Emerging Materiality

Olav W. Bertelsen, Dept. of Computer Science

Morten Breinbjerg, **Søren Pold**, Dept. of Information & Media Science

University of Aarhus, Denmark

ABSTRACT

In this article we look at how materiality emerges from complex chains of mediation in creative software use. The primarily theoretical argument is inspired and illustrated by interviews with two composers of electronic music. We argue that computer mediated activity should not primarily be understood in terms of simple mediation, but rather as chains of complex mediation in which the dominating form of representation is the metonymy rather than the metaphor.

INTRODUCTION

Traditionally, the basic unit of analysis in the field of human-computer interaction has been the constellation of "domain object" - "user" - "tool", in which the user applies the computer tool to mold the domain object, in foreseeable ways according to a well-defined goal. Thus, focusing on efficient and transparent support of routine tasks has been the main concern. In the following, we will refer to this unit of analysis as the HCI triad. The focus implied by this triad has recently been challenged by concepts such as third wave HCI [1], that take a broader perspective on IT as used in non-work settings, and for creative action.

In this article we look closer into the relation between the traditional HCI triad, more complex mediation structures, emerging materiality of software, interface tropes, and creativity in software use. Looking at software as a material rather than just a tool is not a completely new perspective; indeed it is rooted in ideas that have positioned interactive technology as empowerment for creative intellectual work, e.g. Man-machine symbiosis [2], and the Dynabook [3], rather than efficient routine work support. In these early contributions, reshaping the tool itself was understood as an important aspect, thereby implying a more dynamic relation between the software, the user, and the object of interest.

Methodically, we develop a theoretical reconsideration of the HCI triad with respect to materiality and creativity. Our theoretical argument is illustrated and supported by two interviews with electronic music composers about how they creatively use various kinds of software. The interviews were conducted in the composers' studios and focused on how software mediated their creativity. The interviews primarily function as a source of inspiration and are interpreted hermeneutically for the theoretical development.

The two composers, Morten Suder Riis and Peter "PuzzleWeasel" Dahlgren are presented with their names and artistic, musical identities, since their artistic identity, poetics and use of software are closely connected. Both composers were enrolled as students of the Electronic Music program at the Royal Academy of Music, Aarhus. Riis has prior education in Musicology, and has been a performing rock and jazz keyboard player. Dahlgren's background is in the dance scene. Riis' work is centred on Max/MSP, which is a modular, open visual programming environment for electronic and interactive music, but he also uses sequencer software to arrange and control sounds. Dahlgren primarily uses sequencer software that presents sound as building blocks under a temporal perspective. The user can move these blocks around in various tracks played simultaneously, employ loops, repetitions and apply various effects, etc.

We have chosen to refer to the work of Riis and Dahlgreen as composition work because we focus on how they create new music, typically in a process where they iterate between various levels of constructing and exploring musical structures that later become actual music that someone can listen to. Traditionally, composition and performance have been separated practices, but in this field the distinction becomes less clear. Both in composition and performance, music creation may happen in real-time and in dialog with other musicians and technologies. For the present argument, we will not make any sharp distinction between composition and performance.

COMPLEX MEDIATION

As indicated above, the software use of Riis and Dahlgreen cannot be described sufficiently as a single HCI triad. The two composers explained that they use different kinds of software in constellations that we would describe as chains of triads becoming complex mediation structures in which each piece of software is not a simple means to an end. Software mediates the composers' creative action when producing music, but at the same time it is itself mouldable material for the composers.

###INDLED HER#### From the point of view of human activity theory, the direct relation between a subject and an object of interest is always intersected by mediating artefacts [4]. In such a triadic relation that is simultaneously direct and indirect, a composer of electronic music (the subject) would direct conscious action to the music (object), and these actions would be realised through unconscious operations on the software (mediating artefact). The conscious foci of the subject, e.g. the composer, are constantly changing. This is well known when playing music instruments; playing a tone is an unconscious operation in most situations, but becomes a conscious action if the instrument or the purpose changes for some reason. This constant change of conscious focus is a fundamental precondition for development and learning, and it is noticeable in the way the two composers of our study act creatively with their software. It is this kind of creativity in the developing use of technology that we aim to understand; creativity in the use of technology rather than creative action mediated by technology. Thus, we focus on creativity in the continuous reconfiguration of technologies.

In many situations, in particular situations involving information technology, the mediated relation is a chain of mediation where several entities are chained together and attain blurred roles as mediators and objects of interest; artefacts modifying other artefacts, artefacts turning into objects, etc. in elaborate structures. In these chains, human action (from an earlier practice) are often crystallized, or reified, in new mediating tools [5]. The complex mediation structures and the blurred foci in action, dissolves the clear distinction between tool and object, hence turning the tool into material; and the music, or other “object of interest” become instrumental mediators in the process of developing innovative use of the software and in reshaping software itself. A couple of examples from the interviews illustrate the point about creativity and materiality in software use.

The Max/MSP patches that Riis showed to us are clear examples of such reified actions and chained mediation. The patch becomes an extension to the already existing functionality of the software, and to the existing pool of libraries of patches. Patches are patched together to create new patches. A similar process is seen when e.g. popular combinations of filters are crystallized into new filters in subsequent versions of the sequencer software, as explained by Dahlgren. In the interviews it became clear that long chains of mediation are common - from the composer to the music that the audience experience. For example, Max/MSP mediates the programming of a Max/MSP patch, but the patch itself mediates both the performance situation (as a filter and instrument), and the composition situation (as a material resistance and as inspiration).

In the interviews, the two composers repeatedly reflected on the relation between the live performance situation and the compositional planning situation in the studio. Riis told about, and showed fragments of, a composition in progress together with a fellow composer. The performance involved the two composers' operating their PowerBooks and various attachments, as well as two bass players and a viola player. The strings played music partly written in ad hoc notation, and their sounds were to some extent processed through the computers (see figure 2). Not only is this a long chain of mediation, it is furthermore folded together into clusters of functionality, representations and inspiration. This is not specific to electronic music production, but a typical example of how computer applications take several simultaneous roles when used in a creative context, in particular how the alteration of the components between being tools and objects is so rapid that the distinction breaks down. The changed perspective from single triad mediation to complex mediation enables us to be more specific about how the software in the use of Dahlgren and Riis has material qualities. In order to better describe these qualities we conceptualize software as musical instruments.

SOFTWARE INSTRUMENT

Conceptualizing software as music instruments, or rather "software use" as an "instrument - musician relationship", helps us identify aspects of creative use beyond the HCI triad, in Dahlgren and Riis' approach to and understanding of music software. During the interviews, they both expressed their consciousness about selecting software as an integrated part of their creative process, thus pointing directly to a double character of the software as being tool and object at the same time.

Although it can, from an immediate point of view, be difficult to conceptualize software as a musical instrument, several characteristics are traceable. *First*, the software is playable. Sound generating processes run in real time, and sound can be manipulated instantaneously by moving a slider or turning a knob in the graphical user interface, or by typing in numbers or altering codes. *Second*, the software (filters, oscillators, reverbs etc.) has a unique sound profile due to the nature of the sound algorithms, and as such they are, in Dahlgren's words, comparable to different instruments. Different kinds of software sound differently, e.g. Max/MSP's filters, oscillators, and effects sound "clean" and recognizable with a "computer DSP" character (Riis). Therefore, both of them use several kinds of software to obtain a more refined sound and timbre. Consequently, they choose software like composers choose instrument groups (e.g. the strings or the wind) or like musicians choose instruments based on sonic preferences and characteristics, and based on various musical (sub-) cultural preferences and education.

Third, the software as an instrument can be seen in Dahlgren and Riis' understanding and approach to software. Like other professional musicians both of them recognize that the software instrument is part of the music and not just their access to it. Like a musical instrument, Dahlgren and Riis' software is an object of interest prior to and during the creative process of composition. Thus, both composers spend much time observing, configuring and in Riis' case even building the instrument while listening to the output without any predefined goal. As a mode of creative production this explorative approach of listening, configuring, operating and playing [6] involves a constant shift of focus between the sounding output and the settings of the software or, as we discuss below, the materiality of the software. *Fourth*, as is the case of playing a music instrument, Dahlgren and Riis accept that they have to discipline themselves in the use of the software to benefit from its complexity, but most importantly also to extend or perhaps even transcend its limitations.

SOFTWARE MATERIALITY

Software is, for obvious reasons, often is seen as immaterial. However, software holds a material dimension on several levels.

First, software is tangible in the sense that it can be controlled by physical actions. Turning knobs, pulling sliders, patching sound files on different tracks, connecting objects are actions by which we setup, configure and control music as well as software. Also, the functionality and physical appearance of graphical objects are supported by visual design and the possibility of direct manipulation that allows for intuitive interaction.

Second and more important in this context, software is a material in it self – because its functionality (understood in the broadest sense as the way in which music can be represented, accessed, programmed and edited) can be changed and reconfigured. Music as well as software is mouldable. Nonetheless, music software never presents itself as a "lump of clay". It is always preformed due to historical conventions of music: Genre, harmony, form, instrumentation etc that are metaphorically re-mediated in the interface through e.g. music sheets, carillons, tape recorders and synthesizers.

Third, and on broader level, software is materialized when somebody uses it, when it is interpreted and interacted with. This materialization happens on all levels both inside the machine, as bits, data, algorithms and layers of code; outside the machine, as input and output and i/o devices; in the user as interpretation and reaction, and in culture with its conventions and traditions [7]. Compositional software (as any software) is materialized from the low level of code and algorithms, to the interface, its metaphors and interaction, and to the sound, music and even visuals, that constructs the artistic output and how they connect to artistic, musical traditions. The processes and products of the software become form, materialized as text, visuals, interfaces, and sound, i.e., they become sensuous form with aesthetic, musical meaning with which the composer can work on all levels from programming to the graphical user interface, and from the sound to the musical and cultural contexts and traditions, which it becomes part of.

The software forms a resistance that both composers pointed at as a challenge and inspiration. In the creative act of composing they become deeply involved with the process, its intricacies and challenges as the music unfolds in their use of and struggle with the software. As such the resistance of the material is understood primarily as a positive and even necessary premise for being creative, since it brings about a reflective process in which the software is explored and new methods to overcome, reconfigure or transgress the inherent ramifications develop. Materiality can consequently be understood as an embodied resistance in software that composers have to struggle with in order to use the software creatively as something to play on and with.

TROPES AND MATERIALITY

When dealing with digital instruments, a special aspect concerning the resistance of the material is the metaphorical design of the interface. In order to develop the understanding of the materiality we will in this section look at it from the perspective of rhetorical tropes (i.e. words or sentences used in a figurative sense). We will point at the functioning of metaphors and move on to consider metonymy as a trope more adequate to grasp the challenges of the materiality and the chains of mediation pointed out above.

Metaphors are often the foundations of software design. The familiarity provided by metaphors in the interface enable the user to start using the application, and it is also often the foundational building blocks of the software [8]. We argue that the sequencer software that Dahlgren uses is metaphorically connected to earlier music automata and media forms, which deal with music as layers of structures unfolding in time, e.g. pin barrel programmed carillons, the score, player pianos and multi track tape recorders. As such, the continuous scanning of data on different tracks [9] is prominent in the interface and the basis for loop-based music production. In contrast the interface of Max/MSP that Riis uses is metaphorically connected to the synthesizer, because the user connects objects representing different functionalities (generators, filters) by the use of graphical patch cords.

From the composers' own accounts of use, it is clear that they typically transgress the metaphors, e.g. by relying heavily on general computer functionality such as automated and algorithmic procedures of manipulation. Dahlgren automatically cuts up sound samples in rhythmical structures; Riis uses random algorithms to control the flow of data in the patch cords that connect different objects. Thereby they move beyond the basic metaphors of the software and towards how they are implemented as software and connected in chains. A multi track tape recorder is an automatic playback mechanism of pre-recorded layers of sound, but automatically cutting up sound from analyses of volume levels in the source material is beyond what a tape recorder can do. Also, the continuous manipulation of the soundtracks in real-time that Dahlgren performs calls for an instrument approach to music composition that goes beyond the normal use of a tape recorder. In Riis's use of Max/MSP he equally moves beyond the synthesizer metaphor of the interface. Although he connects separate objects by the use of patch cords, his fascination with the software is also caused by the programmability of the software and its generative features, as when he sets up rules for the flow of music and allows for the control of sound by the use of sensors. As such, Riis composes event driven music, which is enacted through chains of performers, instruments, software and hardware.

What happens in these examples is that the metaphors (tape recorder and synthesizer) are subverted and reconfigured by the way the metaphors are implemented in the computer, and by the complex chains of mediation that are set up. *Playing* the software and these chains as an instrument – or perhaps an orchestra – adds to this. Thus the software and the chains are constantly reinterpreted, reconfigured and played through working with its materiality, and this reconfiguration is a key part of the creative process. As pointed out above, materiality cannot be reduced to a single level, but is something to investigate and reconfigure constantly. This is where the composer is engaged in a creative process with the software and the software is not merely a tool or a means towards an end but becomes a creative instrument.

The metaphor, and the intended use situation following from it, is only the starting point. The metaphor must necessarily be reconfigured towards a more personal engagement in order for the user to develop with the tool and use it as a creative instrument. This seems to be a specifically strong aspect of music production with its focus on developing a personal style through interaction with instruments. In addition, this material instrument perspective is relevant in other fields, where software is used in creative processes.

Metaphors have been widely discussed as a successful trope in interface design [10], but we need an additional concept to describe what happens in this sort of creative use situation. Rhetoric, since Jakobson, distinguishes between two major tropes; metaphor and metonymy [11]. While metaphor has been widely discussed, metonymy has only received little, if any, attention in the field of HCI and interface design. Jakobson defines metaphor and metonymy as two contrasting tropes that in different ways substitute one representation for another. Whereas metaphor is a trope based on similarity between domain and object, metonymy is based on contiguity and association. A typical example of metonymy is the crown for the king. Instead of substituting something with something 'like' the thing as metaphoric substitutions do, metonymies combine the representation with the represented on the basis of some material or causal relation of contiguity, e.g. the crown is a material attribute of being the king. When using metonymy one typically does not want to transfer attributes from one representation to another as in metaphors: There is nothing crown-like about the king, but there is a material association of contiguity.

Metaphors guide and control the user. Whereas the user should know what happens when employing a metaphor, as it is a translation of a well-known tool or domain, metonymy is opening up for new combinations and associative relations between material levels and between chains of mediations such as pointed out above. Furthermore, in computer game research, metonymy has been applied to capture the specific experience and pleasure of *playing* describing how the player experiences and plays the complex interplay between representations, dynamic properties, rules, algorithms and cybernetic feedback making up the game [12]. This is of course a clear parallel to the pleasures of playing an instrument or playing (in) an orchestra. As a musician, one does not necessarily fully understand the processes and interplays taking place, but one *plays* them, and in this playing the chains of mediations are associated metonymically. Consequently, metaphor may be the master trope for creating user-friendly and transparent software, while metonymy works better when designing for new and unforeseen uses like creativity and play.

What is metonymical design, and how do we recognize and further it? According to Jakobson metaphor and metonymy are not mutually exclusive but rather competing poles of symbolic representations [13]. They can be applied as design strategies in software interfaces – e.g. using metonymic contiguity and material combinations instead of metaphoric analogies and substitutions, or leaving the software open for metonymic displacements of the basic metaphors, thus creating less totalitarian metaphors open for creative reconfiguration. But metonymy is also often applied through users' (mis-) readings and (mis-) use, i.e. through more or less unconventional and creative uses of the software.

We see both strategies in our interviews and in the software used by the composers: Max/MSP is designed as quite metonymic and open for reconfiguration. The interface of Max/MSP is not just a visual surface, but a deeply layered structure that the composer can delve into and that can be programmed and re-programmed. It can be built entirely from patches, different views and connections. A large developer community benefit from this by making and exchanging patches that reconfigure and change the functionality of the software. This differentiates Max/MSP from the synthesizer metaphor and is a way of extending and modifying it to fit the computer (see figure 2). But in many cases metonymy also evolves in the use situation, for instance when Dahlgren uses imperfect automatic sample cut-ups and glitches, or when Riis uses generative procedures of composition. Such use of the software is furthermore increasingly assimilated in subsequent versions of sequencer software; thereby crystallising sidetracked development away from the grand metaphor originally staged by the developers. It is consequently apparent that the software is developed along metonymical lines, which are already provided for in the design as subversions and reconfigurations of the general metaphors.

Tropes such as metonymy (and metaphor) are designed as representations in the interface, as well as part of the interpretation and (mis)use, where users appropriate the software in a particular relation to the instrument-like characteristics. Strong metaphors have been successful for creating initial familiarity, but problematic in the sense that a strong metaphor closes the software and locks users' creativity, unless they are able to subvert the metaphor. More metonymic designs that involves combinations of views and levels in their interface – giving access to e.g. the sound on various levels, to its notations and to the codes and algorithms controlling both the sound and the software – might initially be harder to use, but allow for more richness and refinement. To a large extend Max/MSP even allows the user to construct his own interface. Which design strategy is most appropriate, and how it should balance metaphor and metonymy depends on musical style, poetics and artistic temperament, but the important point in both strategies is allowing for the possibility of metonymic uses and reconfigurations. The ability to reconfigure the software, reinterpret its use, and play its complex mediations are important features for the inspiration and for developing a creative praxis with the software – a praxis where the software becomes an instrument for finding original artistic solutions, precision and atmosphere, and for the pleasurable playability of the software.

CREATIVE SOFTWARE

In this article we have discussed qualities in software furthering creativity in use, in particular how a certain kind of materiality of mediating software emerges from complex chain structures. We have been interested in creativity emerging from the use of software. In our previous paper [14] we proposed the concept of 'instrumentness' to describe the two composers' feeling of working with software. In this article we have taken a closer look into the underlying mechanisms of the emerging materiality of software. Furthermore, we have argued for the emergence of metonymies in use (and design) as tightly connected to the emerging materiality.

If the challenge to design was earlier to replace a non-computerized tool, now the challenge is to introduce IT into situations that are already highly mediated and digitized, and where the software will be interlinked with other software and users in networks. Consequently, the traditional HCI triad has become insufficient since there is no clear cut domain object and often no well understood tool to model metaphorically. Electronic music composition has proved to be especially interesting for studying this since the focus and goal is creativity, the field is highly mediated to the point where the software has become the instrument, and electronic music emerges through playing the software.

The changing materializations of software, from the low level of code to the more tangible level of controllers, are where actions become reified and users get a grasp of, and are able to play with these chains of mediation. The materiality of software is a dynamically changing experience of a multiplicity of levels and complex chains of mediation.

One aspect of materiality is metaphor, relating the software to the domain object. When mediation becomes complex, other materializations dynamically come into view, and metonymy becomes both a design strategy and a user-interpretation. It becomes important to design in ways that are open for reconfiguration, so that users can have a variety of ways of using the software, and are able to express their use in and through the software. This can be designed as ways to reconfigure the interface, change views and develop patches, modifications and plug-ins, and as ways to develop personal and relatively unique user-styles. Users gain a personal relation with their software through metonymic engagement with the materiality, transgressing the boundaries through creative reconfiguration. This is a breakthrough in both the conception of the software and in the creation of what comes out of working with the software. As such our focus has been how working with software is in itself a creative process, and how the materiality emerging from complex mediation and metonymic representations furthers this creativity.

We have focused on electronic music composition, but we believe our analysis has wider potentials for understanding creativity in software use along a broad range of mundane and intellectual activities in both work and leisure. In computer programming we expect to see an analogous day-to-day reconfiguration and modification of the tools. Similarly, we believe that the analytical concepts could be useful in relation to mash-ups and other web 2.0 phenomena.

REFERENCES

1. Bødker, S. (2006). When second wave HCI meets third wave challenges. *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles NordiCHI '06*. ACM Press, 1-8

2. Licklider, J. C. R. (1960). Man-Computer Symbiosis. *IRE Transactions on Human Factors in Electronics*, HFE-1, 4-11, March 1960

3. Kay, A. and Goldberg, A. (1977) Personal dynamic media. *IEEE Computer*, 10(3), 31-42.

4. Vygotsky, L. (1978). *Mind in society: The development of higher mental processes*. Cambridge, MA: Harvard University Press.

5. Bertelsen, O. W. & Bødker, S. (2002). Interaction through clusters of artefacts. In *Proc. of 11th European Conference on Cognitive Ergonomics (ECCE-11), Catania, Italy, September 2002*.

6. Magnusson, T., Mendieta, E.H. (2007). The Acoustic, the Digital and the Body: A Survey on Musical Instruments. *Proceedings of NIME, 2007*, New York, June 6-10, 2007.

7. Duchez, M.-E. (1989). An historical and epistemological approach to the musical notion of "form-bearing" element. *Contemporary Music Review* 1989, Vol 4, pp. 199-212, Harwood, Academic Publishers GmbH

8. Erickson, T. (1990). Working with interface metaphors. Laurel, B. (ed) *The art of Human-Computer Interface Design,* Addison-Wesley, Reading, MA., 65-73.

9. Ludovico, A. (2004). The Sequencer Paradigm. *Read_Me Software Art & Cultures*. O. Goriunova, O., Shulgin, A., Aarhus. 127-136.

10. Erickson, *op. cit.*

11. Jakobson, R. (1956). Two aspects of language and two types of linguistic disturbance. In Jakobson, R., Halle, M. eds., *Fundamentals of Language*, Mouton, Den Haag.

12. Andersen, C.U (2007). Interfacet som våben, værktøj og legetøj [in Danish: Interface as weapon, tool and toy]. Hansen, L.K. & Pold, S (eds.), *Interface*, Aarhus: Aarhus University Press, 2007.

13. Jacobson, *op. cit.*

14. Bertelsen, O.W., Breinbjerg, M., Pold, S. (2007). Instrumentness for Creativity - Mediation, Materiality & Metonymy. *C&C'07, June 13–15, 2007, Washington, DC*, ACM-Press.

15. Thanks to Morten Suder Riis, Peter Dahlgren for sharing insights into their composition work, to Clemens Klokmose and Susanne Bødker for valuable comments, and to Elisa Giaccardi and Linda Candy for putting the special issue together.

Biographies

Olav W. Bertelsen is an associate professor in human-computer interaction at the department of computer science, University of Aarhus. His research is concerned with design-oriented perspectives on the use situation, e.g. experience oriented perspectives.

Morten Breinbjerg is PhD and Associate Professor in Multimedia at the Institute of Information and Media Studies, University of Aarhus, Denmark. He researches in computer music, digital aesthetics and software culture.

Søren Pold is PhD and Associate Professor in digital aesthetics at the Institute of Information and Media Studies, University of Aarhus, Denmark. He researches in the aesthetic interface and software culture.

FIGURE CAPTIONS

Figure 1: Complex mediation in a concert situation. Riis and co-workers playing at RAMA 06.

Figure 2: Max/MSP screenshot