposed to run in a closed human-machine interaction loop.

It was not until 1994 that software engineering and HCI communities first met in a workshop [19] to start merging their knowledge, models, and processes. We responded to this challenge with design methods and tools combining human-computer activity models with traditional object-oriented design, and with a modality-abstracting toolkit that shields software development from dependencies of the modalities and technology used [9].

This approach yields software products which can be easily adapted to new modalities. It proved to be extremely powerful for software to be used on a variety of devices with appropriate modalities (e.g., a WWW browser that works pen-based on pads, handwriting-enhanced on an organizer, speech-driven when privacy and computing power allow for it, gesture-enhanced when presenting to an audience, etc.).

The approach is not sufficient, however, if an application is tightly bound to specific media by nature, and if the interaction model is to take this into account. This is the part of the problem space that we emphasize here. We will discuss some comparable projects further below, focusing on interaction with musical data.

3.2. Semantic Models for Multimedia

For this field, conferences like the series on "Multimedia Modeling" should provide an excellent source [3, 4, 6]. A closer look, however, indicates that, by and large, they provide models of the following domains only:

- Synchronization models for temporal and spacial coordination of different media "streams"; these models deal with technical characteristics of the media, but not with human-centered contents semantics.
- Models for representing sampled digital signals, adapted to processing challenges like compression, transmission, quality enhancement, editing, etc. They are even more technology-oriented.
- Models for automatic recognition (speech understanding, computer vision, etc.). While these are of course centered around understanding "contents", they seek a match between signal representation and representation of an ontology of the "objects" in the domain (e.g., syllables, or talking heads). As argued before, this approach retrofits information into signals which rather ought to be modeled and captured at the origin of the media right away.
- Models of general application domains, like teaching (e.g., instructional strategies) or exploratory documents (such as hypermedia). Here, up to now media are mostly "plugged in" as different, transparent ways of conveying contents, without taking into account their specific strenghts and weaknesses: Most models do not incorporate knowledge about "what medium to use when", let alone specify how to model a given medium to elicit its strengths for the conveying purpose.

All in all, the manyfold modeling attempts do not serve our purpose very well. There is, however, increasing awareness about this problem. A paper about computer music [18], for example, expressed the "forgotten" importance of interactive use of multiple media: "*The quality of performance often overrides whether what is being performed has come from a struggling genius, a commercial hack, chance decisions, or even a computer program*".

3.3. Trends in Computer Music

With the tremendous development in digital audio synthesizing and processing systems, real-time playing of fixed scores has become a feature of electronic music instruments at the consumer price level. These systems usually lack, however, a useful, more abstract model of the musical data than is represented through the MIDI (Musical Instruments Digital Interface) standard described below.

Modern computer music theorists, on the other hand, have developed highly complex systems, trying to capture the intelligence and creativity of musical composition, but they are generally "batch-processing" systems that cannot react to external stimuli and control input in real-time, and aesthetically their results are mostly still far from comparable to the work of mediocre composers, let alone truly gifted ones.

Thus, most computer music is either a creative, real-time playing activity where the computer just serves as universal instrument without a built-in semantic or didactic concept, or it is a scholar's activity with complex underlying semantic models, but trying to let the computer "be creative" instead of the human, with moderate results. We want to bridge this gap.

3.4. Existing Models and Interaction Metaphors for Musical Data

The lowest level of modeling musical information in computer-readable form is as *digitized audio*, e.g., by sampling an analog source. While this can be done with very high perceived accuracy (consider the Compact Disc), it does not create any model whatsoever of the actual contents and structure of the music on a level above physical signals. Interaction metaphors therefore are mostly derived from the analog world, displaying signal waves, and offering signal-processing operations like cut, paste, or maybe frequency filtering and signal mixing, on the data. Higher-level operations are possible, but the model has to be built from scratch, e.g., by creating "macro" libraries of sound files and using them to "compose" some musical material. Moreover, data volume is naturally large, with high demands on storage capacity and processing power.

MIDI, the Musical Instruments Digital Interface standard, was established in 1983, and has since revolutionized the world of electronic music. It creates a first layer of abstraction from the unstructured digitized audio stream, by introducing the concept of *messages* that are created for each note as it is begun or ended. Those messages encode just instrument, pitch, and velocity of the note, from which the sound-generating hardware then recreates an audio signal as close to the original as possible.

Of course, this makes perfect reproduction of a natural audio signal not generally possible anymore, for two reasons: First, the (usually sampled) instrument sound may not sound realistic. Second and more fundamentally, however, many remaining, hardly quantifiable performance parameters, like perceived brightness, expressiveness, playing noise, etc., are not recorded in the MIDI stream. Special MIDI controller messages can give additional vibrato and similar information, but they remain "playing hints" for the sound generator.

Nevertheless, MIDI offers fundamental advantages: It reduces the data volume enormously (MIDI uses 31kHz connections, mostly without noticeable delays). More important, however, is that it opens up new types of interaction with musical information. The most common application is the *sequencer*, hardware or software to record, store, edit, and play back sequences of MIDI events. Even though it is usually designed like an analog tape deck, its features are much more powerful because of the higher-level editing operations it offers that deal with notes, not audio signals. Since recently, computing power has even allowed for mixing sampled and MIDI data and converting it into each other, blurring the boundary between both representations.

Numerous interactive devices and systems have been developed to work with MIDI data; a good example are the *infrared batons* [16] that convert 2-D movements into MIDI controller and/or note messages. They are described in more detail in the next chapter. One of the first baton MIDI controllers was developed by Mathews [14]. Marrin [13] developed a digital baton with several additional built-in sensors that react to pressure and other input, to control various parameters in musical performance. Roh et al. [17] used an electronic drum pad as touch-sensitive input for an exhibit about Indian drumming styles. Another example is the *Query By Humming* concept presented in [10] which we incorporated into our exhibit (see *Design* below).

Other systems focus more on the goal of modeling stylistic information. One of the most prominent projects in this area is Cope's *EMI* (Experiments in Musical Intelligence) system [5]. It inherits aspects of a composer's style by examining sample music files, and then composes new music in this style. It is not, however, intended for interactive use in the supportive way we envisioned.

A rather successful, though less ambitious, commercial example is the *Band In A Box* software. Here, a few mouseclicks are enough to start an accompanying band in any of many predefined musical styles, and new styles can be programmed. However, it is again purely batch-oriented: Once chord progression and style have been defined, what the system plays depends only on those fixed parameters.

The reason is that those systems are designed to accompany a rehearsing player. Recently, even an improvisation function was added to that particular system to rehearse accompanying, but it still does not offer computer-supported playing where creative input from the user is processed in real-time to modify it in intelligent ways.

With the increased performance of today's workstations, we now consider it feasible to explore the possibilities to create interactive systems that have some even more abstract concept of music semantics, at the same time still being usable in real-time performance like traditional instruments with added intelligence.

4. The WorldBeat Exhibit

This section summarizes the design rationale for our interactive music exhibit as a whole, and gives an overview of its implementation. It is inside this framework that we created the *Musical Design Patterns* module with its abstraction from simple MIDI data.

4.1. Design

The *Knowledge Net* environment dictated several constraints to *WorldBeat*: It had to be a computer-based, interactive exhibit that showed how computers could be used to experience new ways of playing, interacting with, and learning about music. As such, it required an innovative, non-technical, interesting appearance that supported cooperative, activity-oriented, and entertaining use as put forward in our "message". Nevertheless, its interface had to fulfil the usual requirements – be consistent, intuitive, comprehensible, and usable by people with a wide range of physical abilities – to an especially high degree, due to its nature as an exhibit in a public space. Of course, this also required robustness considerations.

On the functional side, we met these requirements by deciding that WorldBeat would consist of several different components, each addressing a certain aspect of computer use in music in detail:

Joy-Sticks: Visitors can play numerous virtual instruments, using the computer to simulate other instruments.

Virtual Baton: Visitors can conduct a pre-recorded MIDI piece played back by the computer, influencing its tempo and dynamics. The system is based on work described in [12], and shows how humans can make computers adapt to their own human requirements and metaphors when interacting with them about music.

Musical Memory: Visitors can try to recognize instruments by their sound, using the computer to play an interactive educational game.

Net Music: Visitors can exchange MIDI compositions with others around the world, or even play together with them via the Internet, showing the use of networks for distributed learning.

Query By Humming: Visitors can hum part of a tune to find the corresponding piece in a database. The system uses pattern recognition algorithms based on an idea presented in [10] to create a "musical" interface to musical data.

Musical Design Patterns: Visitors can "customize" the style of a generated Blues band, then improvise to this accompaniment without having to fear wrong notes. This component is described in more detail in this paper.

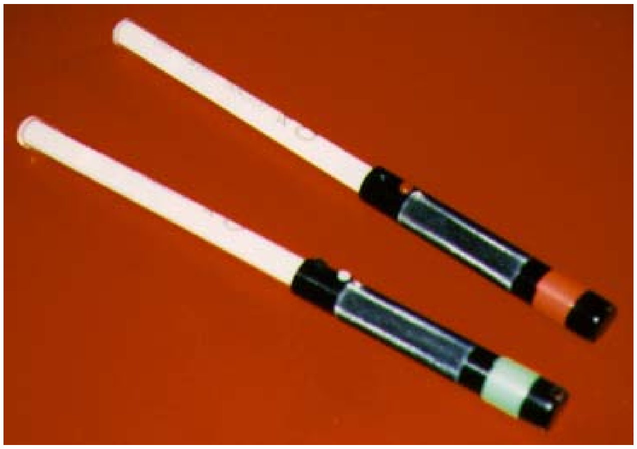


Figure 1. The infrared batons used in the WorldBeat exhibit.

On the *user interface* side, we met the requirements by using two *infrared batons* (see Figure 1) that deliver 2-D position information with an additional action button for special functions. They are used for all interaction with the exhibit, making the interface very consistent.

navigation: one wand replaces the pointing device;

conducting: one wand acts as conductor's baton;

playing: the wands are used to play virtual instruments;

improvising: both wands are used to play an "intelligent" instrument that helps avoiding wrong notes.

4.2. Implementation

The entire system was implemented and runs on an Apple Power Macintosh, using *MAX* [7], a development environment for real-time MIDI applications. It uses a visual programming paradigm to build hierarchical networks of MIDI-data-processing objects, or *patches*. We extended *MAX* with special patches, especially for the user interface engine that manages all input and output of the graphical user interface.

A typical usage scenario, then, would create the following interaction with the system:

- The user, standing in front of the exhibit (see Figure 2), holds the batons in her hands and points them at the monitor. She presses the baton action buttons to select links while navigating, or to trigger a special musical function while playing. The current interaction metaphor and meanings of button presses etc. are always explained on-screen.

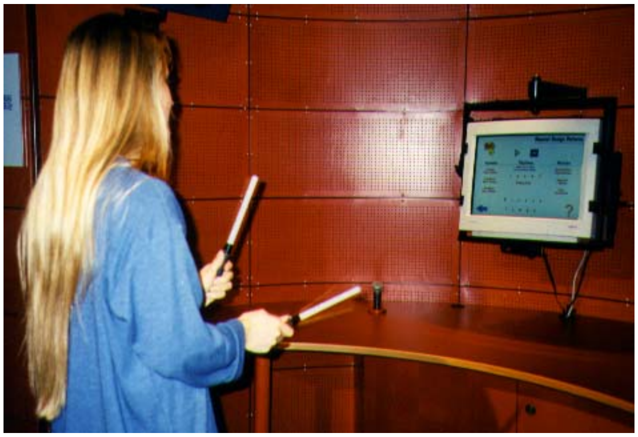


Figure 2. A visitor using the WorldBeat exhibit in the Ars Electronica Center.

- Both batons continuously send infrared signals, with special signals when action buttons are pressed.
- An infrared tracker below the monitor determines from those signals the position and action button state of each baton.
- This data goes to the baton system base unit which converts it to MIDI controller data, or MIDI notes using one of the built-in setups.
- Controller data gets to the Macintosh where our WorldBeat user interface engine triggers the corresponding user interface and/or musical events, depending on the currently active WorldBeat component.
- Notes to play are sent back to the sound card in the base unit. Visual feedback is created by a cursor that shows the current position on the screen as a light spot.

The baton interface, and general design rules for interactive exhibits, are presented in more detail in a different paper [2].

5. The Musical Design Patterns (MDP) Model

As mentioned before, the *Musical Design Patterns* component offers the user the possibility to configure certain parameters of an "electronic Blues band", and then lets him improvise to this band with computer support. We will have a look at the underlying model of this component.

At its lowest level, the MDP model builds onto the MIDI standard as a basis for working with concrete notes. That way, it does not have to care about the concrete sound synthesis etc. which is hidden by the MIDI definition: MDP is a layer on top of MIDI.

5.1. MDP Architecture

MDP consists of five conceptual parts:

- a *database* of predefined musical material;
- a *knowledge base* of musical concepts, or design patterns, in general and for Blues in particular;
- a *harmonic analyzer* that constantly determines the current harmonic context of the music as it is created from the database material;
- a *user input analyzer* that constantly examines the current infrared baton movement data and turns it into meaningful control information for the remaining system;
- and finally the central *MDP processor* which uses the above components to run the interactive music session.

The harmonic analyzer uses a *root-finding* algorithm to first decide on the root of a chord, after which it is easy to determine the actual harmony. It sends any harmony changes to the MDP processor.

The MDP processor then uses the knowledge base to find a set of harmonically matching notes for the current harmony, and extends this into a mapping of user input messages to musical messages. Whenever it receives user input from the UI analyzer, it uses the current mapping to determine the notes to be played. For quick access, most knowledge is stored directly inside *MAX* objects in a list-oriented manner. They contain, for example, scales of matching notes for each harmony, probability functions and markov chain generators for accompaniment sequences, or patterns for the bass line. These objects are part of the hierarchical patch network of the MDP component. Depending on the type of information stored – e.g., whether it is time-dependent or not – different object types, mainly set-like *collections* and sequence-like *tables*, are used for storage.

Of course, it would be possible to hard-code the harmonic information directly within the music database, and save the real-time recognition – but this would hinder the extension with new pieces, and make the system less universal. Precomputing the harmonies would make it less interactive and real-time capable. With our approach, however, the MDP component could easily be extended to receive harmonic input from, e.g., an unpredictable live performer.

5.2. Design Patterns and Interaction Metaphors

Users work with the MDP component in two different ways: First, they can interact with certain design patterns stored in the knowledge base via a graphical interface shown in Figure 3. Then, they improvise to the music using the above support system, probably the most important design pattern of this component. The concepts made accessible include

Harmonic Progressions: Several Blues progression patterns are stored in the database as mode sequences in standard MIDI format. Sorted by harmonic complexity, they are presented to the user. The pattern he selects is used as starting point for the accompaniment and the computations of the harmonic analyzer. The user can thus experience directly its effect on accompaniment and improvisation.

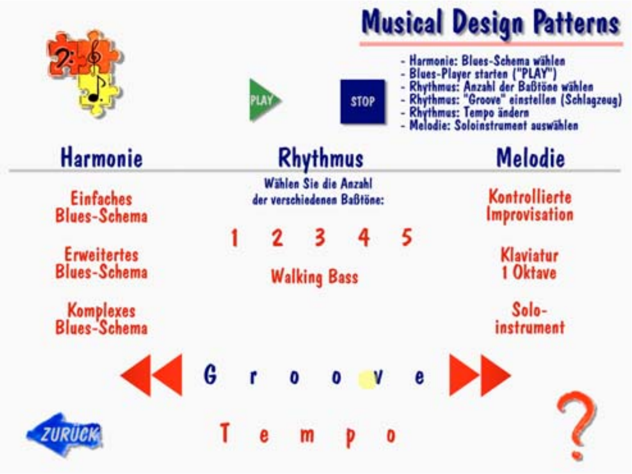
Bass Lines: These patterns are modeled as Markov chains, but with a variable size of the transition graph: The set of notes, determining complexity of the bass line, can be changed by the user. Alternatively, he can select a walking bass pattern to hear the difference in sophistication compared to computer-improvised bass lines.

Tempo: The tempo of the "band" playing can be changed in real-time via an on-screen slider to control the basic MDP processor performance speed.

Groove: The *groove* of the drummer, another important jazz music design pattern, is also directly accessible: A slider controls where inside each beat the drummer will put his intermediate cymbal hit (see Figure 3). The slider setting gets fed into the knowledge base which then tells the MDP processor when to trigger the next intermediate beat. The knowledge base also contains a pattern which automatically lets the bass player "pick up" on this groove occasionally, for a more varied bass line. Adjusting this parameter opens up an entirely new way to understand this important, but hard to describe feature of jazz music: By sliding the drummer's intra-beat timing back and forth, visitors get an immediate idea of the differences between a "straight" timing (like in a march), a triolic, or a "laid-back" timing (as in jazz), which is quite hard to explain and understand otherwise: "inter-active" learning by doing here really makes the difference.

Timbre: To demonstrate the importance of instrumentation as a design pattern in musical performance, the solo instrument sound to use when improvising can also be changed from a list of instruments with suitable playing metaphors at any time (vibraphone, xylophone, marimba, piano, etc., see below).

Improvisation Freedom: The user can choose how much support he wishes to receive from the MDP system when improvising: Beginners usually choose "computer-supported improvisation" in which the design pattern described above comes into action: The



Musical Design Patterns

- Harmonie: Blues-Schema wählen
- Blues-Player starten ("PLAY")
- Rhythmus: Anzahl der Baßtöne wählen
- Rhythmus: "Groove" einstellen (Schlagzeug)
- Rhythmus: Tempo ändern
- Melodie: Soloinstrument auswählen

Figure 3. The main screen of the Musical Design Patterns component in the WorldBeat exhibit. The user is just changing the 'groove' parameter.

system determines a scale of matching notes to choose from. However, improvising more freely is also possible (see Adaptivity below).

When satisfied with the accompaniment, the visitor can play along with the Blues band. The playing metaphor is defined in the user input analyzer. It can be arbitrary, but we use a simple and natural mapping of "playing a xylophone with two mallets": The visitor stands in front of the exhibit and makes downbeat gestures with the two batons in his hands. Gesture velocity determines volume, horizontal position determines pitch. That way, it almost behaves like a "normal" xylophone.

However, in reality the underlying keyboard of the virtual xylophone is constantly changing – each time the accompanying band changes to another chord, a matching set of notes is computed, replicated over several octaves, and put as "new notes" onto the invisible keys. This process uses the built-in knowledge about musical theory, harmonic context, and style-specific information like the pentatonic Blues scale to infer a set of matching notes.

This results in an interesting experience for the visitor: He has full control over three of the four fundamental degrees of musical freedom – rhythm (at which moment he plays), dynamics (how loud he plays), and melody (whether he plays high or low notes, runs, etc.). The system only takes care of the harmonic dimension, carefully mapping the keys played to the nearest harmonically sound note.

5.3. Adaptivity

Musically experienced visitors might complain (and frequently do) that this limits creative freedom of the performer – she cannot play any note she wishes, because the system always maps her input to matching notes.

We catered for this by extending the user interface analyzer to support different levels of user input processing: Apart from the mapping described above, the user can also choose to play a chromatic keyboard without harmonic mapping. The keyboard has a standard piano layout (over one octave only, so that the region for each "key" remains large enough to hit it correctly in the air), with the black keys being put above the white keys (like a "tipped-over xylophone"). With this metaphor, users can play any notes they wish to the accompaniment – of course, in this case they are responsible for wrong notes. This shows, however, that computer-based interactive learning systems can remain interesting and challenging for much longer if they are adaptive or adaptable to the learner's experience level.

5.4. Extensions

We have defined further musical concepts that we are currently implementing. For example, we will allow users to play disharmonic notes for a short duration, however with an automatic shift to the nearest harmonic note after that. The goal is to allow slightly more experienced users to play "sliding blue notes". In addition, we have extended the scale computation algorithm to create different sets of notes relative to the base note, depending on the modus the accompaniment is in. We are also working on another kind of support in the rhythmic dimension which quantizes the user's input to the next sensible timing point. We believe, however, that any rhythmic manipulation will be perceived by users as much more noticeable and irritating than melodic or harmonic manipulations, due to the interactive value of immediate feedback. Nevertheless, such an approach could be interesting for very slow music, and especially for people with motorical disabilities.

6. Evaluation and Conclusion for Human-Centered Interactive Multimedia Models

The success of a system like WorldBeat must ultimately be rated by the opinion of visitors that use it. During the opening week, the authors watched thousands of users at the exhibit, and talked to hundreds of them. Thanks to our user-centered design approach, only minor changes were necessary (see [2] for dertails).

It turned out that the MDP component was the most famous part of WorldBeat. With its interface that looks neither technical nor professionally musical, we managed to get people to improvise freely who insisted on being completely unmusical before, and who would never have played on a traditional keyboard in front of dozens of other people. Also, visitors of virtually all ages, and with a wide range of physical abilities, have used the system, finding especially the MDP setup very satisfying, and simply "fun". Because of the approximate nature of playing with the batons in mid-air, users have never complained (or even noticed!) that they are playing on a keyboard that changes its notes frequently.

The missing kinesthetic feedback was also accepted very well, mainly because the baton hardware and fine-tuned software create very prompt acoustic feedback without noticeable delays in musical interaction.

MDP has obviously found a good balance between user-induced, creative, free activity, and computer-based support and guidance. We think that this middle course between non-didactic real-time MIDI, and batch-oriented composition software is very promising, and will continue our research in this area.

A survey of over hundred AEC visitors confirmed this: WorldBeat was among the three most popular exhibits of over one dozen major AEC stations. This is especially interesting when taking into account that the two more famous stations were between 10 and 50 times more expensive than WorldBeat. This success can be put down to the fact that we designed a system that delivers our message by putting it into reality: Visitors can become active from the start, they can use it together, and it wraps learning into an entertaining experience.

All in all, feedback suggests that WorldBeat, and especially the Musical Design Patterns component, is indeed a novel type of system that gives access to multimedia data in ways that were not known before. The MDP model, and its translation into a set of interface metaphors, have proved successful in this respect. This shows that new, good interaction metaphors are crucial to the success of interactive hypermedia exhibits. We believe that our system and its semantic model of musical information will be of interest to professionals from computer science, education, and the performing arts, and hope to foster an interdisciplinary discussion of interaction-oriented semantic models of music and other multimedia data.

References

[1] M. Blattner and R. Dannenberg, editors. *Multimedia Interface Design*. ACM Press, New York, 1992.

[2] J. Borchers. Worldbeat: Designing a baton-based interface for an interactive music exhibit. In *Proc. CHI'97*, pages 131–138, Atlanta, GA, Mar. 22–27, 1997. ACM Press.

[3] T.-S. Chua and T. L. Kunii, editors. *Proc. First Intern. Conf. on Multimedia Modeling*, Singapore, Nov. 9–12, 1993. World Scientific.

[4] T.-S. Chua, H. K. Pung, and T. L. Kunii, editors. *Multimedia Modeling: Towards Information Superhighway*, Singapore, Nov. 14–17, 1995. World Scientific.

[5] D. Cope. *Computers and Musical Style*. A-R Editions, Madison, WI, 1991.

[6] J. P. Courtiat, editor. *Proc. Third Intern. Conf. on Multimedia Modeling*, Toulouse, France, Nov. 12–15, 1996.

[7] J. C. Dobrian. *MAX Reference Manual*. Opcode Systems Inc., Palo Alto, CA, 1995.

[8] S. Fischer, R. Lienhart, and W. Effelsberg. Automatic recognition of film genres. In *Proc. ACM Multimedia 1995*, pages 295–304, San Francisco, CA, Nov. 5–9, 1995. ACM Press.

[9] H. W. Gellersen and M. Mühlhäuser. Design of workplace-integrating user interfaces based on work scenario graphs. In D. Benyon and P. Palanque, editors, *Critical Issues in User Interface Systems Engineering*. Springer Verlag, Berlin, 1995.

[10] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith. Query by humming: Large musical information retrieval in an audio database. In *Proc. ACM Multimedia 1995*, pages 213–236, San Francisco, CA, Nov. 5–9, 1995. ACM Press.

[11] S. Janko, H. Leopoldseder, and G. Stocker. *Ars Electronica Center: Museum of the Future*. Ars Electronica Center, Linz, Austria, 1996.

[12] M. Lee, G. Garnett, and D. Wessel. An adaptive conductor follower. In *Proc. Intern. Computer Music Conf.*, 1992.

[13] T. Marrin. Toward an understanding of musical gesture: Mapping expressive intention with the digital baton. Master's thesis, MIT Media Lab, Boston, MA, 1996.

[14] M. Mathews. The radio baton and conductor program, or: Pitch, the most important and least expressive part of music. *Computer Music Journal*, 15(4):37–46, 1991.

[15] M. Mühlhäuser, J. Borchers, C. Falkowski, and K. Manske. The Conference/Classroom of the Future: An interdisciplinary approach. In *Proc. IFIP Conf. "The International Office of the Future: Design Options and Solution Strategies"*, pages 233–250, Univ. of Arizona, Tucson, AZ, Apr. 9–11, 1996. Chapman & Hall.

[16] R. Rich. Buchla lightning II. *Electronic Musician*, 12(8):118–124, Aug. 1996.

[17] J. H. Roh and L. Wilcox. Exploring tabla drumming using rhythmic input. In *Proc. CHI'95*, pages 310–311, Denver, CO, May 7–11, 1995. ACM Press.

[18] S. W. Smoliar. Computers compose music, but do we listen? In Chua and Kunii [3], pages 73–77.

[19] R. Taylor and J. Coutaz, editors. *ICSE'94 Workshop on SE-HCI: Joint Research Issues*, Sorrento, Italy, May 1994. Springer LNCS 896, 1995.