Journal of New Music Research, 28 (1999), No. 2, pp. 90–104 0929-8215/99/2802-090$15.00

From Symbol to Semiotic: Representation, Signification, and the Composition of Music Interaction

Michael Hamman

National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA

ABSTRACT

This paper examines notions of interaction in order to synthesize an approach to the use of computers in the arts which respects the fact that, oftentimes, creative work is facilitated by task environments in which "surprises" can happen. Typically, interface design concerns the rendering of an interaction such that it requires minimal cognitive engagement with the task in question, relying heavily on historically and culturally determined patterns of behavior and cognition. Ill-structured problems (like music composition), however, benefit when the interface presents concepts and interactions in ways that are novel. Computers can be understood as tools for the projection of such an interface when they are conceived as generators of *semiotic* rather than *symbolic* ordering frameworks.

INTRODUCTION

Computer music is concerned with the design and composition of acoustical and musical representations made with a computer. In this context, representations articulate domains of interaction into which one might enter while designing musical processes and structures. An aggregate of representations is projected within a collection of objects and operating principles that define an *interface*. Often, the interface is structured around a static set of interactions which project a similarly static set of representations: the machine becomes a device which the human 'uses' in order to accomplish well-structured tasks. In this case, the interface effectively conceals the potential underlying complexity of a system by situating interaction according to familiar and therefore cognitively redundant patterns of action and observation.

Within the context of facilitating task domains (e.g., driving a car, using a word processor, etc.), such a notion of interaction makes sense: we want to be able to leverage our history of experiences (both cultural and personal) in order to reduce the cognitive strain that might otherwise be involved. In problem-posing task domains, however, like music composition, such a notion of interaction can have an inhibitory effect. One way to encourage creative problem-solving is to allow the human actor to somehow construct the very representations and interactions in the context of which a problem might be framed (Truax, 1976, 1986;

Correspondence: Michael Hamman, National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, 705 W. Nevada St. #4, Urbana, IL 61801, USA. E-mail: m-hamman@uiuc.edu

Manuscript submitted: 4 May 1998

Tipei, 1990). By this means, a human can begin to free her/himself from historical and cultural patterns of thought which often block fresh insight (Smith, 1995; Luchens et al., 1950).

In this paper, I attempt to decouple the notion of interaction from its mechanistic orientation in order to understand the computer as a tool for composing musical interaction and representations. The paper has two parts. In the first part, the notion of *representation* and *interaction* are interrogated. Representations orient a cognitive system toward the appropriation of a particular model of experience. Typically, however, representations encapsulate historically and culturally bound epistemologies by which human performance is constrained. Computers can be understood as tools for *problematising* that encapsulation; that is for dislodging the historical references according to which representations are embodied and appropriated. The computer becomes a *semiotic,* rather than a *symbolic* processor, when it assists in the postulation of possible, though as-yet non-existent real-world scenarios, as opposed to referencing already existing ones.

In the second part, I posit a descriptive framework which attempts to foreground this epistemological aspect of interaction. Following up on the discussion given in the first part of the paper, two different approaches are outlined: one approach exercises *symbolic* and *denotative* notions of interaction, while the other approach exercises *semiotic* and *connotative* notions of interaction. The former projects a task environment through the production of metaphors by which already existent cultural artifacts might be synthesized; the latter projects a task environment through the hypothesis of design criteria by which as-yet non existent cultural artifacts might be posited. This distinction is examined within the context of computer music practice and composition theory. Two computer music systems are briefly considered, each of which, in its own way, conceives the computer as a tool for the problematisation of interaction — a tool, that is, with semiotic comportment.

REPRESENTATION, INTERACTION, AND SIGNIFICATION

Representation

An interaction is an exchange of energy between two differentiated existents, be they imagined or physically realized. An "interface" defines the mechanisms — physical, conceptual, and cultural — by which a set of interactions appropriate to a particular cognitive domain is engendered. An interface can be as simple as a door knob or as complex as an airplane cockpit, or it can delineate a conceptual framework such as a text or a score.

A representation constitutes the agency through which an interface is embodied by orienting a particular way of conceiving and understanding a signal. A signal is a perturbation within some phenomenal domain: it can be a pattern of air pressure changes, an analog voltage, a collection of data, and so on. Representations determine the descriptions by which a signal might be acted upon or otherwise used, thus orienting possible modes of interaction with respect to that signal. To say, for instance, that a signal has this or that amplitude and frequency characteristics is to give a description which distinguishes a particular way of representing and interacting with it. An oscilloscope projects a set of representations according to which electrical signals might be so described.

When a particular representational framework becomes the prevailing means for producing and observing a signal, cultural practice often collapses the identity of the signal with that representational framework. With this collapse, signals come to us already prepackaged according to a particular way of describing them. However, through acculturation and education, we fail to notice this and instead treat such a signal as though it were in fact "raw data" (Feyerabend, 1981: 52—54). This is the case even (or perhaps *particularly*) when the traces of those representational frameworks are most acutely present within the signal. Through habit and acculturation, we become unable to observe the descriptive and representational apparatus by which a signal obtains its coherence: signal and representation are subsumed as one.

Interaction as 'circumspective being'

The possibility of the equation of data and representation motivates standard approaches to human/computer interaction. Following principles of human factors engineering, human/computer interaction understands the computer interface as structured around the designs, goals, and plans of a projected human actor (Norman, 1986). The input and output mechanisms of the computer are structured in order to map the task domain referenced by those plans and goals. The human is thus able to focus on the tasks at hand, without having to stop and figure out what the computer output 'means' or how it works.

One way to facilitate such a mode of interaction, is through the use of 'interface metaphors' (Laurel, 1993; Erikson, 1990). Interface metaphors work by relating tasks associated with the computer to task domains outside of that which would otherwise encompass use of the computer. For instance, the modern word-processor references tasks associated with using a typewriter. As a result, anyone who knows how to type will already know much of what there is to know in using a word-processor.

Interface metaphors constrain human interactions according to an embodied set of cognitive and epistemological principles whose basis lies within a history of experiences (both personal and cultural). They simplify human/computer interaction by orienting the human toward the appropriation of familiar conceptual frameworks. An interface metaphor is, in this sense, treated as a *signifier* (a naming or 'indexical' agent) that denotes, in some form or another, real-world principles of which a human actor has easily accessible historical experiences.

Through repeated application, interface metaphors orient how we think about the computer and its use. Through training and exposure, we learn to think of it largely as a *denotative* apparatus. Consequently, we are directed toward the appropriation of an *expected* performance; a situation in which subjectivity is neutralized under the imperative of a signified task. The human becomes a "user" and the computer becomes a "device." Subjectivity therefore comes to be fixed according to a static set of well-known and well-rehearsed tasks and procedures.

This is consistent with the goal-oriented approach to interface design taken by Donald Norman and others (Norman, 1988). An interface – be it a door or a software system – should leverage personal and cultural history in order to simplify interaction. When well-designed, the interface should tell us, by reminding us of our history of experience, how it works. We shouldn't have to think about how to use a door knob, for instance: it should tell us by reminding us, through its shape, material, and form, how it is to be used.

At precisely the moment when an interface becomes sensible and useful, however, the shapes, materials, and structures which constitute its physical and epistemological frame, cease to exist in themselves – they effectively disappear from within our attentive frame, absorbed in the functioning of the projected interface. Things become pieces of 'equipment' with which we get something done (Heidegger, 1962; Dreyfus, 1993).

Heidegger referred to such a mode of being as *circumspective*: that way of being which is elicited by familiar tasks and task environments (Heidegger, 1962). The skilled carpenter, for instance, doesn't have to think about what s/he is doing when s/he is hammering a nail: both hammer and hammerer disappear in the immediacy of the task of *hammering*. Similarly, when writing a document using a word processor, if one is skilled in the use of that word processing software, the functional aspects of the software disappear in the interaction. One no longer thinks about individual finger movements, or questions the appearance of text on the screen.

In such experience, both subject and object are absorbed into the apparent *immediacy* of the task and the actions required for its execution. Meanwhile, the human is directed toward the appropriation of an *expected* performance; an arrangement by which her/his subjectivity is neutralized under the imperative of the signified task. Only when something "breaks down" can the objects which constitute the circumspective unfolding of the interaction begin to appear *as things*.

Engineering a breakdown in circumspective interaction

Sometimes we would like to explicitly interrupt such a circumspective passage through an interaction – to retard the rate at which human subjectivity is consumed by the signifying imperative of the interface. One way to do this is to direct the interface toward the disruption of an expected performance. No longer a vehicle for replicating historical methodologies, the interface instead orients a *hypothesized* domain of interaction, thus engaging the generation of an *un*expected performance. Understood as such, interaction projects a notion of the 'subject' that is mutable and emergent, rather than fixed and transcendent. As emergent, a subject arises in the moment at which something unfamiliar, or foreign, appears, and, in its labor over the comprehension and synthesis of that something, projects itself toward it (Adorno, 1973: 21–22). In a sense, the moment of the appearance of an object represents the very commencement of the subject – its beginning, that is, as an activated and activat*ing* agent, as opposed to a static, *a priori*, existent. At this moment, the subject recognizes its labor over the comprehension of the object as a self-constituting process by which it quite literally becomes other to itself.

To create such an environment is to constitute the appearance of things as foreign objects – to cause the appearance of the object of interaction *as Other*. By situating the object as Other – as something which confronts the subject with its very Otherness – the subject itself comes into crisis. This moment of crises manifests an ontological "breakdown" in the occurring of an interaction. A breakdown constitutes the interruption of an otherwise *circumspective* way of being. The hammer breaks or the software crashes and the "user" – previously absorbed in the immediacy of the task at hand – suddenly appears as a subjectivity correlated to the breakdown. Such a moment occasions the sudden appearance not only of the subject, but of a newly contextualized object (Dreyfus, 1991: 70–72).

Computation and signification

As a tool for the construction of interfaces, the computer enables the design of mutable constructs by which such a breakdown might be engineered. However, the tendency within the commercial software industry has been to mitigate this capability and to understand the computer dualistically; constrained according to limitations set forth by the thing which it represents. The search for the proper representation is motivated by performative criteria: what will best assist a human in comprehending and synthesizing the relevance of her/his actions *vis* the effects generated. The underlying metaphysical assumption is that there exists a "real" world which the computer system references in the domain of interactions that it projects (Anderson, 1997). As a consequence of such design criteria, computers become 'virtual' machines which project representations based on historical imperatives. In this regard, they effect *symbolic* modes of representation.

A *symbol* is a special kind of sign: one that is "based on conventional relations between signifier and signified" (Anderson, 1997: 5). For example, smoke can be understood as a "symbol" for fire and the eagle a "symbol" of heroism. As Julia Kristeva notes, "in the case of the symbol the signified object is represented by the signifying unit through a restrictive function-relation" (Kristeva, 1986: 64). This function-relation defines an epistemological framework that is both immutable and non-porous: it is, effectively, a *black box*. It orients an ontology in which interaction is circumscribed by a history of use, and thus prone to that circumspective mode of being in which – as was the case with the carpenter and hammer – both subject and object disappear into the apparent immediacy of the interaction.

Given such an understanding of the computer, the methods and data which define a task environment are conceived as statements which reference a domain. However, within the context of problem-posing and problem-solving task domains, such as music composition, there often is no pre-existing thing to reference. In such a case, strongly referential interfaces can block creative activity. One desires an environment in which that which is *as-yet* unimagined might be formulated and realizable through the projection of an unfamiliar pattern of interactions. For this, one requires a signifying system that is *connotative* rather than *denotative* – one that is *semiotic* rather than *symbolic.*

Through semiotic interaction, historically determined – and therefore *circumspective* – modes of interaction are thrown into crisis. Subject and object are foregrounded, while process itself becomes primary. Interaction thus becomes a context for *hypothesizing* a domain of actions and descriptions, rather than a means for the simulation of already existing ones. As a consequence of this shift in emphasis, representations move from being immutable referencing agents – through which particular kinds of objects and artifacts are referenced and understood – to becoming *orienting* agents wherein the actions by which such objects and artifacts are produced are themselves hypothesized and thrown into question. While the symbolic is concerned with representations of objects through a fixed "function-relation", the semiotic is concerned with the "playful" aspect of signification: the state of the signifier prior to its becoming fixed to some particular meaning or social order. "Without becoming a system," writes Kristeva, "the site of semiotics, where models and theories are developed, is a place of dispute and self-questioning, a 'circle' that remains open" (Kristeva, 1986: 78).

As semiotic, the computer becomes an instrument for epistemological "play." It becomes a tool for constructing the representations with which as yet un-imagined worlds might be realized – an environment in which the outcome of an interaction cannot be determined beforehand. One becomes empowered to construct the conditions under which a signifier (the name of a thing) signifies and denotes things in a world. As an agent in this empowerment, the computer is no longer bound to its otherwise *denotative* imperative; rather it enables the construction of representations and of domains of interaction within which those representations are engendered.

Understood as semiotic, the computer becomes a framework for the explicit problematisation of interaction; a means for the interruption of circumspective being. The computer comes to be conceived as a context for reclaiming the 'thingly' nature of equipment, for reconstituting the object as Other.

TOWARD THE COMPOSITION OF INTERACTION

Modeling a domain of interaction

At perhaps the lowest level, an interface is embodied through the engagement of a *mechanism.* A mechanism encapsulates the design principle by which material is shaped and structured in order to generate a particular outcome, given an action. The door knob, for instance, has a mechanism which maps a specific action or set of actions to an outcome: passage through the door.

An interaction involves two overlapping dimensions of human performance:

– Action
– Description

An action is that which, when coupled with a mechanism, can effect a change within an environment. An action can be of a symbolic nature (e.g., program code, data) or it can be kinesthetic (e.g., physical gestures). In all cases an action is made in order to alter the state of a mechanism, and thus its outcome.

The effect of an outcome is discerned when a description is given. Through description, a mapping between action and outcome is hypothesized. A description need not necessarily be something which we say aloud – rather it becomes, most often, an internalized framework that determines our actions and observations regarding our use of some mechanism.

The relation between action and outcome can be understood as a feedback circuit such as that depicted in Figure 1. As depicted, an action is made in relation to a mechanism. A change in state within the mechanism generates an 'outcome.' An outcome is correlated to a potential action through the generation of a description. As recent theories of embodiment have shown, however, there is not a hard and fast separation between description and action. Rather, they constitute overlapping aspects of a single cognitive frame (Varela et al., 1996). Description comes to be 'embodied' within the action, and vice versa.

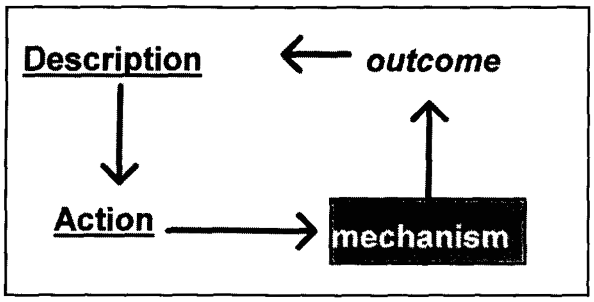

Fig. 1.

A description encapsulates a history, both cultural and personal. As such, it is epistemologically framed. As so framed, a description constrains the manner in which a human might conceive the outcome of a mechanism. If, for instance, the mechanism is unfamiliar, one attempts to come up with a description by which that mechanism makes sense. In other words, an attempt is made to come up with a description which will explicate a valid mapping of action to outcome.

Michel Foucault uses the term *episteme* to describe the unfolding of such a frame (Foucault, 1970). An episteme denotes "the underlying, largely hidden grounds on which a statement or claim counts as knowledge during a particular period of human history" (Colapietro, 1993: 99). Within the context of the current discussion, we could say that an episteme defines the process by which the outcome of a mechanism, within an interaction, comes to make sense through description (Fig. 2).

The episteme as *open* framework

An episteme can be either *open* or *closed.* As closed, it is deeply coupled to the cultural/technical program according to which the mechanism is designed, and to the domain of interactions specified for that mechanism. As a consequence of this coupling, the very epistemology which prefigures the design of the mechanism determines, by virtue of that very prefiguration, the descriptions according to which its outputs become distinguishable as sensible units or 'gestalts.' When descriptions are fixed according to a well-understood history of interactions (as is the case when using word processors or driving an automobile), our actions and descriptions themselves become *effects* of those descriptions and of the episteme according to which those descriptions are mediated. In such a context, human action is couched within circumspective being.

By contrast, when understood as *open*, an episteme is no longer so deeply coupled to historical frames. Rather, it becomes, in part at least, an *emergent* frame, immanent in the very particularity of the thoughts, actions, and descriptions made with respect to a hypothesized object of interaction. The historical frame which normally circumscribes the domain of interactions appropriate to a particular task domain is disrupted. As a consequence, new distinctions are allowed to come to the foreground, situating a newly contextualized task domain. The episteme thus becomes *porous*, open to input from the particularized situation (Fig. 3).

Composition of process as the composition of interaction

The intentional disruption of task orientation constitutes an important aspect of creative problem-posing and problem-solving. Artist Robert Morris observes that art-making is a form "of behavior aimed at testing the limits and possibilities involved in that particular interaction between one's actions

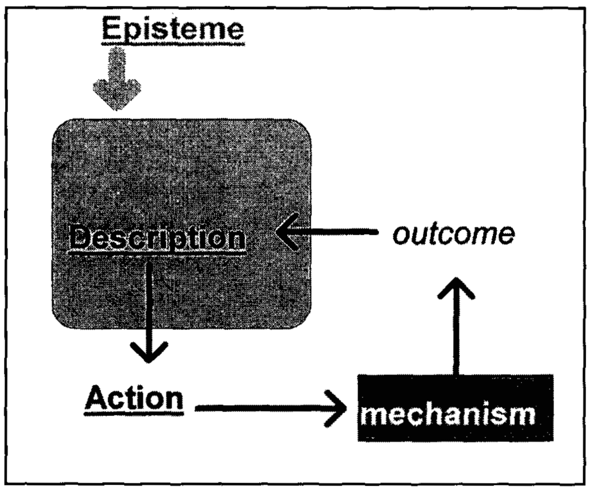

Fig. 2.

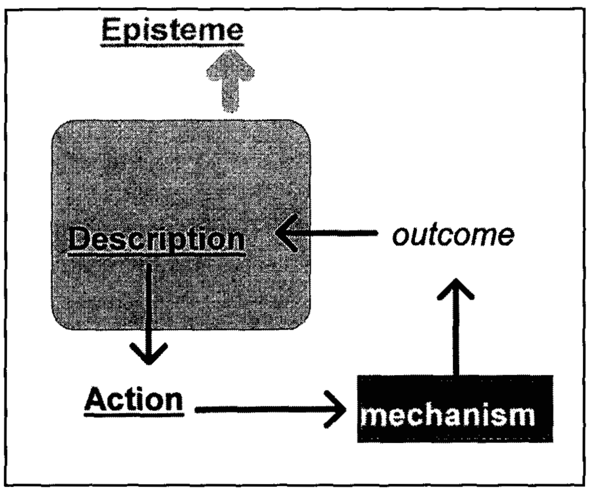

Fig. 3.

and the materials of the environment" (Morris, 1970: 62) Painter Jackson Pollock exemplified this approach to artmaking, laying the canvas flat on the ground in order to re-engineer the behavior of paint and of the bodily movements by which it is applied to the canvas. By recontextualizing the working environment, Pollock sought to alter the configuration of problems as they presented themselves for himself as a painter – to restructure the mechanisms, and thereby the descriptive framework according to which painting process might be exercised.

Similarly, many composers, from Cage to Stockhausen to Xenakis, have exercised various approaches to the situation of 'pre-compositional' activity in order to be able to generate information-rich musical data and processes for composition. Electronic music constituted one example of the application of this principle. While the use of electronic means for producing music was not in itself unprecedented – Theremin, along with many others, had already introduced electronic instruments – the notion that the technological problems of electronic music could become interchangeable with aesthetic problems was (Di Scipio, 1997). As Gottfried Michael Koenig pointed out, technical problems in the studio prompted investigative procedures "which [translated] the musical structure into a technical one" (Koenig, 1960: 53). The particularity of a technology and the means by which musical structures might be conceived and realized were thus understood to be mutually determinative. These efforts reinforced the notion that just as one might compose musical and acoustical materials *per se*, one might also compose aspects of the very task environment in which those materials are composed.

Composition with computers

With the introduction of the computer, the design possibilities of the task environment has been greatly expanded. As Otto Laske has observed, the computer can be understood as a tool which extends a human's self-referential activity into an allo-referential domain (Laske, 1980: 427). This is because the computer forces "musicians to focus on the pro-active, rather than re-active, aspect of their activity, [giving] them a chance to choose, rather than suffer, their processes" (Laske, 1991: 236). For this reason,

> [t]he computer has changed the potential of music theory since, for the first time, it has given composers a tool for capturing their processes, and for articulating a theory of music based on their knowledge of compositional planning and problem solving [as distinct from their knowledge of historical musical artifacts] (Laske, 1989: 46).

Laske posited "rule-based" approaches to composition in which "the task environment (including working memory) is cleansed of remnants of past experience as much as possible, in order to begin with a clean slate" (Laske, 1991: 240). Rule-based approaches to composition allow the composer to structure her/his task environment in order to short-circuit the data-oriented memory structures according to which human performance frequently lapses into habitual and historically determined models and to instead focus on procedural concerns. This involves structuring the representations and interfaces according to which tools are to be fashioned in order to investigate "how ideas activate and utilize reason, its objects and processes, in order to get realized, and how they predispose an agent to a specific approach" (Laske, 1991: 240).

This foregrounding of representation and interaction exhibits semiotic comportment in that it views music composition as a task that is as much concerned with the theories and procedures by which musical artifacts might be generated as it is with the actual generation of those artifacts. Rather than being deterministically coupled to a fixed or natural order, the computer is understood as a tool for situating domains of interaction whose ordering structure is immanent within the particularity of human activity. Historical and cultural frames can now be foregrounded and 'bracketed,' allowing composers "to choose, rather than suffer, their processes."

Taking this approach, we could say that composers are researchers in, and theorists of, interaction. Koenig's *PROJECT 1*, for instance, posited an interface in which, given a set of rules, a composer is asked to "find the music." Musical form was understood as a "strategy" by which the basic conditions for composition are established, rather than the details of a particular musical work (Koenig, 1991: 176). In other words, it was concerned with the "procedural" as much as the "declarative" dimension of compositional activity (Laske, 1989). For Koenig, this meant "generalizing formal relationships, which was at variance with the common practice of expressing musical ideals as concrete musical forms" (Koenig, 1991: 176).

In a somewhat parallel fashion, Koenig's *SSP* was concerned with the algorithmic composition of sound itself. This constituted a marked departure from the model-based approaches to sound design found in so-called 'sound synthesis' practice (Berg et al., 1980). In *SSP*, sound was conceived as patterns of amplitudes, a descriptive framework which could account for individual sounds using precisely the algorithms that it uses in composing entire compositions.

Paul Berg's *PILE* was also concerned with the algorithmic composition of sound, as opposed to synthesis using acoustical models (Berg, 1987). Here, as in *SSP*, the use of the computer is utterly compelled by an imperative wherein one might "hear that which could not be heard without the computer, to think that which would not be thought without the computer, and to learn that which would not be learned without the computer" (Berg, 1987: 161). The computer is no longer merely a convenient tool for synthesizing and modeling *natural* phenomena (musical instruments and the like), but one for formulating explicitly human-made and *artificial* phenomena. Toward this end, *PILE* understood sound design as a programming task, where a single program constitutes the entire sonic structure of a composition, from the lowest level microstructure of sound to the highest level musical syntax.

Herbert Brün articulates a similar sympathy for the computer as a tool for the design of artificial systems (Brün, 1969). In his composition *Infraudibles*, for instance, small pieces of waveforms were joined together through a variety of operations in order to form larger waveforms and, eventually, patterns of evolving waveforms. By this means, "the composer is able to control the infrastructures of the event, forming sounds just as precisely as the macro events of his composition" (Brün, 1969: 117). Brün's *Sawdust* was similarly concerned with the composition of waveforms in which musical structures were created "through the interaction of composer-specified waveforms with each other via predetermined algorithms" (Grossman, 1987: 218). Composing waveforms through a variety of combinations of basic operations on short sequences of digital samples allowed the composer to conceive musical signals as a consequence of choices made by a human being acting within a system of her/his own design. As Brün understood it, "only artificial systems will clearly show that they have been elected by choice, thus implying the intended rejection of other, equally possible, yes, even equally reasonable systems" (Brün, 1969: 119).

Other, more recent, projects manifest a similar understanding of the computer as *semiotic*. In the following discussion I describe, in some detail, two such projects. Both projects are designed for software sound synthesis and musical composition. Moreover, both projects implement, each in a different way, some notion of non-linear chaotic systems. One project does this through non-linear instruction-based synthesis, using self-modifying programs, while the other does this through the use of a physically based model of an analog system. Nevertheless, both projects understand the problem of human/computer interaction as a context for the projection of heretofore unrealizable notions of music composition and performance. Each such system focuses on a particular aspect of human performance, casting that performance within the context of unfamiliar, though highly articulated, notions of interaction, in an effort to transform the basis upon which action and outcome are interrelated.

Case study #1: The Ivory Tower Program

Ivory Tower is a real-time sound synthesis and music composition system developed by the composer Kirk Corey (Corey, 1997) along lines similar to *PILE*. It is a program plus a basic hardware configuration targeted for low-cost Intel PCs which reflects the composer's own idiosyncratic approach to composition through the specification of non-linear feedback systems (Corey, 1992). With *Ivory Tower*, a composer designs entire compositions through the specification of small chunks of program code and their interactions. In these systems, musical patterns arise from the interaction of component structures, the overall organization of which is entirely deterministic (there are no random or stochastic processes involved). Corey describes these systems as *autopoietic* in that the structure of their behavior arises solely from the interaction of their components and not from externally defined data (Corey, 1997: 81). As is the case in *PILE*, with *Ivory Tower* the program alone explicates the composition: there is no input data.

The software interface for *Ivory Tower* is comprised of a set of text fields (or as we will call them 'subroutine bins') in which the composer types in sequences of assembly language opcodes. These sequences of opcodes are executed immediately: the composer does not have to recompile the system. Among the opcodes that are specified in each of the eight subroutine bins are those which generate changes in the accumulator. The least significant eight bits of the accumulator are sent to the printer port once after each iteration of the control loop. Printer port outputs are sent to individual audio inputs of an audio mixer: each 1-bit output signal path of the printer port becomes one of eight inputs to the mixer. The input levels of the mixer are attenuated in order to bring the voltage level down to within line level range. The result is eight input channels of audio mixed down to two output channels (Fig. 4).

Waveforms are created through the toggling on and off of states within each of the eight bit positions of the accumulator (whose value is sent to the printer port once for every iteration of the program loop). By toggling states on and off within a

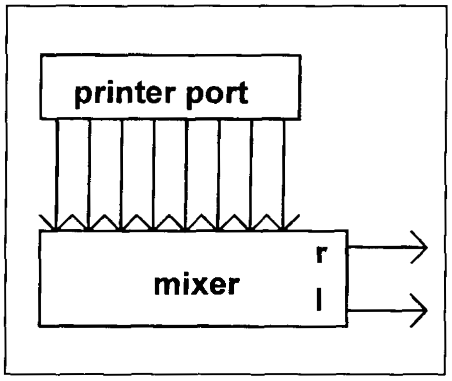

Fig. 4.

particular channel, rectangular waveforms with varying cycles (and thus frequency) behaviors are generated. Response from the system is in real-time, so the composer is able to observe the results of an action immediately after making it.

In addition to toggle instructions, the composer might enter instructions which change the cycle period for the various subroutine bins. Changing the cycle period of a particular subroutine bin effectively changes the rate at which the instructions defined for that bin are fired. If, for instance, the cycle period is 11, then that subroutine bin is fired once every eleventh iteration of the loop. By dynamically changing cycle periods for subroutine bins, one changes the time spans according to which patterns of accumulator values are toggled, thus effecting various kinds of acoustical behavior.

Another, related, type of instruction is that which makes changes to the *period mask*. The period mask is tagged to the cyclic period in that it defines a mask through which the period cycle is filtered (using a bitwise AND operation) before its value is determined. So, for instance, if the period cycle is 4 (bitwise 100) and the period mask is 15 (bit-wise 1111), then the filtered period cycle remains 4: the position of the bit '1' in the period cycle corresponds to the position of a '1' in the mask.

Finally, the composer might enter instructions which modify other instructions, either within that same subroutine bin, or within other subroutine bins. This introduces the notion of interaction among subroutine bins, a result of which is that "any change in one instruction in one [subroutine bin] may propagate throughout all the [subroutine bins], changing the entire course of unfolding of the sounds" (Corey, 1997: 86). Through this use of self-modifying code, the degree of complexity moves up to a higher level of abstraction. Low-level perturbations within the organization of a system can now propagate into its highest level of organization. The composer may now observe the consequences of her/his choices as they unfold over potentially long durations of time.

The acoustical results which obtain from the use of *Ivory Tower* exhibit a broad range of behaviors, from apparently random patterns of events and dense timbres, to sequences of utter redundancy (often the system may fix on a single waveform, or even DC offset for a long period of time). While using the basic operations and changeable period masks might generate waveforms and higher level event sequences, the introduction of self-modifying code allows for composition across a wide range of temporal levels: from the level of the waveform to that of an entire composition.

In this regard, *Ivory Tower* can be understood as a system for composition by *non-linear* instruction. In this case, the programs don't model a particular non-linear system: they themselves *are* non-linear systems.

Case study #2: synthesis by a physically based 'chaotic' model

Taking a different approach, Insook Choi has investigated use of a chaotic oscillator based on Chua's circuit (Chua, 1993; Kennedy, 1993) in the computational synthesis of sound (Choi, 1997, 1993; Mayer-Kresse et al., 1993). Typically, sound synthesis technique employs patching or other ways of combining different signal generators: complexity is generated through an accretion of interacting signal generators (Choi, 1993: 2). With Chua's circuit, however, the complexity of a signal is encoded within a single signal generator. Output signals range in complexity from simple periodic sinusoidal signals to complex non-periodic signals.

Chua's circuit is an analog system which is comprised of a small number of elements which, nevertheless, results in a wide range of characteristic behaviors. The dynamics of this system is defined by a set of three ordinary differential equations plus a nonlinear function. A digital simulation of the circuit has been implemented as part of an interactive sound synthesis software system running on an SGI (Bargar et al., 1994). Here, the simulated circuit is understood as an acoustical signal generator with seven parameters, each of which relate to the modeled circuit. These parameters are:

1. C1 (capacitor 1)
2. C2 (capacitor 2)
3. L (an inductor)
4. R (a resistor)
5. R0 (an added resistor)
6. BP1 (non-linear resister breakpoint)
7. BP2 (non-linear resister breakpoint)

The relationship between the values of these parameters and the state of the system is non-trivial. Experimental research involves varying any of the six components and observing the resulting output signal. Observation of the output signal is less *evaluative* than it is *informative*: the signal is understood as a trace of the state of the system and, as such, can assist the observer in framing hypotheses regarding the system's behavior.

Two different interface tools, which form a part of the NCSA Sound Server (Bargar et al., 1994), assist in this research. One such tool constitutes a bank of sliders, each one of which represents the range of voltages to be applied to the simulated component (Fig. 5) (Choi, 1993: 13). This interface permits investigation of the effect which changes of a single component can have on the behavior of the system. Such investigation provides valuable initial experience in learning the behavior of the system even if it does not necessarily permit compositionally relevant hypotheses. With such an inter-

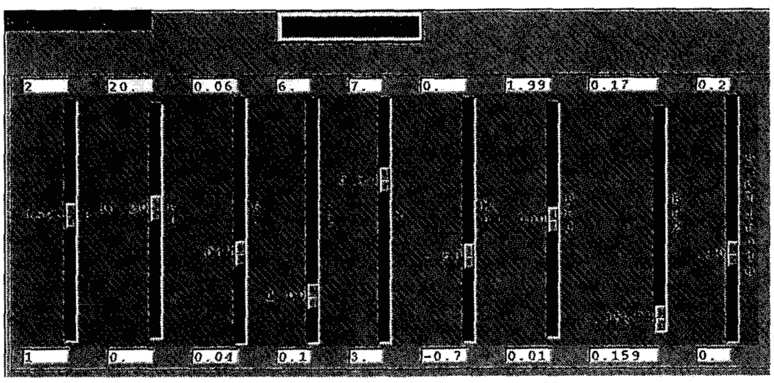

Fig. 5.

face, the composer can begin to understand the part each component plays in the overall behavior of the system and, as such, forms for her/himself a framework for higher level investigations.

Such higher level investigations are enabled through a second interface tool. This tool is a *manifold interface* (herein referred to as the 'manifold controller', or MC) for graphical navigation of multi-dimensional spaces within a CAVE environment (Fig. 6) (Choi et al., 1995). Three navigation concepts are employed within Manifold Controller:

1. phase space
2. path space
3. window space.

Phase space refers to an n-dimensional Euclidean space in which points define n-tuples which correspond to states (parameters) within the underlying system. The phase space defines all of the possible combinations of parameter values constituting a system or algorithm (Choi et al., 1995: 387). *Path space* refers to the possible trajectories that can be taken within the phase space. A *path*, as such, is a mapping from a time interval $[0,t_{Max}]$ to the phase space. The *Window space* "defines how a three-dimensional visual representation is embedded in the high-dimensional phase space" (Choi et al., 1995: 385). It constitutes a 3-dimensional graphical display representation of the phase space. A *window* constitutes a graphical display in which state is changed through the movement a pointing device (such as a wand or 3-dimensional space-ball). Movement of the cursor within the window defines changes in state within the window space.

An arbitrary number of window spaces can be linked to the phase space. Each window space defines a set of parameter attributes that form a subset of the total attributes possible within the phase space. The subset of attributes is designated through the choosing of a set of points within the phase space that are relevant to a particular desired investigation. Once a window space has been constructed, a composer defines paths within the window space by moving a cursor around within the 3-dimensional representation which constitutes the window space. That trajectory is mapped to the N-dimensional phase space through the use of a genetic algorithm.[1] The cursor position in the window space at any one moment maps to an n-tuple point within the phase space. A window space, as such, delineates a method for generating and modifying classes of paths, and thus constitutes a *view* into the phase space.

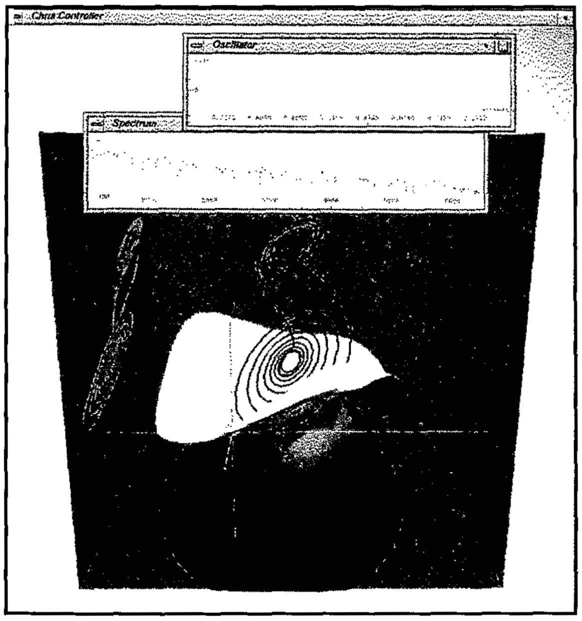

Fig. 6.

When interacting with the Chua circuit simulation, a composer may, with the Manifold Controller, explore the state space of that simulation. A composer begins by constructing different windows into the phase space (which defines the state space of the circuit simulation). Then s/he constructs paths (trajectories) within those windows in order to investigate the various acoustical outputs generated. Regions within a window, whose mapped acoustical behaviors are of interest to the composer, can be explored, with paths tracing those explorations being saved for later use. Paths become gesture notations that can be saved for later synthesis within larger compositional frameworks at which time other tools may be used in the facilitation of further experimentation and composition (Choi et al., 1995: 389).

Through the construction of windows and paths, a composer delineates a set of strategies for

[1] Space does not permit a discussion of this particular mapping; see (Choi et al., 1995, p. 38388) for more details.

exploration which become the basis for real-time 'performances', through which performances a composer constructs and assembles materials for composition. In this sense, her/his explorations are similar to that which a musician makes with respect to a musical instrument. Earlier on in the investigative process, one might make initial explorations of the system through the manipulation of single parameters using the sliders interface. Through use of the Manifold Controller, one gains what might be referred to as an *intuitive* orientation by which a correlation between action (the physical movement within the window space, the delineation of windows and paths) and outcome can be hypothesized. As with a standard musical instrument, one discovers various integrities and feedback principles through performance and practice. Unlike a standard instrument, the mapping of action to mechanism can be explicitly structured.

Discussion

Each of the projects just described situate a domain of interaction in which notions of structure and process arise within an emergent frame. With *Ivory Tower*, for instance, a composer can, at first, only make guesses as to how an action might effect the underlying mechanism. The nature of the outcome is dependent upon the nature of the actions taken. After a while, however, and with greater experience, systemic integrities begin to emerge – integrities which reveal themselves both as acoustical structures and as knowledge structures according to which one understands those acoustical structures as traces of an unfolding systematic organization. One learns about the system one is constructing by observing the sonic results as traces of the behavior of that system. For instance, as the consequence of a series of toggle instructions in *Ivory Tower*, a square wave has, by virtue of the 'strangeness' of the action taken to produce it, a new significance not afforded the square wave produced through use of a black box within the analog studio. It is no longer merely a square wave: it is also a mapping of a pattern of on/off states within a computer memory.

Similarly, with the *Manifold Controller* and Chua's synthesis instrument, a composer specifies interfaces with respect to which particular behaviors can be investigated and integrated into higher-level compositional formalisms. Again, through interaction, a composer not only produces interesting sounds; s/he also becomes knowledgeable about how those sounds are understood as traces of a particular phase space. This knowledge empowers the composer to construct higher level musical patterns based on observed mappings between particular phase and path spaces and the resulting signals.

In such an environment, one composes not only musical artifacts, but also the very context in which those artifacts are understood as relevant traces of a particular articulated interaction. The *episteme* according to which musical signals come to be comprehended is no longer determined purely through historical referencing agents. Rather, it is emergent within the particular activity through which a composer frames hypotheses regarding the sonic and musical organization of a signal. Through such an emergent framework, the episteme arises within the context of *particular* descriptions made, and *particular* actions taken, in relation to a hypothesized mechanism. As such, it is no longer strictly constrained by cultural and methodological notions of a mechanism, and the domain of interactions appropriate to its use.

The composition of music interaction

A composer thus enters into a mode of interaction that has *semiotic* comportment. No longer bound, through immutable "function-relations," to already existing referents, the object of interaction takes on a playful relation with respect to which, as a signifier, it represents objects in a world. Rather than being arbitrarily fixed to signified elements through equally arbitrary and restrictive function-relations, the relationship of the object of interaction (the 'signifier') to the object(s) it signifies can be playfully rendered. This is done through the composition of interface agents (graphical objects as well as more deeply structured task environment principles) in which the denotative and metaphoric dimension is muted. Accordingly, the objects and principles involved in the projection of an interaction, rather than referencing already existing and well-known performance practices, orient patterns of action and description that introduce new and less well-known problem-domain concepts. Interaction becomes a means by which a problem domain is effectively extended, rather than a means by which it is appropriated in the accomplishment of a particular desired goal.

In this mode, a composer not only produces musical artifacts (i.e., 'pieces', 'sounds', etc.) – s/he also produces a theory of "model-making" with respect to which those artifacts are imagined, designed, and realized (Kristeva, 1986: 17). By enabling the composition of interface agents, the computer empowers a composer in shaping the epistemological environment in which compositional "strategies" (to use Koenig's term) are engineered.

A domain of interaction is delineated. Such a domain of interaction orients the composer in structuring compositional procedure as a set of tasks by which musical artifacts are generated. Artifacts include sounds and musical patterns as well as descriptive frameworks by which the composer understands those musical artifacts as traces of particular processes and interactions (Fig. 7).

Those traces impinge upon the experience of the listener: the musical artifact itself becomes a kind of 'interface.' *Listening* is one way in which an observer encounters it. As Kunst (1976) elaborately details, listening can be modeled as the discernment of differences through a process of "learning/unlearning." A composer shapes and structures the unfolding of such a process, and is as such interested in generating cognitive frames within which musical patterns playfully reference choices made within a system. Precisely how the composer sets this up is conditioned by the notion of task environment in which s/he works (Laske, 1991, 1989). When empowered to participate in the shaping of that environment, the composer reconditions the manner in which otherwise historically framed notions of materials and process are engendered. Within the context of that reframing, the composer is able to imagine and implement musical processes that might not otherwise occur to her/him (Laske, 1992).

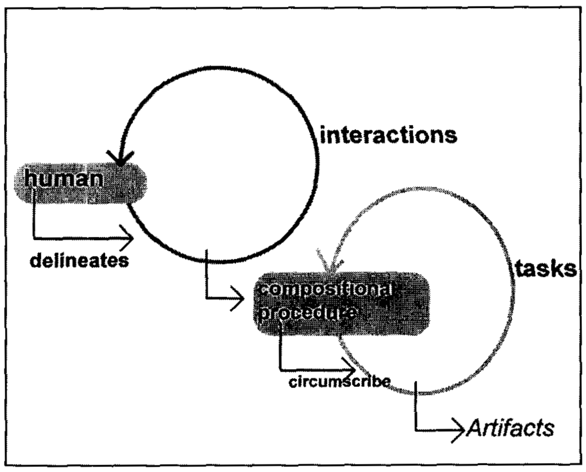

Fig. 7.

CONCLUSION

A composer makes traces of processes by which abstract ideas are concretized according to particular performances and interactions *vis* a task environment. The computer is a valuable tool in such an endeavor, since it enables the composition of interfaces and representations according to which musical problems may be formulated and solved. Consequently, the nature of the relationship between human and machine is transformed such that the human is no longer merely a "user" in the sense that the carpenter is the "user" of a hammer, but rather is a *participant* in a "game" of hermeneutic significance. This is because, the computer is itself a tool for the construction of tools – tools with which one might generate epistemological frameworks for imagining and solving problems of compositional significance. Like Pollock laying the canvas flat on the ground in order to effect a new way of interacting with his materials, the composer is empowered to playfully experiment in the elaboration of her/his relationship with acoustical and musical materials, and with compositional procedure itself.

In this paper, I have attempted to foreground the epistemological dimension of human/computer interaction in order to argue that it is precisely in this capacity that the computer becomes a valuable tool for problem-oriented tasks like music composition. In unfolding this argument, I began by describing the manner in which culturally and historically determined representational environments constrain human action and description in ways that are not always obvious, or even apparent. Having tools which allow for the explicit formulation and investigation of the representations and the interactions which encapsulate those constraints, however, a human is empowered to shape the occurring of her/his experience. As a tool for the articulation and investigation of one's own processes, the computer is more than merely a 'device' -- it becomes an agent in the fashioning of the very structures of representation by which one is oriented in thought and action.

For composers, such an understanding of the computer *as tool* invites a speculative attitude regarding the nature of musical articulation — an attitude which understands compositional procedure as that in which "the contemporary relevance and significance of the composition should be achieved, not by appealing to existing means of understanding music, but rather by creating new means for musical understanding" (Brün, 1970).

ACKNOWLEDGMENTS

A number of friends and colleagues have contributed, through conversation and collaboration, to the formulation of ideas presented in this paper. These include Mark Sullivan, Robin Bargar, Camille Goudeseune, Agostino Di Scipio, Sever Tipei, and Arun Chandra. I am grateful to Otto Laske, Paula Pribula-Hamman, Thomas DeLio, Insook Choi, Camille Goudeseune, and Kirk Corey for comments and feedback regarding the text of this essay.

This paper is dedicated to Gottfried Michael Koenig and to Herbert Brün, both of whose teaching and projects have inspired many of the ideas presented in this paper.

REFERENCES

Adorno, T.W. (1993). *Hegel: Three Studies*, transl. S. W. Nicholsen. Cambridge, Massachusetts: The MIT Press.

Anderson, P.B. (1997). *A Theory of Computer Semiotics*. Cambridge: Cambridge University Press.

Bargar. R., Choi, I., and Goudeseune, C., (1994). Model-based Interactive Sound for an Immersive Virtual Environment. In *Proceedings of the 1994 International Computer Music Conference* (pp. 471–474). San Francisco: Computer Music Association.

Berg, P., Rowe, R., and Theriault, D. (1980). SSP and Sound Description. *Computer Music Journal* 4(3). 25–35.

Berg, P. (1987). PILE–A Language for Sound Synthesis. in *Foundations of Computer Music*, ed. C. Roads and J. Strawn. Cambridge, Massachusetts: The MIT Press.

Brün, H. (1970). From musical ideas to computers and back. In H. B. Lincoln (ed.), *The Computer and Music*. Ithaca: Cornell University Press.

Brün, H. (1969). Infraudibles. In H. Von Foerster and J. W. Beauchamp (eds.), *Music by Computers*. New York: John Wiley and Sons.

Choi, I., Bargar, R., and Goudeseune, C. (1995). A *manifold interface* for a high dimensional control space. In *Proceedings of the 1995 International Computer Music Conference* (pp. 385–392). San Francisco: Computer Music Association.

Choi, I. (1997). A chaotic oscillator as a musical signal generator in an Interactive Performance System. *Journal of New Music Research* 26 (1), pp. 17–47.

Choi, I. (1993). A chaotic oscillator as a musical signal generator in an interactive performance system. Dissert., University of Illinois at Urbana-Champaign.

Chua, L.O., Wu, C.W., Huang, G.Q., and Zhong, G.O. (1993). A universal circuit for studying and generating chaos, part I: Route to chaos. *IEEE Transactions on Circuits and Systems — I: Fundamental Theory and Applications*, vol. 40 (10). 732–744.

Colapietro, V.M. (1993). *Glossary of Semiotics*. New York: Paragon House.

Corey, K. (1997). My Algorithmic Muse. *Sonus* 18 (1), 81–93.

Corey, K. (1992). *Music From the Ivory Tower and Elsewhere*, CD sound recording. Dubuque, Iowa: Curious Music.

Di Scipio, A. (1997). Interpreting Music Technology — From Heidegger to Subversive Rationalization. *Sonus* 18(1), 63–80.

Dreyfus, H. (1993). *Being in the World: A Commentary on Heidegger's* Being and Time, *Division I*. Cambridge, MA: The MIT Press.

Erikson, T. (1990). Working with Interface Metaphors. In B. Laurel (ed.), *The Art of Human-Computer Interface Design*. Reading, Mass.: Addison-Wesley.

Feyerabend, P. (1981). *Problems of Empiricism: Philosophical papers, Volume 2*. Cambridge: Cambridge University Press.

Foucault, M. (1970). *The Order of Things*. New York: Vintage Books.

Grossman, G. (1987). Instruments, Cybernetics, and Computer Music. In *Proceedings of the International Computer Music Conference 1987*. San Francisco: International Computer Music Association.

Heidegger, M. (1962). *Being and Time*, transl. J. Macquarrie and E. Robinson. San Francisco: Harper Collins.

Kennedy, P.M. (1993). Three Steps to Chaos, Part II: A Chua's Circuit Primer. *IEEE Transactions on Circuits and Systems — I: Fundamental Theory and Applications*, vol. 40 (10), 657–674.

Koenig, G.M. (1991). Working with Project 1: My experiences with computer composition. *Interface* 20. 175–180.

Koenig, G.M. (1959). Studio Technique. *Die Reihe* 1. 52–54.

Bryn Mawr, PA: Theodore Presser.
Kristeva, J. (1986). *The Kristeva Reader*, ed. T. Moi. New York: Columbia University Press.
Kunst, J. (1976). Making Sense in Music I: The use of mathematical logic. *Interface* 5, 3–68.
Laske, O. (1992). The Humanities as Sciences of the Artificial. *Interface (*21), 239–255.
Laske, O. (1991). Toward an Epistemology of Composition. *Interface* (20), 235–269.
Laske, O. (1989). Composition Theory: An Enrichment of Music Theory. *Interface* (18), 45–59.
Laske O. (1980). On Composition Theory as a Theory of Self-Reference. In *Allos*. La Jolla: Lingua Press.
Laurel, B. (1993). *Computers as Theater*. Reading, Massachusetts: Addison-Wesley.
Luchens, A.S., and Luchens, E.H. (1950). New Experimental Attempts at Preventing Mechanization in Problem Solving. *Journal of General Psychiatry* (42), 279–297.
Mayer-Kress, G., Choi, I.,Weber, N., Bargar, R., and Hübler, A., (1993). "Musical Signals from Chua's Circuit," *IEEE Transactions on Circuits and Systems — I: Fundamental Theory and Applications*, 40(10), 688–695.
Morris, R., (1970). "Some Notes on the Phenomenology of Making: The Search for the Motivated." *Artforum* (8), 62–66.
Norman, D.A., (1988). *The Psychology of Everyday Things*. New York: Basic Books.
Norman, D.A., (1986). "Cognitive Engineering." In D. Norman and N. Draper (eds.), *User Centered System Design: New Perspectives on Human-Computer Interaction*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.
Smith, S.M. (1995). Fixation, Incubation, and Insight in Memory and Creative Thinking. In S. M. Smith, T. B. Ward, and R. A. Finke (eds.), *The Creative Cognition Approach*. Cambridge, Massachusetts: The MIT Press.
Tipei, S. (1990). The Computer, A Composer's Collaborator. *Leonardo* 22 (2). 241–258.
Truax, B. (1986). Computer Language Design and the Composing Process. In S. Emmerson (ed.), *The Language of Electroacoustic Music*. London: MacMillan Press.
Truax, B. (1976). A Communicational Approach to Computer Sound Programs. *Journal of Music Theory* 20, pp. 227–300.
Varela, F.J., Thompson, E., and Rosch, E. (1996). *The Embodied Mind: Cognitive Science and Human Experience*. Cambridge, MA: The MIT Press.

Michael Hamman
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
705 W. Nevada St. #4
Urbana, IL 61801, USA.
E-mail: m-hamman@uiuc.edu

Michael Hamman composes and conducts research in composition theory, human/computer interaction, and computer-assisted composition and sound synthesis. His compositions include music for solo instruments, instrumental ensemble, tape, and computer. His music has been performed and broadcast in the US, Europe, and Asia and is recorded under the Neuma label (*Electro Acoustic Music V* and *Electro Acoustic Music VI*). His writings appear in *Interface, Computer Music Journal, The Music of Morton Feldman* (Greenwood Press), *Sonus*, and in the book *Otto Laske: Navigating New Musical Horizons* (forthcoming, Greenwood Press). Mr. Hamman's research in computer-assisted composition, sound synthesis, and composition theory has been presented at the International Computer Music Conference, Technology and the Composer Conference, International Conference on Auditory Display, Society of Composers Conference, Computers and the Arts, and the Contemporary Musical Festival in Seoul, as well as lectures in the US and Europe. Currently, Michael Hamman is working on experimental software systems for sound and music composition engineering, and on compositions for outdoor and indoor installation, human/machine interaction, and computer. In addition he is developing sound authoring tools for the internet as well as tools for research in human/computer interaction, both at the National Center for Supercomputing Applications at the University of Illinois Urbana-Champaign.