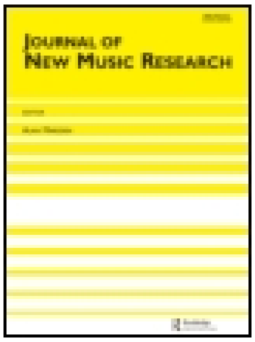

Journal of New Music Research

ISSN: 0929-8215 (Print) 1744-5027 (Online) Journal homepage: https://www.tandfonline.com/loi/nnmr20

Designing Musical Cyberinstruments with Body and Soul in Mind

Tamas Ungvary & Roel Vertegaal

To cite this article: Tamas Ungvary & Roel Vertegaal (2000) Designing Musical Cyberinstruments with Body and Soul in Mind, Journal of New Music Research, 29:3, 245-255

To link to this article: https://doi.org/10.1076/jnmr.29.3.245.3088

Published online: 09 Aug 2010.

Journal of New Music Research, 29 (2000), No. 3, pp. 245–255
0929-8215/00/2903-245$15.00

DESIGNING MUSICAL CYBERINSTRUMENTS WITH BODY AND SOUL IN MIND

Tamas Ungvary[1] and Roel Vertegaal[2]

[1]Dept. of Speech, Music and Hearing, Royal Institute of Technology, Stockholm, Sweden
[2]Department of Computing and Information Science, Queen's University, Canada

ABSTRACT

For centuries, traditional musical instruments have been built to control specific types of sound production processes according to the same principles: their control and sound generation parts are typically tightly integrated with a solid physical interface. Current-day digital counterparts lack such requirement of physicality in their control, and very few last more than a couple of years. In this paper, we argue that the flexibility of computer music instrument designs has inhibited the development of dedicated control structures, as well as the training of musicians with associated motor skills. As a candidate solution to this problem we present SensOrg, first in a class of Cyberinstruments that we envision to be a more generic and hopefully long-lasting class of control structures for digital instruments. SensOrg is a hand controlled Cyberinstrument, sculpted from a modular assembly of input/output devices and musical software that are mapped and arranged according to essential functional characteristics of the Human-Instrument system. We discuss how the cognitive ergonomics of non-verbal performance and symbolic compositional tasks influenced the design of our hardware interface. We also sketch the design of our underlying software framework, IGMA (Interactive Gesture Mapping), which provides interactive musical functionality for improvisation, composition and performance alike.

INTRODUCTION

Musicians strive many years in order to connect their neural pathways to a vibrating segment of string, wood, metal or air. Even when musicians master their instruments, their sweet sorrow is not over. The chin marks of violinists, and the Repetitive Strain Injuries of pianists demonstrate the problems musicians face in the every day maintenance of their mastery. One might argue that the high learning curves and the physical contortion are symptoms of bad ergonomic design of musical instruments. In this paper, however, we will take the *opposite* standpoint: there is a reason why acoustical instrument designs include physical hardship. Musicians need to achieve an extraordinarily sophisticated level of non-verbal communication. This functionality involves heavy cognitive requirements. From the point of view of *usability*, it is these cognitive requirements that dominate the physical design of the instrument. We should therefore approach the design of the physical Man-Instrument interface as a cognitive ergonomical problem. In the four cognitive ergonomical criteria for assessing the *usability* of systems defined by Shackel (1990), functionality is described by the concept *task*:

(1) **Learnability**: the amount of learning necessary to achieve tasks;
(2) **Ease of Use**: the efficiency and effectiveness with which one can achieve these tasks;
(3) **Flexibility**: the extent to which a system can adapt to new task and environment requirements;

Correspondence: Tamas Ungvary, Dept. of Speech, Music and Hearing, Royal Institute of Technology, Stockholm, Sweden. E-mail: ungvary@mdw.ac.at

(4) **Attitude**: the positive or negative attitude of the user towards the system.

When what is important is expert achievement of the task result, *learnability* and *attitude* requirements are inhibited by the *ease of use* and *flexibility* requirements. The *ease of use* and *flexibility* requirements, in their turn, are conflicting. According to Polfreman and Sapsford-Francis (1995), no single musical system is likely to fulfil individual task requirements. Computer music instruments should be customizable to musicians and uses: the *flexibility* criterion. However, continuous flexibility of musical instruments would require constant adaptation and memorization from the musician. The cognitive load of dealing with a constantly changing system would never allow a musician to internalize his instruments and achieve the efficiency and effectiveness of the *ease of use* criterion (Keele, 1968). It is these two conflicting issues, *flexibility* and *ease of use*, that we tried to address in the design of a computer music instrument.

Traditional musical instrument designs seem less affected by the above conflict. In traditional instruments, there is the requirement to have control and sound generation parts tightly coupled with a solid physical interface. Instead, computer interfaces have the unique ability to uncouple the input representation (physical manipulation) from the output representation (physical auditory and visual stimuli of a performance). The thought that control input would not be hindered by the practicalities and idiosyncrasies of the acoustical generators has indeed sprouted a wealth of new instrument designs, each with its own controllers and control mappings. New controllers using inexpensive pressure triggers, light beams, and motion detectors or video cameras have been developed, such as David Rokby's *Very Nervous System* (Cooper, 1995), and the *Virtual Stage Environment* (Lovell & Michell, 1995). Examples of the use of body sensors include *MIDI Dancer* (Coniglio), which analyzes body shape by measuring the angles of arm, leg, and hip joints; *BioMuse* (Knapp & Lufsted, 1990), which measures electrical voltages produced by muscle contraction, and a host of hand-based controllers. The above designs all accentuate the design of control input and musical output as two separate processes, each with their own degrees of freedom. Indeed, we believe the artistic gains made with this approach were considerable. It allowed indirection in the control of the sounding result by the performer, with generative computer processes adding to the richness of the music. It allowed radically new performance settings in ways not possible with traditional instruments. However, uncoupling also impaired the interaction between the musician and his musical instrument, because the freedom of information structure in uncoupled instruments easily results in a mismatch of information flow across human input–output modalities.

In the Hyperinstrument paradigm, Machover (1986) tried to overcome the above problems by combining the qualities of a tight coupling in traditional instruments with the qualities of a loose coupling in computer devices. Although we feel this was an important step towards recognizing the cognitive issues associated with the matching of input and output modalities, we felt that such augmentation of traditional instruments was, in many ways, a circumvention rather than a solution of the problem. Instead, we propose the new paradigm of *Cyberinstruments*. In a Cyberinstrument, the musician forms part of the computer music instrument, and vice versa: the instrument forms a part of the musician's body. Each element of the musician's body, the computer software, or the input/output hardware that participates in the process of making music forms part of an intricate, interconnected network of musical information streams. In this network, feedback processes at different levels provide a self-regulation or tuning of the Human-Instrument system towards optimal musical expression. Cyberinstruments essentially consist of computer input and output modules, with musical functionality modules in between. In a Cyberinstrument, each module is ordered such that modalities of human input and output are always mapped with musical functionality. This made our approach to the design of SensOrg, our first Cyber Instrument, a cybernetic one. The Human-SensOrg system is seen as a whole, a whole of constituting elements with optimized mappings, rather than as a set of simple input–output relationships (Clynes, 1970; Pressing, 1990). These elements include: rational and non-verbal intent, human actuator channel, input device, software functionality, output device and human perceptual channel, with information flowing across elements. The optimization of information feedback paths between elements played a key role in our design. The following sections will show how we tried to use the *structure* of traditional instruments, rather than the instruments themselves, in an attempt to find an optimal mapping of the above elements for different task situations, including composition, improvisation and performance.

DESIGN RATIONALE: COGNITIVE ISSUES & PHYSICAL DESIGN

In this section, we will discuss our design rationale for the physical interface of the SensOrg, identifying themes that recurred throughout the design cycle. First, we discuss the tasks to be performed with the instrument: composition, improvisation and performance. We categorized these tasks according to their timing constraints and subsequent level of abstraction. Then, we discuss how we used this categorization to match the flow of information between a musician and a set of physical transducers. Finally, we discuss the physical fit of the interface to body parts and to different users and uses.

Achieving Nonverbal Communication: Symbolic and Non-Verbal Task Modalities

We consider the ability of music to directly communicate non-verbalizable information via non-verbal channels (in particular, as a form of paralinguistic audio) to be its most important functionality. Behavioural sciences have only recently started to address the role of non-verbalizable information in human functioning, perhaps relating it to specific hemispheric activity in the cerebral cortex (Iaccino, 1993). Although it is unclear what the relation is between lower-level human emotion and higher-order associative intuition, these concepts for us define the essence of what is communicated in music. Although this has always been considered a speculative theory, Clynes (1970, 1982) suggested early-on that passionate states of emotion correlate with patterns of muscular tension and relaxation in such a way that the direction of causal connection is no longer clear. We believe the same pattern occurs in many forms of non-verbal expression, from facial expressions, sighs, body position, gestures, paralinguistic speech, to touching one another (Argyle, 1967). Somehow, sensory-motor activity seems to be associated with the same cognitive functions that process non-verbal information. The efficiency of sensory-motor processing might be a requirement for managing the complexity of non-symbolic information in the process of expressing it, as well as in receiving it (Edwards, 1979; Tenney, 1986). It is therefore that we consider sensory-motor tools essential in the process of musical expression. This does not mean that symbolic tools are not required. One *can* identify structural elements in non-verbal communication which can be characterized in a verbal or symbolic fashion. Although such elements are perhaps not of the same conceptual level as, for example, language, *order* in the form of rhythmical structures, compositional sequences, etc., introduces a form of redundancy. According to Wiener, this redundancy may be essential to the *understanding* of information (Wiener, 1948). We believe that in the design of this order, analytical tools can play an essential role. It is evident that in the communication of compositions, symbolic representations (such as scores) can be very effective.

We therefore regard the musical production cycle as a process in which non-verbal and symbolic task modalities complement each other, feeding back information from one to the other, and dominating at different stages of the process. It is in this light that we regard the traditional taxonomy of musical production modes: *composition*, *improvisation* and *performance* (Sloboda, 1985). To us, this classification characterizes the time-complexity constraints of verbalization and non-verbalization in asynchronous and synchronous communication situations (Dix, 1993). Composition maps onto asynchronous verbalization, while performance maps onto synchronous non-verbalization. Improvisation includes aspects of both. In the usability design of the SensOrg, the asynchronous verbalization constraint maps onto the *flexibility* criterion, and the synchronous non-verbalization constraint maps onto the *ease of use* criterion. In order to communicate the verbalization process to the instrument, we needed to be able to specify symbolic relations. In an asynchronous situation, this is done by means of the computer equivalent of pencil and paper: a graphical user interface with visual feedback. These symbolic relationships are then mapped onto a sensory-motor representation in the form of a completely flexible set of physical interaction devices arranged in space. By freezing the physical representation of the internal state of the system, the human sensory-motor system can then be trained to achieve the *efficiency* and *effectiveness* required for expressing non-verbal information in synchronous situations. However, in order to continue support of the verbalization modality in synchronous situations, the physical interaction devices retain their capability to modify the symbolic relationships inside the system throughout, e.g., an improvisation.

Ease of Use: Reducing Problems of Cognitive Load and Recall by Freezing Functionality

As discussed above, our task modalities essentially reflect two ways of dealing with time-complexity constraints of information: complexity as-is (non-verbal mode) and complexity structured (symbolic mode). We believe *cognitive overload* (as a semantical form of information overload) might occur due to a mismatch between time-complexity constraints of functional information and time-complexity constraints of modalities that process that information. Miller (1978) defines *information overload* as when channel capacity is insufficient to handle the information input. According to him, when the information input rate goes up, the output rate increases to a maximum and thereafter decreases, with the latter being a sign of overload. However, in our view, channel capacity depends on the interaction between the semantics of information *and* its rate (Schroder et al., 1967, see Hedberg, 1981). This yields a measure of *cognitive load* in the Wiener (1948) sense, rather than *information load* in the Shannon & Weaver (1959) sense (see Sveiby, 1998 for a discussion). Addressing problems of cognitive overload thus requires more than a simple reduction of information flow per channel by decreasing rate of information or by using multiple channels. It requires more than the selection of a channel on the basis of the load of other channels. It requires representing information in such a way that the processing of the meaning of that information is most efficient. Wiener suggests a negative relationship between entropy of meaning and entropy of information signal (Wiener, 1948). If this is correct, the usefulness of symbolic representations may be related to their ability to convey highly entropic semantics using little information. If we, however, assume a positive relationship between entropy of meaning and processing time required, we immediately see the benefit of non-symbolic representations. Thus, in designing a representation, the rate and entropy of the semantics that need to be communicated by the underlying function are important factors. This implies that a good mapping of the time-complexity constraints of a situation might ease cognitive load. Since in a cybernetic approach, we should regard human input/output as a feedback process, this mapping should not only occur in the design of system output, but also in the design of system input.

In an attempt to address some of the above issues in the hardware design, we collected a comprehensive set of input–output devices, carefully matching them onto the functionality of the system by identifying input/output channels associated with human processing of the information required by that functionality. We used visual feedback for the more asynchronous symbolic functions; and auditory, tactile-kinesthetic feedback for the more synchronous non-verbal functions. We selected input devices in a similar fashion: buttons, faders, touchscreen and mouse for the more asynchronous symbolic functions; and buttons, faders, trackballs and touchsensors for the more synchronous non-verbal functions, in that order. For a more complete discussion of these mappings, see (Vertegaal & Ungvary, 1996). This mapping of I/O devices with software functionality also addressed the highly related issue of recall. We tried to introduce as much explicit knowledge into the real world as possible, attempting to reduce the requirements for knowledge in the head (Zhang & Norman, 1994). Essentially, we tried to externally represent the state of internal software functionality as much as possible. All I/O devices can be frozen into a unique spatial arrangement. Each device is coded by color, shape, orientation within groupings and textual information. For example, we put the touchscreen onto a picture of a Kandinsky painting (Fig. 1). By association of the position of virtual buttons with the arrangement of graphics on the picture, we tried to improve memorization of their function.

Flexibility: Adaptation to Individuals and Task Situations by Malleable Functionality

In the design of the SensOrg, we wanted to combine the qualities of a tight coupling with the qualities of a

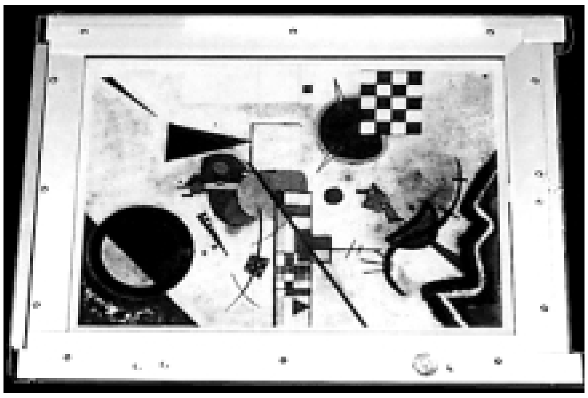

Fig. 1. The Image-in-Kit with Kandinsky painting.

loose coupling. As we have seen, in a loose coupling there is indirection in control, in a tight coupling there is not. The field of tension between tight and loose coupling is reflected in the conflicting requirements of the *ease of use* and *flexibility* criteria. We will now discuss how we made the system flexible, so that it could be adapted to different individuals and task situations such as compositional requirements. We could only choose to reflect the state of internal software functionality in the external devices if we also reflected the malleability of software functionality in the external devices. If the software functionality changes, the external devices should change and vice versa. If the software functionality stays the same, the external devices stay the same, as long as it is satisfactory. We did this by taking a modular approach to both software functionality and hardware devices. The software modules can be configured in an asynchronous, symbolic fashion by means of the graphical user interface. They can be driven in a synchronous, non-verbal fashion by manipulating the corresponding hardware modules. Similarly, hardware modules can be configured in a more asynchronous symbolic fashion by mapping them onto a software module, labeling them with a concept describing that functionality (with the device type being a label by itself), coloring them, positioning them freely within groups, and orienting groups freely within the instrument. They can be configured in a more synchronous, non-verbal fashion by selecting predefined configurations of software mappings using predefined buttons.

Apart from cognitive constraints, an important criterion for organizing hardware modules is the physical fit with human body parts. This is an extremely complex issue, where there are many individual differences. In addition, the task modality as related to musical functionality plays a role in this. Basically, the SensOrg hardware is so freely configurable, that it is almost totally adaptable to circumstances. It can accommodate individuals with special needs, including physical impairments. However, there are some basic functional and physical constraints which can be generalized across situations. The SensOrg is divided into two parts: one for the dominant hand, and one for the non-dominant hand. The dominant hand exercises mostly the more synchronous non-verbal functions, while the non-dominant hand exercises mostly the more asynchronous symbolic functions. This is because of the time-complexity constraints of information flow in these modalities.

In the center of the dominant hand is the *FingerprintR*, a 3D sensor which conveys states of tension as exerted by subtle changes in force (see Fig. 2). This is the most important device for the asynchronous non-verbal modality. For a more detailed discussion of this issue, see (Vertegaal & Ungvary, 1995). In order to meet the haptic feedback requirements of this process, the FingerprintR knob is concavely shaped, following the form of the finger with which it is played. This knob can be replaced to account for individual differences. In order to reflect the non-verbal intent in the muscle tension of the player, it is vital the upper-torso is in a relaxed position, while not relinquishing the ability to exert force. Since the SensOrg does not include devices operated by breathing force, the instrumentalist is typically seated like double bass players in an orchestra, so that his hand can be placed on the FingerprintR without necessarily exerting weight. Since the thumb opposes the other fingers, and can move more or less independently, the thumb of the dominant hand is used to control the more synchronous non-verbal button functions. In order to minimize the path and effort needed to press these buttons, they are placed below the FingerprintR knob. The area covered by the non-dominant hand is much larger. In the center of this area are groupings of faders and buttons. These are the most important devices for the

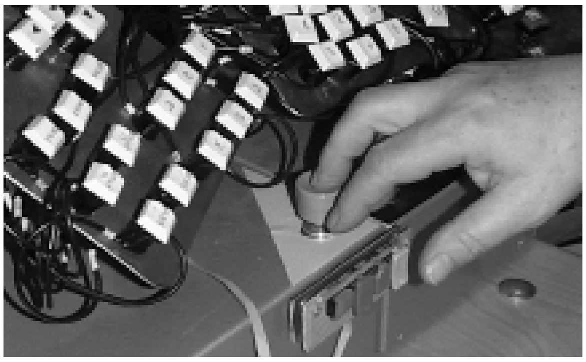

Fig. 2. The FingerprintR knob.

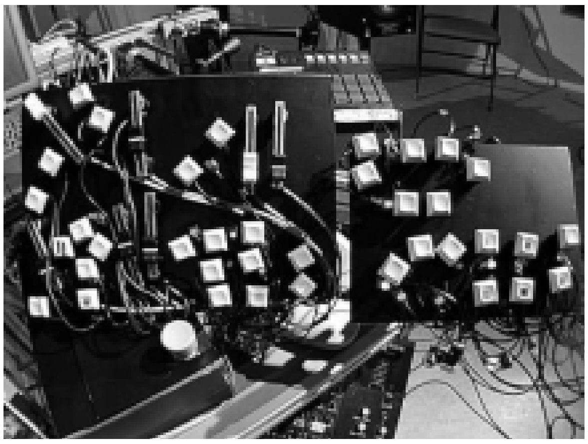

Fig. 3. Flexipads with magnetic buttons and faders.

more asynchronous symbolic functions. Button and fader modules stick to a position on a metal pad by means of small magnets. These pads (called *Flexipads*) can be positioned and oriented freely in space, and button and fader modules can be freely positioned on the pad (see Fig. 3). Fader modules can be grouped so that they can be operated simultaneously with one hand gesture. Fader modules and button arrangements can be fitted to the hand by putting the hand onto a selection of devices, and then moulding the devices around the physical contour of the hand.

AN OVERVIEW OF THE PHYSICAL INTERFACE OF SENSORG

In Figure 4 we see how the discussed hardware modules fit together in the current implementation of the SensOrg Cyberinstrument. All modules are mounted on gimbals attached to a rack with adjustable metal arms. This effectively allows them to be placed at any position or orientation. On the left, we see an arrangement of several Flexipads. On the Flexipads, modular structures of faders and buttons are shown. In the middle of the figure, we see the right hand subsystem with two FingerprintR knobs in the middle. Around these, two smaller Flexipads are arranged with real-time functionality. The above modules are the main physical ingredients of the SensOrg. Each hardware module is connected to software functions

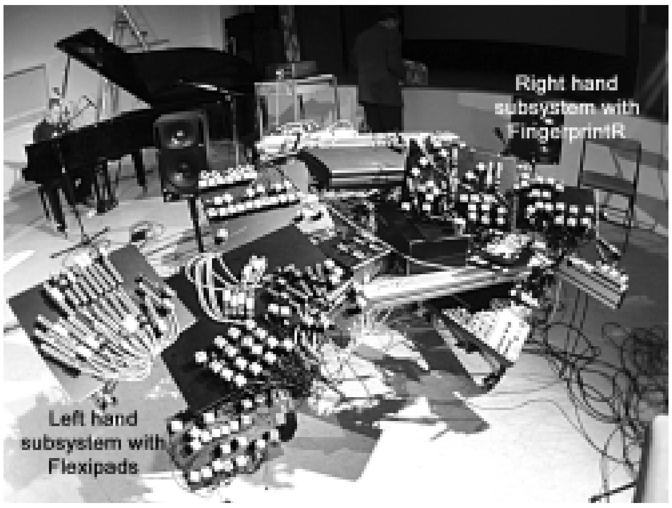

Fig. 4. The SensOrg. For more information, see our website: http://www.speech.kth.se/kacor/sensorg/main.htm

running on a PowerMac computer, which will be discussed in the next section.

IGMA: AN OVERVIEW OF THE SOFTWARE INTERFACE OF SENSORG

The mapping of the input control data onto the musical parameter space is provided by means of the IGMA (Interactive Gesture Mapping) system, which is implemented in Max (Puckette and Zicarelli, 1990). This software front-end connects hardware modules with musical functions that provide real-time high-level control of composition or sound synthesis algorithms. It also allows the output of such algorithms to be mapped to, e.g., a MIDI sound synthesizer. IGMA is not an application, not a finished software product, but a framework. It is a set of software entities tuned to collaborate with each other and consists of both generic and specific software objects such as Units, Modules, Sub-modules and Databases with integrated Data Editors. The current set of software entities was designed to serve both improvisation, interpretation and composition modalities, providing functionality such as:

- Flexible use of any kind of external or internal input–output device;
- Iterative performance of repetitive tasks;
- Modeless switching between task modalities, from performance to composition and vice versa, automatically preserving the state of all variables;
- Efficient editing of data objects;
- The swift propagation of changes throughout a network of objects.

IGMA Units

A Unit is a self-contained entity with specified musical behavior that may run independently or in parallel and may be controlled separately or as a group. Units can also function as standalone applications. The three basic Units we implemented are STREAM, SINO and LONO. We will explain each Unit in detail below. Each Unit models a different attitude toward the real-time triggering of tones. The SINO and LONO Units came into existence after a jam session with musicians who played acoustic instruments such as trombone, saxophone, electric guitar, and tabla. During this session we found that the STREAM Unit, which creates complex clouds of sounds and textures, was not sufficient. One could not play one single note at the right moment or adequately join the rhythmical sections played. With the other two Units added, musical performance of SensOrg was dramatically improved.

The STREAM Unit
The underlying musical and gestural idea of the STREAM Unit is to generate a progression of tones based on the three-dimensional parameter stream of the FingerprintR. This parameter stream is connected to three main musical parameters of the STREAM Unit: pitch, density of tones, and loudness. More symbolic asynchronous input modules (sliders, buttons etc.) control the setting ranges for the pitch and density parameters of the Unit. They also control various filters (speed sampling frequency of incoming data, scale) as well as secondary data generators (such as chord generators).

The SINO Unit
The SINO-unit offers a distinct control of the triggering of individual tones with fast attack and decay characteristics. The tone is triggered only at the peak of the stream of amplitude values from the input device, typically a FingerprintR.

The LONO Unit
The fundamental demand to dynamically alter the parameters (timbre, pitch and loudness) of a sustaining tone after its triggering has been the starting point for the design of the LONO-Unit. Tones are triggered in a similar way as in the SINO Unit, but they sustain. The controller information that triggered the tone is dynamically remapped to control the sustain and release characteristics of the tone, for example, by using the ISEE system (Vertegaal & Bonis, 1994).

The SEQINT Unit
With the SEQINT Unit up to 32 independent sequencers can be controlled and coordinated. The basis of this Unit is the Max SEQ-object (Puckette & Zicarelli, 1990). The user can control playback speed,

the percentage of triggered tones in a sequence (100% meaning that all notes are triggered), the volume, transposition, scale pattern and voice. One may also choose from a menu of preloaded sequences in order to avoid delays caused by file I/O.

The 8 × 8ASR Unit

The 8 × 8ASR Unit was realized to treat complex layers of sound and focuses on the use of the sound processing functions in the Ensoniq ASR-10 Sampler. This Unit lets one loop up to eight complex sound clusters with a maximum of eight sounds each. Since the ASR-10 can record sounds previously created, recorded or treated, the output of this Unit can become very complex indeed. Loop position, frequency and volume of the sound clusters can be controlled individually. Individual envelope duration may also be defined for all parameters in order to achieve a smooth changing sound texture. Parameter values may be set using a mouse or one of the other input modules.

Other IGMA Modules

Input Mapping Modules

The input mapping modules provide the routing mechanism for all types of data originating either from external input devices, from Units, from recorders, or from other applications connected via OMS inter-application communication. The input signals may serve as:

- Generators;
- Modifiers;
- Selectors;
- Activators.

In certain cases these functions may be combined. The function typically defines the type of input device to be used. For example, we do not use a comparatively asynchronous, symbolically used input device such as the mouse to generate data, but to select or activate. The function also defines our expectations. In case of generation we definitely expect a signal to be of continuous character, a selector however conveys a discrete event. This contrasting behavior of the above functions acts to limit the freedom with which one can link input signals as to provide a framework for the mapping of information flow.

Output Management Modules

The output management modules translate the output of Units into a specific command language for different (hardware) sound synthesizers. The information is stored partly inside a database that provides the routing information for the output mapping, and partly inside Max objects that communicate with the external devices directly. The database contains knowledge about:

- The address and the channel of a certain device or external computer system;
- The type of device;
- Specific system specifications;
- The control structure of the sounds stored.

This module makes the user deal with sounds rather than with the devices that generate these sounds. During a performance one only needs to select the label of a sound patch, without being concerned about which devices it plays on, or what synthesis model is used in the process.

Currently, the following sound generation devices are supported:

- Ensoniq ASR-10 (sampler)
- Proteus 3XR (synthesizer)
- Proteus Proformance (synthesizer)
- Korg Wavestation SR (synthesizer)
- Yamaha TX81Z (synthesizer)
- Yamaha TG77 (synthesizer)
- The ISEE System (MIDI timbre space software, see Vertegaal & Bonis, 1994)

Function Control Modules

Function Control Modules are modules designed to provide general control over internal functions (e.g., the loading of mappings into several Units). Two of them offer very significant structural control: the Control Data Recorder and the Snapshot Module.

The Control Data Recorder (CDR) makes it possible to *Record, Merge*, *Overdub*, *Replace*, *Remove*, *Modulate*, *Add* or *Insert* a stream of incoming MIDI controller messages. The above operations have a universal character and support an iterative compositional process. Any data from the Input Management Module or generated inside a Unit is automatically recorded inside the CDR. The CDR is actually a multi-track recorder of control shapes. Because of current limitations in our software platform (*Max*), time-domain operations (Remove, Insert, Add) must be done using an external application, such as MidiGraphy, that supports the graphical manipulation of control shapes.

The Snapshot module instantly stores all of the software parameter values that define the internal state of the system. Musical output as it occurred at the moment of storage may be obtained by retrieving a snapshot. A button on the instrument is typically used for this relatively asynchronous, symbolic function.

IGMA Sub-Modules

Despite their separate interpretation of incoming data, different Units sometimes need a common type of data treatment. For this purpose, we designed general sub-modules that can be applied to a specific Unit to alter its behavior. These Sub-modules typically modify the data coming into the Unit. For example, the Scale-Pattern Sub-module changes MIDI note values to match a predefined scale, chord or interval pattern. Sub-modules typically rely on associated databases with dedicated editors.

Metronom – Speed Control Sub-Module
A Metronom Sub-module is provided within both the SINO and STREAM Units. The Metronom generates a regular stream of time beats. Incoming data may be timed irregularly. This feature quantifies the incoming beats to fit into a regular timing as to produce a regular rhythm. The speed of the regular beats may be controlled by external input devices. Instead of a regular rhythm, the Metronom data interpretation mechanism may be used to trigger tones according to predefined rhythms, as will be explained below.

Scale Pattern Database and Editor
Complex parameter settings, such as scales with their transpositions and intonations, chord-structures and rhythmical patterns may be prepared before a performance using dedicated editors. These patterns are stored in a dedicated database. Stored parameter settings may be retrieved during a performance in real-time. To support the Scale Pattern object, which changes MIDI note values to match a predefined scale, chord or interval pattern, one first defines a database. When adding a new item to the database one may choose between three input alternatives. To select the pitch pattern to be applied to all octaves repeatedly, one uses the one-octave editor. Given that we have twelve fundamentals on which patterns may be applied, the number of possible patterns may be large. To counter this problem, we devised a simple grouping mechanism. Small databases, called Custom Lists, may be prepared with different predefined fundamentals for each item. These Custom Lists may be retrieved and the predefined patterns may be triggered in real-time. During the preparation phase, a maximum of six custom lists of any length may be prepared to contain scales, intervals and chords. Attributes of the sub-module then allow us to step through the items in a forward direction, to step backwards, to jump to the start of a list or to choose random items from a list.

Rhythm Pattern Database and Editor
Instead of a steady stream of regular beats, predefined rhythm loops may be produced by the Metronome Sub-Module. This is done by filtering of regular beats. A forty cell long Rhythm Pattern Editor is provided to define a rhythmical pattern. Patterns must be assigned to Custom Lists. One of the Custom Lists must be selected as active. Its items may be activated by moving the list index in real time. Similar to the Scale Editor, the user creates a menu with an unlimited number of different patterns, either by using a 32 cell graphical Beat Pattern Editor or by recording the patterns into the editor in real-time.

IGMA Software Feedback Displays

Each IGMA Unit may contain several feedback displays which have to be adapted to the visual feedback required by the composer or performer. A visual feedback window displays the actions of the performer, allowing him to see whether input signals have been received and treated correctly. A good design is characterized by a high level of confidence offered while the system is running. Another important objective is to establish the coherence between the actions of the performer and the resulting alterations of the visual elements on the screen. In order to reduce the mental effort that is needed to interpret the visual information, spatial and conceptual ordering on the screen has to correlate with that of the hardware modules used in the system. Ideally, the positioning of the visual objects on the screen should be analog to the arrangement of the physical transducers (Kieslinger & Ungvary, 1997).

CONCLUSIONS

In this paper, we presented the SensOrg, a musical Cyberinstrument designed as a modular assembly of input/output devices and musical generator software, mapped and arranged according to functional characteristics of the Human-Instrument system. We have shown how structuring access to, and manipulation of information according to human information processing capabilities are essential in designing instruments for use in composition, improvisation and performance task situations. We regard this musical production cycle as a process in which non-verbal and symbolic task modalities complement each other, feeding back information from one to the other, and dominating at different stages of the process. We identified how these task modalities may be mapped onto the time-complexity constraints of a situated function: asynchronous verbalization vs. synchronous non-verbalization. By matching time-complexity constraints of musical functions, transducers, human I/O channels and body parts, we carved functional mappings between the more asynchronous symbolic elements on the one hand, and the more synchronous non-verbal elements on the other. To allow these mappings to be adaptable to individuals and situations, hardware as well as software configurations were designed to be totally flexible. To allow mappings to be effective, however, physical interface devices can be frozen in any position or orientation.

ACKNOWLEDGEMENTS

We would like to thank the Bank of Sweden Tercentenary Foundation, the Royal Institute of Technology, the Knut & Alice Wallenberg Foundation, Mario Zlatev and Michael Kieslinger.

REFERENCES

Argyle, M. (1967). *The psychology of interpersonal behaviour*. London: Penguin Books.

Clynes, M. (1970). Toward a view of man. In M. Clynes, & J. Milsum (Eds.), *Biomedical engineering systems* (pp. 272–358). New York: McGraw-Hill.

Clynes, M. (1982). *Music, mind and brain, the neuropsychology of music*. Plenum Press.

Clynes, M. (1980). The communication of emotion; theory of sentics. In R. Plutchik & H. Kellerman (Eds.), *Emotion, theory, research and experience. Volume 1: Theories of emotion*. New York: Academic Press.

Coniglio, M. MidiDancer. http://www.troikaranch.org/mididancer.html

Cooper, D. (1995). Very nervous system. *Wired Magazine*, *3*(3), 134–170.

Dix, A., Finlay, J., Abowd, G. & Beale, R. (1993). *Human–computer interaction*. London: Prentice-Hall.

Edwards, B. (1979). *Drawing on the right side of the brain*. Los Angeles: J.P. Tarcher.

Iaccino, J. (1993). *Left brain–right brain differences*. Lawrence Erlbaum: Hillsdale NJ.

Keele, S. W. (1968). Movement control in skilled motor performance. *Psychological Bulletin, 70*, 387–402.

Kieslinger, M. & Ungvary, T. (1997). Aspects of visualizing information for a realtime hypermedia musical environment. In V. W. S. Chow (Ed.), *Multimedia technology and applications* (pp. 172–180). Berlin, London, New York: Springer-Verlag.

Knapp, B. & Lusted, H. (1990). A bioelectrical controller for computer music applications. *Computer Music Journal*, *14*(1), 42.

Lovell, R. & Mitchell, J. D. (1995). Using human movement to control activities in theatrical environments. In N. Zahler (Ed.), *Proceedings for the Fifth Biennial Symposium for Arts and Technology*. New London: Connecticut College.

Machover, T. (1986–1992). Classic Hyperinstruments. A Composer's Approach to the Evolution of Intelligent Musical Instruments. MIT Media Lab. http://parallel.park.org/Events/BrainOpera/text-site/growth/tech/classic.html

Miller, J. (1978). *Living Systems*. New York: McGraw-Hill.

Polfreman, R. & Sapsford-Francis, J. (1995). A human-factors approach to computer music systems user-interface design. In *Proceedings of ICMC'95* (pp. 381–384). Banff, Canada: ICMA.

Pressing, J. (1990). Cybernetic issues in interactive performance systems. *Computer Music Journal 14*(1), 12–25.

Puckette, M. & Zicarelli, D. (1990). MAX – An Interactive Graphic Programming Environment. Opcode Systems, Menlo Park, CA.

Shackel, B. (1990). Human factors and usability. In J. Preece & L. Keller (Eds.), *Human–computer interaction: Selected readings*. New Jersey: Prentice Hall.

Shannon, C. E. & Weaver, W. (1959). *The Mathematical Theory of Communication*. Uni. of Illinois Press.

Sloboda, J. (1985). *The musical mind: The cognitive psychology of music*. Oxford: Oxford University Press.

Sveiby, K. (1998). What is Information? Sveiby Knowledge Management, Australia. http://www.sveiby.com.au/information.html

Tenney, J. (1986). *META/HODOS and META Meta/Hodos*. Hanover: Frog Peak Music.

Vertegaal, R. & Bonis, E. (1994). ISEE: An intuitive sound editing environment. *Computer Music Journal*, *18*(2).

Vertegaal, R. & Ungvary, T. (1995). The sentograph: Input devices and the communication of bodily expression. In *Proceedings of ICMC'95*. Banff, Canada: ICMA.

Vertegaal, R. & Ungvary, T. (1996). Towards a musician's cockpit: Transducers, feedback and musical function. In *Proceedings of ICMC'96*. Hong Kong: ICMA.

Waiswicz, M. (1985). THE HANDS: A set of remote MIDI controllers. In B. Truax (Ed.), *Proceedings of ICMC'85*. San Francisco: ICMA.

Wiener, N. (1948). *Cybernetics*. Cambridge, US: MIT Technology Press.

Zhang, J. & Norman N. (1994). Representations in distributed cognitive tasks. *Cognitive Science*, *18*, 87–122.