Media in Transition 6: Storage and Transmission, April 2009

Paint, Sketches, Tools and Notes: Analogies of Code in Digital Arts

Aleksandra Kaminska
York University, Toronto
PhD Candidate, Program in Communication and Culture

The so-called Big Screen of Death made a dramatic appearance in Toronto during the Christmas season in 2007. On one of the major corners of the city, at Queen and Yonge, on the surface of the iconic Bay department store, there was a major screen disruption. Four giant screens were caught in all of their malfunctioning starkness, a scene that lasted a couple of days allowing photographers to capture this quasi-event and distribute it all over the internet's techy sites and local blogs. Some even suggested it might be the Biggest Big Screen Death of all time (Gizmodo). In many ways there is nothing unusual about this ubiquitous blue screen, and one can find numerous pages online documenting their occurrence around the world on screens of all kinds – from ATMs and airport information screens to one at the opening ceremonies of the Olympic games this past year, or on the various public displays on the Vegas strip. What was so grand about Toronto's own proud 'world-class' moment of computer failure was partly the amount of time the screens remained blue, neither fixed nor turned off, making one wonder if anyone was actually paying attention and in charge of this display. Viewers, almost in embarrassment, were left to gasp at the naked and bare screen almost wishing they could cover it up, or at least dim its bright 'look-at-me' lights. This dead and broken screen leaves no hints as to what was on it before nor what would come afterwards, reminding us of the fragility of our pixilated screen-based environment.

It doesn't really matter what the Screen of Death actually says. For most of us it is the visual appearance of that particular and powerful hue of blue that on its own is often responsible for immediately signaling alarm. Any language other than the error code, any display of numbers pointing to the location of failure is meaningless to the layperson, one step captured in time somewhere between bits and bytes, ones and zeros and a recognizable image on a screen. It is an error message that reveals a fracture between the storage and transmission processes, in this case a public art installation betrayed by its technological form. What the Blue Screen points to is a problem of transmission between what is stored and what is experienced by the viewer, an inability to transmit and translate the coded world into something visible and meaningful. The Blue Screen reminds us that what we see on screen is but a programmed mirage of sorts, that what we are experiencing is temporary and fleeting, that the screen itself is only the form, one which can in fact, at any moment, jolt us with that frightful blue, that storage and transmission are separate, and that the process of communication between the two is fragile. The idea that the so-called content, the 'work of art' exists in a coded and stored form somewhere, that what we are seeing is the product of complicated processes which are more scientific than artistic, only complicates the reception and understanding of the digital artifact. It is this question of how we understand what is behind the screen that I want to reflect upon today. One of the ways we have tried to investigate the behind-the-screen nature of digitality is through the use of analogies with other materials. So, what is code like? Is it like pain, or diagrams, or notes? These analogies are helpful not only in exposing the screen, but also in forming a base through which to create probes for analyzing digital materials and the nature of the storage and transmission processes which

is so central to the digital artifact in a way that focuses our comparison between the new and the old on materials, particularly on the transformation of artistic materials, and not only on the evolution of media technologies. This helps us frame digital art not only as a medium, but also as the result of artistic practices and histories.

It is perhaps a curious situation, this need to find a perfect comparison, since scholarship on digitality has also suggested that the digital is in fact the convergence of forms, that the medium is the message (McLuhan 1964) and that the separation of storage and transmission has created a new situation which bares little comparison with what came beforehand. But the way we conceive and conceptualize the materials of our digital landscape is, arguably, an important marker for how this material shapes our being in the world, for the ontologies of building and making digital media generate, and the new conceptual phenomenologies they produce (Knospe and Zhu 2007).

An important starting point for this analysis is an understanding that code, something defined as a system of symbols (Flusser 2002, 36) is one of the materials of digital art. If we can agree that digital culture is based on codes and languages, from programming languages to binary code, then it becomes difficult to dismiss the entire digital project as immaterial, whereby it loses all the information digitality contains within its material structure which defines and shapes its processes. We might be able to refine our analysis by accounting for the materiality that is found in these layers of language themselves (Drucker 2002) or suggest that materiality exists because code can only exist in its integration with other materials, such as hardware, in a sort of continuous materiality (Knospe and Zhu 2007). Therefore, the way we understand the nature of digital materiality provides us with a way of grounding storage and transmission

processes in material occurrences, an exercise which ultimately can be more productive than the almost fantastical notions which emerge from the rhetoric of immateriality. Digital code is more than a tool, and rather is inextricable to the artifact it transmits. Moreover "our understanding of technological processes must rely on the "specificities and materialities of different forms… and the phenomenological conditions they may generate" (Parks 2007, 208), and art works serve the function of translating technical knowledge…into aesthetic discourses" (Parks 2007, 208). In other words, the code of digital art serves as a mediator between scientific and artistic knowledge and experience, but is also constitutive of the digital experience for both artist and audience.

Analogies, as well as metaphors, abound in the computer realm and are revealing because they point to certain relationships, often highlight similarities at the expense differences, and can bring disparate things onto a common ground where they can be compared. The analogy we use has a big impact on our conceptualization of the digital form, on how we determine the value of coding, and what our expectations are in regard to the possibilities and limitations in the storage and transmission of digital artifacts. It speaks to the relationship between content, materiality and signal, but is also illustrative of the many layers of a digital artifact, or the complexity of digital processes, and the different values we assign to these components depending on their position or moment within the storage-transmission process. In finding productive ways of thinking through the material nature of the digital and digital artistic practice, we create nuanced ways of grounding, or separating, the new and the old. While the number of analogies is limitless and by no means is this presentation a comprehensive analysis of each possibility, what I

am questioning is why and how analogies matter – what are the elements which we are, effectively, comparing, and, moreover, what are the consequences of these comparisons?

Scholars in media studies and other fields have already spent considerable time analyzing the digital and the way it relate to previous forms through such ideas as remediation (Bolter and Grusin 2002) or the McLuhanesque notion that the content of any medium is another medium. Here what I am suggesting is that in focusing on the transformation and translation of materials themselves, on the way we conceive new materials using frameworks of older materials, we might be able to better understand processes such as storage and transmission, which help us characterize differences between materials.

So what is code like? It is like the paint of the painter, or the blueprints drawn by an architect, the tools of a welder, the notes of a musician, the letters of a novelist?

- **Paint**, perhaps more than any other of these examples, is intimately joined with its end product – in other words, the paint produces the image in such a way that the paint used can never be reused again. Any information is tied to the manipulation of the paint. The work becomes old or is destroyed if the material deteriorates. Storage and transmission are one an the same;

- **Sketches**, for example like those of the architect's blueprints, are a two dimensional visual representation, or plan, for a larger 3D project which will use a wide array of materials and tools to be completed. There is a transformation of materials from the blueprint to the end product, between the architect and the engineer. The metaphor of architecture is one that is widely used in computer studies. Knospe and Zhu describe the diagrams in architecture as "visual containers of a hierarchy of codes, [which]…

negotiate the space between the semiotic system and the physical world in a similar way to computer codes" and where "diagrams are not fixed but transient moments in an emergent material practice" (2007). Arguably, to see a sketch, diagram or blueprint provides us with more insight into a final built structure and enables the structure to be built multiple times, now or in the future. A sketch stores information but the final built structure, as the embodiment of the sketch, is in effect its continuous transmission;

- **Tools**, such as a hammer, contain no information, can be used for a variety of projects without inheriting a burden of meaning. They neither store not transmit, but can be helpful in the process between the two;
- **Notes**, such as those used by a composer, represent or store within them information of a particular sound, which can be performed using an instrument through a particular kind of reading, which must be explicitly learnt, to form a melody which can be transmitted to an audience in order to be appreciated;
- **Letters**, like notes, are reusable. There is an infinite supply of letter "A" and they can be read to form words and meaning. Like notes they represent a sound, but they only have meaning when they are strung together and can be read, at which point they represent things and ideas. They store and transmit and are what an audience directly experience. This is unlike the code of a computer, for example in programming language, or even less the binary, which the audience rarely reads or has access to, and even it did, would not necessarily be able to decipher its meaning or imagine its final product.

Though this is a sample and an oversimplification, we can see certain patterns or trends emerging in these analogies which point to the elements that we use as a basis for comparisons. These are analogies which try to find similarities with the digital but at

times can propose contradictory possibilities, revealing our ambivalence about code and spur us to combine, or recombine, the characteristics of these materials to explain digital code. Differing opinions are in part the result of differences in the way people experience code – some are very familiar with this language while for others it presents a barrier of inaccessibility; some find it artistic others utilitarian. And we continue to disagree on the point at which information becomes meaningful, on the relationship between what is represented and what is stored, and ultimately on how code relates to a final work.

So it is perhaps by turning to other materials and finding or focusing on the points of comparison that we can see what code is and isn't. For this we are still well served by using spatial and temporal critiques. The first encompasses frameworks such as the structure of a material, the number of transformations which are undertaken between an artist and the audience, and the matter of storage: where, what and how does storage occur in these different materials? These are related to the second type of analysis which addresses the temporality of materials through questions of longevity, permanence, and fleetingness.

1. Spatial Differences

Vilem Flusser proposed a historical outline of the evolution of codes, from the 4-dimensionality of our actual life-world to the 3D nature of sculpture; the 2D nature of images such as those that can be found in photographs, films and TV; the singular dimension of writing and the zero-dimensionality in the binary code and its representational form, the pixel (Fusser 2002, xxvi). This chronology or history differs from the premises of convergence in that it proposes a shrinking of our dimensions

depending on our materials, rather than the idea of an amalgamation and the creation of a mega-form which includes all dimensions and all information. The perception of the digital as either an absence or unison, hints at the way we perceive the kind of information that is contained in the material of the digital form. Depending on the analogy we choose, there could be a separation of 1, 2 or 3 dimensions between code and other materials. For example, if we choose the analogy of a sculpture, we lose 3 levels of dimension, three characteristics of space in our transcoding into digital code. On the other hand, an analogy such as a blueprint offers the possibility of gaining, rather than losing space in its transformation from 2-D through 3D and eventually 4D. Different analogies allow for different amounts of loss or gain of information as contained through each dimension.

Flusser also understood codes as a mode of thought, and as the structures which can change our being-in the world (Flusser 2002, 16). Digital codes in particular are understood as leading to "structural, systems-based, cybernetic modes of thought" (Flusser 2002, xiii), an extension of McLuhan's idea that writing changes the way we think. And yet in digital art we use this rigid language to create images and art, to be used as an artistic code which transmits all of the abstraction and nuance of artistic works, to be anything in fact, but linear. Flusser further distinguishes between 3 types of structures in codes, linear sequences such as spoken language and the alphabet; surfaces such as painting, and space such as theatre or architecture. TV in this scheme is a mixture of linear and surface structures. And we can assume that the digital in this scheme is also a mixture of these different structures, a combination of linearity through its code and language, and surface with the interface of the screen. The leap between these structures

is not only one of dimension and space, but also a difference in temporality, so that the linear and the surface exist in different time (Flusser 2002,16). This idea leads us to the second feature we can use to understand our analogies of materials, that of temporality.

2. Temporality

The idea of time as a crucial point of distinction between the linear and surface is not new, and can be traced to the writing of Innis, McLuhan and others. It is useful to think of the transformations of time which occur in digital processes to understand the fluid state of materiality between storage and transmission. From the period of storage and of textuality, where information exists as coded data, to the moment of the surface, there is a transformation of time. The transmission is not only of information, but a conversion between the temporality, the historicity and permanence, of the linear to that of the impermanence of the surface. This is unlike paint or letters, which do not undergo such transformations, or a blueprint, which, when the building is built exists only as a plan, and is no longer intrinsic to the building's existence. Perhaps the most similar comparison in this regard is the musical note, which undergoes a transformation from the linearity or surface of the notes on a page, to the auditory experience which give them meaning. Digital code, like musical code, must also be learned in a systematic way, like a new language which must be deciphered before it can be transmitted and communicated for the layperson to see or hear. The Big Screen of Death, for instance, is the visual manifestation of a failure of conversion, leaving the art work stored but not seen, existing in the temporality of storage and unable to transmit to the present and the now. Here is perhaps where our analogies are lacking, for they can't account for these system failures,

for the constraints on storage, archiving and reproducibility. Even the blueprint, which arguably reflects to a better extent than the other analogies, the complexity of a digital system, exists as a permanent code. Our analogies ultimately constrain us by trying to understand one of digitality's emblematic features, its complex ephemeral nature, through the lens of more archaic forms and archival codes.

While some media theorists have argued that in digitality there is no longer a distinction between content and materiality, many digital artists conceive of the code as a purely utilitarian tool distinct from their visible or audible artifact or performance. The invisibility of coding and language in much of digital art conceals the material in a way that forces us to question what constitutes the artistic product and the artist. In trying to find an appropriate analogy we are indeed torn between theories of digitality on the one hand, and of art on the other. It is binary code which, perhaps at the most fundamental level, prevents us from making accurate analogies to other materials even though many artists feel a disconnect with all levels of coding. The binary is the storer of information, yet information given in an unrepresentational way, at a level of fragmentation incomparable to anything else, meaningless when read by anything that isn't a machine programmed to interpret and translate this language. It is not the language an artists uses to create a digital work. It is the way the computer reads programming language before it can be transmitted onto a surface for the audience. The complexity in the transformation of the material, of the code, from language to binary to signal, while an oversimplification, is part of the very essence of digitality which is lost in analogies using analog materials.

It is perhaps not necessarily surprising that a viewer might be bemused by code, or any revelation of what goes on underneath a screen – after all in most art work the coding is kept hidden, exception being certain forms such as software art which exposes the codified material nature of a work or projects which aim to create an aesthetic of coding. What is perhaps more surprising is the disagreement among digital artists themselves in their perception and relationship to their material. During a debate on the Rhizome listserv a year ago ("Regarding Conceptual and Aesthetic Implications of Code in Computer-based Art," March 2008) one of the issues at hand was whether code merits studying and what, if anything, does the code of a digital work reveal about the work itself. Some suggested that in fact code is not important, not revealing, and "just not that interesting from anything other than an engineering perspective." This show the type of disconnect even the artists themselves have with their material, the separation of the code from the displayed object, but also as an increasingly artificial division between the engineer and the digital artist. Others in the meantime argued the opposite, claiming that the code itself is the art, that in looking at it we might gain better understanding of the final work, in the same way we might, arguably, better appreciate a building if we have a look at its plans. The list also proposed some analogies to code as a way of speaking and justifying these positions. One important insight gained from the discussion was that there is an important distinction between artists who use code, or software, as art itself, as is the case in software art, and the artists who use software and pre-written code to produce art, essentially relying on programmed options to produce a final work. Undoubtedly, the artist who manipulates the code himself has a different relationship to code than the artist who learns how to use the functionalities of software and access the

stored coded possibilities written by a programmer. Here I was primarily concerned with the artist who is also programmer, regardless of whether she uses code as art or code to produce art, though these differences in the way digital artists use, treat and understand their material should be looked at more carefully to further explain code as material and the digital as art form.

Conclusion

The analogies we use are either privileging certain aspects of code, or adopting the perspective that anything can be interchangeable with anything else. In other words, analogies fail us in their inability to account for the complexity of the digital structure, while at the same time by using analogies we construct an actuality, we imagine processes in a way which give them meaning.

Notions of materiality are entwined with concerns over storage and transmission, and our struggles with finding accurate analogies for code reveal our persistent ambivalence towards describing and understanding digital code. The implications of finding the right analogy are important for locating the storage possibilities of digital art in the material concreteness of language and code, but without losing the digital characteristic of immaterial, fleeting and ephemeral transmissions, processes and experiences. The use of analogies provides a way of comparing the form and materials of the digital and analog in a way that offers a glimpse at the peculiar position of digital artists and addresses the disconnect many artists feel in regard to their material. Our analogies have repercussions on how we understand digital art and lead us to ask questions such as whether we should be showing code alongside a work to the viewer?

Would an audience eventually learn to read and appreciate code, as it does a painting? Those working in digital art, whether as artists or curators, are concerned with the archiving of works, but can the way we understand the material nature of digital artifacts help us decide what should be stored, and what should just exist as a temporary transmission? Are the forms of documenting these temporary performances tied to our preferred analogies, and how do we account for information that is lost between these translations or transcodings, between these transformations of dimensions and time, and ultimately, between these different material universes (Flusser 2002, 14)?

Works Cited

Bolter, Jay and Richard. Grusin. *Remediation: Understanding New Media*. Cambridge, MA: MIT Press, 2000.

Drucker, Johanna. "Intimations of Immateriality: Graphical From, Textual Sense, and the Electronic Environment." *Reimagining Textuality, Textual Studies in the Late Age of Print*. E. B. Loiseaux and N. Fraistat, eds. Madison: The University of Wisconsin Press, 2002, 152-177.

Flusser, Vilem. *Writings*. Trans. Erik Eisel. Minneapolis: Univeristy of Minnesota Press, 2002.

Knospe, Kenneth J. and Jichen Zhu. "Continuous Materiality: Through a Hierarchy of Computational Codes." 2007. *Fiberculture*, Issue 11. Available at: http://journal.fibreculture.org/issue11/issue11_knoespel_zhu.html. Accessed on March 1, 2008.

McLuhan, Marshall. 1964. *Understanding Media: The Extensions of Man*. Toronto: Signet Books.

Parks, Lisa. "Orbital Performers and Satellite Translators: Media Art in the Age of Ionospheric Exchange." *Quarterly Review of Film and Video*. 24 (2007): 207-216.