

UNIVERSITY OF WESTMINSTER



WestminsterResearch

<http://www.wmin.ac.uk/westminsterresearch>

Perspectives on aesthetic computing.

Paul Fishwick¹
Stephan Diehl²
Jane Prophet³
Jonas Lowgren⁴

¹Department of Computer & Information Science & Engineering,
University of Florida, U.S.A

²Computer Science, Catholic University Eichstätt, Germany

³Centre for Arts Research, Technology and Education,
University of Westminster,

⁴School of Arts and Communication, Malmö, Sweden

Copyright © [2005] MIT Press. This is an electronic version of an article published in Leonardo: Art Science and Technology, 38 (2). pp. 133-141, April 2005. Leonardo: Art Science and Technology is available online at:

<http://www.mitpressjournals.org/doi/abs/10.1162/0024094053722372>

The WestminsterResearch online digital archive at the University of Westminster aims to make the research output of the University available to a wider audience. Copyright and Moral Rights remain with the authors and/or copyright owners. Users are permitted to download and/or print one copy for non-commercial private study or research. Further distribution and any use of material from within this archive for profit-making enterprises or for commercial gain is strictly forbidden.

Whilst further distribution of specific materials from within this archive is forbidden, you may freely distribute the URL of WestminsterResearch. (<http://www.wmin.ac.uk/westminsterresearch>).

In case of abuse or copyright appearing without permission e-mail wattsn@wmin.ac.uk.

Perspectives on Aesthetic Computing

*Paul Fishwick,
Stephan Diehl, Jane Prophet
and Jonas Löwgren*

In collaboration with Leonardo, we participated in the Aesthetic Computing workshop in the hills of southwestern Germany in July 2002. The workshop [1], organized by Paul Fishwick, Roger Malina and Christa Sommerer, was one week in duration and was attended by over 30 representatives of the following disciplines: art, design, computer science and mathematics. The purpose of the workshop was to define this area and to try to bring to light key aspects of the field from a variety of perspectives. An aesthetic computing manifesto was recently published in *Leonardo* [2]. Aesthetic computing is the application of art practice and theory to computing. Given this brief definition, several questions come to mind; we shall address these questions and then proceed with the purpose of this paper, which is to cover sample projects in the area and the different ways in which the authors proceed to accomplish their singular crafts.

One question regarding the name “aesthetic computing” may well be whether we can justify its existence. After all, is not all computing within the realm of aesthetics, and is there not already a significant number of projects that capture some combination of computing and art?

Let us begin with the first question. We will define aesthetics broadly as a combination of cognitive and sensory modes of experience, according to its scope in philosophy [3]. Evidence of the employment of the purely cognitive aspect of aesthetics can in fact be found within mathematics [4] and computing: One speaks of an elegant proof of a theorem, or a beautiful representation. With such qualifiers, the mathematician is usually referring to cognitively grounded aesthetics. Our intent is to explore aesthetics by means of the bridge that separates the cognitive and sensory perspectives—it is by crossing this bridge that we may enrich both design and computing. As Knuth points out in his discussion of Metafont, which underlies his TeX typesetting system, “Type design can be hazardous to your other interests. Once you get hooked, you will develop intense feelings about letterforms” [5]. In-

terpreted more generally, Knuth is expressing the idea of aesthetics as being more than the purely cognitive, even within computing, which grew out of mathematics. A textual section of a computer program will have both denotative as well as connotative signifiers, and it is easy to imagine that the program might align itself with the goals of art by using the types and variety of representations employed in the art world, thereby stretching the traditional boundaries of what may be considered a usable computer program representation.

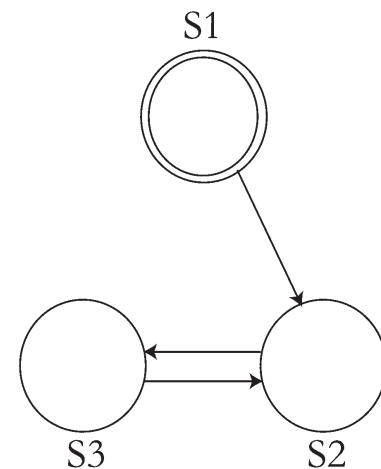
The second question, regarding the combination of science with art, is addressed by first noting that, unlike computer or digital art, the idea of aesthetic computing implies that art is affecting—and reflecting—some aspect of computing. This results in an agenda quite different from that found in many other fields of digital art.

To acknowledge our interest in aesthetics in computing is to expand human-computer interaction and representation to reach into numerous areas of computer science, from the primary operating systems interface to the interface used by computer scientists to create programs and by scientists to create models of geometry and dynamics. This step leads to two observations that partially justify the move toward aesthetic

ABSTRACT

The authors present an introduction to the new interdisciplinary area of aesthetic computing and proceed to define this area with examples from each of their own disciplines, practices and research. While several decades of publication and work have resulted in significant advancements in art as implemented through technology, less emphasis has been placed on studying the converse issue of art’s effect on computing, or “aesthetic computing.” The authors present their individual work in this area and then follow with brief criticism of one another’s work to elucidate different perspectives on the idea. By approaching the topic of aesthetic computing in this manner, the paper serves as an introduction to and survey and analysis of the field.

Fig. 1. Paul Fishwick, a 2D static snapshot of an interactive diagrammatic FSM interface. (© Paul Fishwick)



Paul Fishwick (educator), Department of Computer & Information Science & Engineering, University of Florida, Gainesville, FL 32611, U.S.A. E-mail: <fishwick@cise.ufl.edu>.

Stephan Diehl (educator), Computer Science, Catholic University Eichstätt, Ostenstr. 14, 85072 Eichstätt, Germany. E-mail: <diehl@acm.org>.

Jane Prophet (artist, researcher), Centre for Arts Research, Technology and Education, University of Westminster, 70 Great Portland Street, London, U.K. E-mail: <jane@carte.org.uk>.

Jonas Löwgren (designer, researcher), School of Arts and Communication, Malmö University, SE-205 06 Malmö, Sweden. E-mail: <jonas.lowgren@k3.mah.se>.

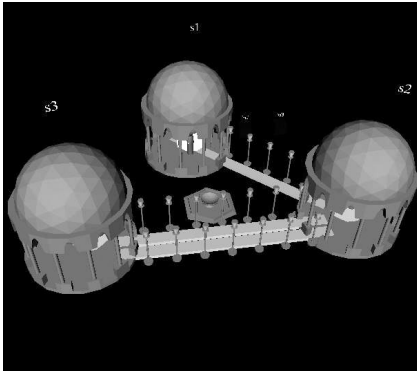


Fig. 2. Paul Fishwick, a 3D agent-based presentation of the FSM using the Virtual Reality Modeling Language (VRML). (© Paul Fishwick)

computing: (1) aesthetics in computing are broader than the purely cognitive dimension; and (2) the art-science confluence embedded within the discipline of interaction design is broader than the primary “desktop” interface.

The first observation is one of both ontology and epistemology—we might leverage existing aesthetic principles toward the sensory. Consider the case of “software patterns” [6] as one example. One can surely conclude that these patterns reflect abstractions of software structures and that they are found lurking in numerous software applications. There are two ways of looking at patterns, reflecting two ways of defining aesthetics: cognitive and material. What if the patterns revealed themselves in a way that was more attuned to material embodiment? This would allow us to build upon the existing pattern literature and extend it, as well as to extend the pattern representations. The “factory” method espoused by Gamma et al. could be the basis for a two-dimensional or three-dimensional scene that looks and operates like a factory. The look and feel of the factory would improve with artistic influence and guidance and would provide strong metaphorical cues. This sensory dimension seems lacking in the pattern literature; one generally finds that representation is limited to rectangles and arcs. And yet it is not only in the pattern literature that it is lacking, given that the visual minimalism of program structures and the mathematical structures underlying them is fairly common in computing.

The second observation implies that interaction design needs to concern itself with all questions of interaction and presentation found in computing, including that of how to represent data and program structures, for instance. The emerg-

ing areas of information visualization [7] and software visualization [8,9] represent approaches toward this goal, and yet their representations of information and software could benefit from greater emphasis on a wider range of artistic expression without sacrificing utility. So it is not that the area of design does not presently concern itself with incorporating aesthetics but that the current level and degree of this incorporation need to be expanded beyond those of the typical user interface at the operating systems level.

We now have a working idea of aesthetic computing, and we need to explore different approaches. There follows a statement on aesthetic computing by each author, followed by a discussion of one another’s views and statements.

PAUL FISHWICK: FORMAL REPRESENTATIONS

Scope

In applying aesthetics to computing, we need to confine ourselves to some aspect of computing, or one of its subfields such as human-computer interaction, visualization or discrete structures, to name a few. At the University of Florida, we have constructed a software system called RUBE [10–13], in which we have focused primarily on representations in mathematics and computing notation, from the notation of algebraic and differential equations to that of program and data structures, informed, however, through an artistic sensibility. Our basic idea has been to build a system that allows a multiplicity of different notations to be constructed so that one may see and hear the same underlying formalism in numerous ways. Not only do different people and cultural entities enjoy working with a formalism using different metaphors, but a single person or group can also benefit from exposure to diverse presentations.

Implementation

RUBE therefore allows for different representations to be applied to a select number of formal dynamic model specifications. Using RUBE, it is possible to change the way formal models look and sound. By formal models, I am referring to a large class of models used to specify systems that incorporate time for analysis and simulation, such as finite state machines, Petri networks, Markov models, queuing models and System Dynamics graphs, as well as ordinary and partial differential equations. RUBE uses XML (eXtensible Markup Language), which separates content from presenta-

tion. In XML parlance, content refers to an abstract specification defined as a document tree, and presentation refers to the way the tree is presented to the user—the way it looks and sounds. Thus, using RUBE and guided by the XML philosophy, one may specify an equation and choose to present the equation as either linear text, a network or a 3D structure. The choice of which presentation to employ can be determined by XML style sheets and their associated transformations.

RUBE’s architecture is based on open-source software and begins with authoring toolkits: SodiPodi for 2D vector drawing and Blender for 3D modeling. Let us consider the 3D pipeline beginning with Blender. The user creates a 3D model in Blender and then uses a Python scripting interface, which allows attributions to be made regarding semantics. For example, one might point to an object and designate it as a state or a function. After the semantic assignment, an X3D (eXtensible 3D) file is created for the presentation and a special XML file is created for specifying the formal model. After some XML transformations, this XML file is translated into Javascript or Java, whereby it can be reincorporated into the VRML (Virtual Reality Modeling Language) file, resulting in an interactive VRML world. The 2D transformations are similar, except that Scalable Vector Graphics (SVG) is used for presentation.

Let us begin with a formal definition of a Finite State Machine (FSM) M [14]. These machines have states that are interconnected through transitions that go into effect when an input to the machine is of a particular value. Here is a formal definition for M :

$$M = \langle Q, I, O, \delta, \lambda \rangle$$

$$Q = \{S1, S2, S3\}, \delta: Q \times I \rightarrow Q$$

$$\delta(S1, 0) = S1; \delta(S2, 0) = S2; \delta(S3, 0) = S3;$$

$$\delta(S1, 1) = S2; \delta(S2, 1) = S3; \delta(S3, 1) = S2$$

$$I = \{0, 1\}, \lambda: Q \rightarrow O$$

Even though this text might seem to be the formal specification for M , it is actually one of many types of presentation of the underlying formalism that is encoded in XML. In general, all presentations require additional natural-language semantics if we are to make sense of them. Q is the state set for M , I the input set, O the output set, δ the transition function from one state to another and λ the output function.

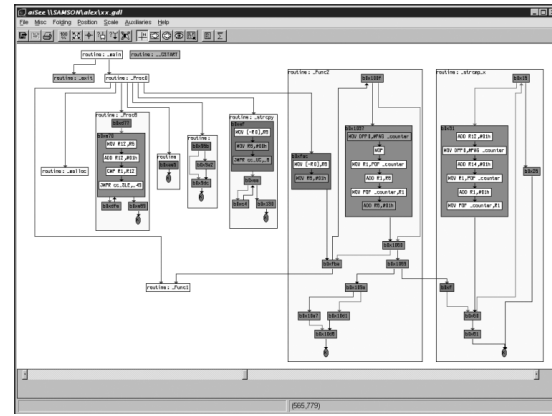
Figure 1 illustrates our second presentation of the FSM. It has iconic properties such that when the machine is in state $S2$, the presentation of a circle for $S2$ encodes the concept of a boundary and

that which is inside it. That is, the graphical depiction of S2 is consistent with the underlying metaphors of set theory, whereas the purely textual presentation does not capture these metaphors. Moreover, Fig. 1 is incomplete on a non-interactive medium, such as paper, since the additional information encoded but not visible in the text representation is obtained through minimal human-computer interaction (e.g. pressing a button to yield the text layer). Similarly, the arrows capture the notion of transition from one state to another, since anyone who has seen an arrow fly knows that it is aimed toward a target. The metaphors of the figure dramatically improve visualization of the semantics of the machine, and so one is led to wonder whether employing presentations with alternative aesthetics might further improve the impact of the metaphor. The underlying assumption is that material aspects of levels of representation are based largely on what is available for a society, as well as what is affordable and materially efficient. Consider Fig. 2 as a representation that has only recently become possible through computer graphics and the ability to employ 3D components. The metaphor of the circle as a boundary has been replaced by a small gazebo-like structure. The arrow in Fig. 1 is now shown as a woman walking from one state to another along a lamp-lit walkway. While Fig. 2 could be seen as introducing gratuitous imagery, it can alternatively be seen as a structure that reinforces the spatio-temporal metaphors that Fig. 1 only begins to reveal. The idea here is that an FSM, by its formal definition, is rich with metaphor and that Fig. 2 exploits the metaphor by surfacing it through visualization, interaction and familiar icons. Movement from one state to another looks like movement, and the thing moving is more familiar to us than a mental construct would be.

A host of philosophical issues comes into play here. Is there not a need to enforce visual minimalism within this sort of structure? What are the cultural barriers that might prevent the adoption of models like Fig. 2 in science and engineering?

With respect to the first issue, we should note that it is quite possible to maintain abstraction without requiring visual minimalism. Within an artistic context, this can be seen when comparing and contrasting the genres of Abstract Expressionism and Surrealism. Both of these genres contain a wide variety of works that employ symbolism, iconography and rich semiotics even though

Fig. 3. Stephan Diehl, program structure with analysis results. (© Stephan Diehl)



the visual presentations are strikingly different.

The expression of an abstract state in an FSM, for example, need not require that the state be presented visually in a minimalist fashion. The key task is to strengthen the metaphor of what it means to be a state and the corresponding elements of boundary that go along with it.

The second question, about cultural barriers, may lie at the heart of the aesthetic computing challenge. Computer scientists have been educated with minimalist figures and text, and so it may come as a shock to realize that our representations of formal objects are not as constrained as we may have thought. Until the era of computer graphics and fast computers, we had little need to inquire about what initially appeared to be exotic ways to encode formal knowledge. However, this is a challenge not only for computer scientists but also for artists, since artists should be encouraged to consider the computer and computing practices as subject material in addition to raw material. This suggestion of formal structures as raw or subject material may strike some artists as a modernist-era agenda; however, the computer and its mathematical foundations enable the creation of significantly higher complexity as a tool, or as a subject, than paint, palette knife or chisel ever could.

STEPHAN DIEHL: SOFTWARE VISUALIZATION

Software is neither matter nor energy. It is just a kind of information. Matter and energy are media that carry information and thus software. For new software to be developed or existing software to be understood, it has to be projected into a humanly readable form—the program text. Note that the program text is not the software, just a representation thereof. The

program text is written in an artificial language with a strict syntax and more or less well-defined semantics. Trying to understand a real software system by reading its millions of lines of program text is a vain task. As a consequence, many tools have been developed to support software understanding. These tools rely on analysis and visualization techniques. In a famous paper, Turing Award recipient Fredrick Brooks even stated that “software is invisible and unvisualizable” because each kind of visualization only addresses “one dimension of the intricately interlocked software elephant” [15]. These dimensions include the static structure of the software, its dynamics and its evolution. To put it in other words, different kinds of visualizations show how software is encoded, how it behaves and how it is developed. In the following I present and discuss examples of visualizations for each of these dimensions.

In Fig. 3 a graphical representation of a program, its control-flow graph, is shown. In addition the graph contains some information computed by a program analysis. With the help of this visualization, developers can detect certain kinds of errors, so-called stack overflows, in their programs [16].

In Fig. 4, a snapshot of the animated execution of a sorting algorithm is shown. The window contains several representations or views of the data sorted. Algorithm animations are typically used in education. Most of the animation techniques do not scale for real software systems.

Finally, Fig. 5 is a pixel map. In this example the color of the pixel at position (x,y) represents the number of times files fx and fy have been changed together relative to the total number of times file fx has been changed. From this figure the developer can see how strongly different files are coupled. We call this kind of cou-

pling evolutionary [17], because it is based on the change histories of files, to distinguish it from the logical coupling usually used in software engineering. As the files are sorted by their containing directories, the pixels form blocks. These blocks indicate that files within a directory are coupled, that is, often changed together. Software developers are mainly interested in the outliers: those pixels representing couplings between files in different directories, such as those labeled “Patches” in Fig. 5. Outliers can be a sign of a bad system architecture.

We have now seen three very different kinds of visualizations as they are used for understanding software and its development process.

Now what about the aesthetics of these visualizations? It may come as a surprise to the artistic reader, but there actually were some aesthetic criteria involved in the computing of these visualizations:

In Fig. 3 the number of edge crossings and bends has been reduced, and directed edges are mostly drawn downwards. In Fig. 4 color is used consistently in the different views. While the algorithm actually performs discrete changes, the animation of these changes is executed smoothly. In Fig. 5, color coding based on the heat metaphor was originally used: The color red showed the highest coupling, blue low coupling and white no coupling at all. In other words, hot files are those that are often changed together.

The sole purpose of the above-mentioned aesthetic criteria is to produce visualizations that convey the information as clearly and effectively as possible. As a consequence of concentration on the automatic generation and usability of vi-

ualizations, the current research on software visualization is rich in the different properties of software that have been visualized but poor in the spectrum of visual metaphors used: boxes, circles, lines and color.

JANE PROPHET: INTERDISCIPLINARITY AND AESTHETIC DIFFERENCE

I consider aesthetic computing through a discussion of the project *Cell*. *Cell* is an interdisciplinary collaboration between an artist (myself), a liver pathologist (Neil Theise), a mathematician (Mark d’Inverno), a computer scientist (Rob Saunders) and a curator (Peter Ride). *Cell* explores new approaches to the representation of cell behavior, using mathematics to bridge the gap between scientific theory and computer visualization. Related literature is found in Goodsell’s research [18]. Results include solely authored and co-authored papers in peer-reviewed medical, mathematical modeling and simulation journals; mathematical models of a new paradigm of stem cell behavior; dynamic simulations of the mathematical model; art installations; and illustrations of cells and their behavior generated using artificial life (Alife) techniques. Here my focus is on the graphic outputs from *Cell*’s agent-based Alife simulations.

Context

Cell’s practical research context ranges from Theise’s medical laboratory to d’Inverno’s and Saunders’s mathematical and computer science labs and to my art studio. Each setting has particular embedded methodologies, from the hy-

pothesis-driven ethos of the medical research lab to reflexive practice in the art studio to the empirically driven environment of mathematics. The aesthetic context of each discipline is equally diverse. Visualization in the laboratory differs from visualization in the art studio. In the medical laboratory, representation is usually taken literally, as scientific illustration. In the art studio, “representation” is a term and process framed by cultural and art historical theories of representation (for example, an image, sound or object can signify something without actually sounding or looking anything like it). In *Cell*, we document, develop and evaluate the interdisciplinary collaborative process itself, including discussion of our different aesthetic values.

Through studio and laboratory visits, Theise and I identified significant aesthetic differences. In cell biology the “photographs” of tissue slides have a truth-status and are accepted as “proof” of experiments within papers. The beauty of these representations is also valued. Figure 6 shows one of Theise’s images, representing skin tissue from a female mouse who received a bone marrow transplant from a male mouse. The tissue has been dyed, as is typical in such experiments, and blue nuclei of hair follicle lining cells surround the orange, autofluorescent hair shaft (large arrow). Two of these nuclei contain fluorescently labeled Y-chromosomes (small arrows) indicating that they derive from the donated male bone marrow, not from the female’s own original cells. Thus, bone marrow stem cells have given rise to skin-type lining cells.

As is typical of contemporary artists, I was educated to resist beautifying artifacts and taught that photographic representations have no truth-status and that their meanings are subjective rather than objective. Alife systems that have a graphical output pose further challenges to notions of representation. Such outputs are at least semi-autonomous; images can endlessly change (some might say evolve) as they are produced in real time to represent the software running beneath them, which is constantly changing as a complex system of interactions between agents takes place.

Converging: Focusing on Behavior across Time

Theise’s research examines the plasticity of adult stem cells and their function, using processes including the analysis of specimens of cell tissue. Because the tissue is dead at the time it is analyzed, it represents a frozen moment in time,

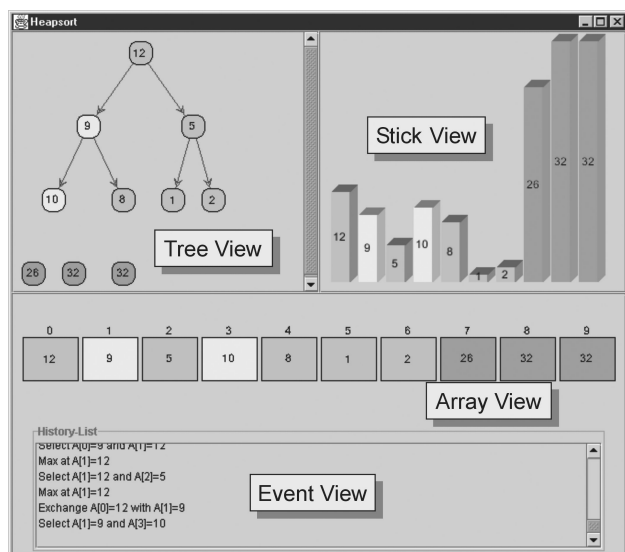


Fig. 4. Stephan Diehl, animation of the Heapsort algorithm. (© Stephan Diehl)

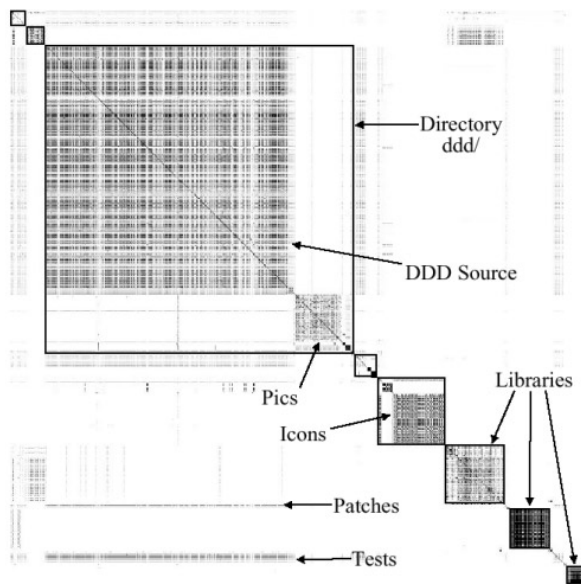


Fig. 5. Stephan Diehl, number of simultaneously changed files. (© Stephan Diehl)

from which researchers hope to understand another aspect of stem cell behavior and extrapolate further hypotheses to test. My experience as an artist working with time-based media and Alife suggested a different approach to assessing stem cell behavior. We therefore developed an Alife engine to enable the scientist to study simulated stem cell behavior as it happens within the complex system of a wider community of cell types and enzymes. Stills from the real-time 3D graphics driven by our simulation (Fig. 7) show images that draw on the aesthetics of medical illustration. The user clicks on a moving cell to see a data read-out. Figure 7a shows photorealistic semi-transparent cells moving, dividing and dying in 3D space. Figure 7b illustrates information available when a user clicks on a cell or activates a global command that makes the data for all cells visible.

The initial *Cell* collaboration expanded to include d’Inverno, who determined the mathematical rules that describe stem cell behavior as proposed by Theise and modified by me (that stem cells can evolve into mature cells of other organs such as skeletal muscles, bones and brains, with unexpected plasticity). Saunders interpreted the mathematical model producing real-time graphical displays with me. The aesthetic (designed by Saunders and Prophet) is influenced by medical and scientific illustration.

Aesthetics and Conceptual Art

The notion of “an” aesthetic is challenged in *Cell*. As in much conceptual art, the idea behind *Cell* (namely, modeling the behavior of stem cells) and the means of producing it (via interdisciplinary collaboration) are more important than the

finished work or its (fixed) appearance: “In conceptual art the idea or concept is the most important aspect of the work . . . all planning and decisions are made beforehand and the execution is a perfunctory affair. The idea becomes the machine that makes the art” [19].

Conceptual art can be defined as the “appreciation for a work of art because of its meaning, in which the presentation of shape, colour and materials have no value without the intentions of the work” [20]. If conceptual art has an aesthetic, it is the dematerialization of the art-object; the object only has value as a materialization of the idea, not in and of itself. Mathematics and computing science can both operate without materiality and can describe the immaterial, which is one reason why there may be a mutual attraction between computer scientists, mathematicians and artists working conceptually using digital media.

Aesthetics and Visualization

To date, the real-time graphic representation arising from *Cell*’s complex adaptive system (Fig. 7 and Color Plate A No. 2) is not what I would consider a piece of fine art in its own right. This is scientific visualization, or graphics, informed by my aesthetic framework. I want the graphic look and feel to reflect the underlying software, to draw attention to the essence of the idea or concept. From this standpoint the 2D Java version (Color Plate A No. 2) is a more satisfying outcome than the 3D version. Here, my color palette has white stem cells. Cells in each lineage tree are a particular hue, becoming darker as they differentiate through division. Lines signify genes within circles defining cells. I use the

screen as a framing device; stem cells are “born” at the center of the screen, moving to the edges as they age, divide and die. The line and single tone for each cell is deliberately abstracted and looks inorganic. This is more aesthetically pleasing to me—it makes no attempt to suspend the viewer’s disbelief. It is obvious that this work is computer generated.

By contrast, the 3D version has been influenced by the aesthetics of medical illustration and its goal of explaining via precise observation of the appearance of things; this is at odds with my emphasis on the behavior of things (in this case, stem cells). This version (using the same underlying Alife simulator but with a different graphic output) is characterized by its photorealism: detailed surface rendering, depth and transparency (Fig. 7). This was pleasing to Theise, as it was familiar to peers used to seeing video and stills of actual cells recorded via microscopes. However, Theise might determine system-level changes in behavior that could influence wet laboratory work more easily via the abstract 2D version. Anecdotal evidence suggests that abstraction prompts the viewer to look for patterns of behavior rather than focusing on the activity of individual cells.

JONAS LÖWGREN: INTERACTION DESIGN AND AESTHETICS

Interaction design is concerned with shaping the use qualities of digital artifacts. Another way of putting it would be to say that interaction design is design of the digital materials, where the word *design* is used in the strong sense of exploring all aspects of a possible future: aesthetic and ethical aspects as well as structural and functional.

It may seem odd, and in fact it is odd, to talk about aesthetic computing as if it were something new and hitherto unexplored. All computing is aesthetic in the sense that all use of digital artifacts entails aesthetic reactions. To be sure, many contemporary digital artifacts tend to elicit aesthetic reactions along the lines of frustration, indifference or boredom, but these are aesthetic reactions nevertheless, and as designers, we are free to aim for other kinds of reactions if we like.

Of course, there are historical reasons for the existence of such blind spots. The academic roots of interaction design stem from the field of human-computer interaction, where the main focus has always been on the task-oriented use of digital artifacts and in particular on its efficiency and absence of errors. It is not

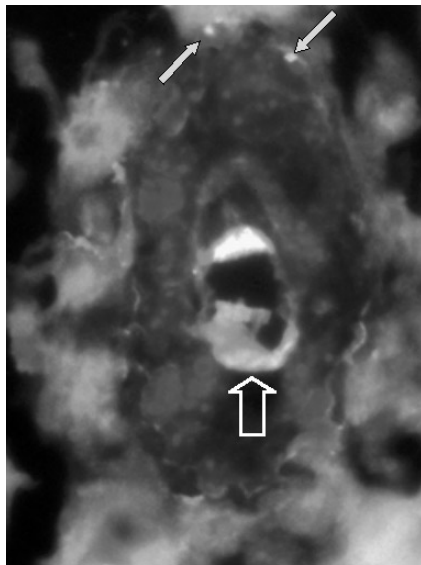


Fig. 6. Hair follicle with lining cells derived from bone marrow. (© Neil Theise)

surprising that the image of computer use is sometimes simplified to the extent that useful, efficient and error-free use is seen as the whole picture. The disregarded parts of the picture, including aesthetic qualities of the use, will then appear as new when they are brought into consideration.

How, then, should aesthetic use qualities be dealt with in interaction design? A common fallacy is to equate aesthetics with pleasing visual design. To be sure, there are design situations in which the immediate visual impression is an important factor in determining the outcome of the interaction. A good example is a web shop with one-time customers who ideally should spend money on their first and only visit. The visual design of the web shop's front page is crucial in establishing the right combination of desire and credibility to actually make customers enter, shop and pay.

In nearly every other case, however, we need to realize that a digital artifact is constituted primarily not by its static visual design but by its dynamic gestalt—the character of the interaction it allows over time. Digital design materials are temporal in this respect, at least as much as they are spatial. Below are three examples of aesthetic qualities in the use of digital artifacts. They serve as illustrations of how the aesthetics of interaction design can be approached and articulated for further debate as well as for practical application in concrete design situations.

Pliability is the quality of plasticity, of malleability of the digital artifact under the hands of the user. A set of informa-

tion is pliable to the user if it feels like a responsive material, a matter for inquiry that can be manipulated and experienced in a tactile sense. Pliability contributes to a highly involved process of exploration where the loop between senses, thought and action is rapid and physical rather than elaborate and abstract. I make a small, quick move—the material shapes and responds—I notice something new—I make another move—and so on. Ahlberg and Shneiderman were the first to articulate this quality, under the label of “tight coupling” [21].

In the interpretation above, pliability concerns the micro-qualities of interaction, the qualities of the surface. Examples of interaction design aiming at pliability include the influential concept of dynamic queries [22] as well as the more recent interaction technique Sens-A-Patch [23]. Another aspect of pliability has to do with the user's capacity to act freely and shape the material according to the larger situation at hand, such as annotating the margins of a paper form to communicate something outside the rigid boundaries of the form itself. As Henderson and Harris [24] point out, this kind of deep pliability is often unnecessarily lost in the transition from paper to computer systems in, for example, administrative work. It can be straightforwardly observed that many existing administrative systems could be extended to accommodate free-form annotation and the equivalent of sticky notes.

Fluency as an aesthetic quality of digital artifacts is brought to the fore in relation to the increasingly pervasive digital infrastructure. Use is not necessarily a primary activity at the focus of attention; it is not a binary variable in which a digital artifact is either used or not. With ubiquitous and mobile computing, it becomes more of a dance among multiple representations and mediations. Streams of information flow between center and periphery as we move through the shifting environments of everyday life and work. Transitions need to be graceful and nondisruptive.

As a simple but conceptually powerful example of fluency, consider the Hazed Windows concept by interaction design students Trine Freiesleben, Miska Knapke and Henrik Moberg at Malmö University [25]. The idea is simply to offer a more lightweight and transient communication channel, for instance between a little girl and her granny who live in different cities. The girl draws or writes with her finger on a display in her home. Her signs are redrawn at Granny's display but gradually fade away, as the metaphori-

cal “window” fills again with metaphorical condensation. If Granny happens to see the signs before they are gone, she can reply in the same way. A sign on the display disappears completely in a few hours.

The strength of the Hazed Windows concept is not in its focus on lightweight, emotional communication, which has been realized thousands of times before, but rather in the elegant questioning of the hidden core assumptions of computer-mediated communication. When we deal with e-mail, we devote our full attention to it. We expect messages to persist until we choose to file or delete them. In short, we approach the communication situation as a binary task. The Hazed Window hints at the possibility of a new middle ground, a new approach to computer-mediated communication in text and pictures that exhibits a much greater degree of fluency.

Seductivity refers to the captivating qualities of a digital artifact. Following the seminal analysis by Khaslavsky and Shedroff [26], seduction is described analytically as a process of enticement, relationship and fulfillment. Enticement concerns grabbing attention and making an emotional promise. The subsequent relationship is based on making progress with small fulfillments and more promises, possibly lasting for a long time. The fulfillment, or ending, involves making good on the final promises and ending the experience in a positive and memorable way.

It should be clear from the description above that seductivity, in the sense used here, is a quality that crucially depends on the dynamic gestalt, the temporal qualities of the interaction. It has nothing to do with sexually explicit pictures on the front page of a web site. The example offered by Khaslavsky and Shedroff is the Visual Thesaurus [27] by Plumb Design. It is a web application that duplicates the contents of a traditional thesaurus but takes on entirely new qualities by virtue of its interactive properties. The user explores words, their synonyms and eventually the transient nature of language itself by navigating a beautifully animated network of words and their interrelations.

Khaslavsky and Shedroff argue that the Visual Thesaurus is seductive in the sense that it offers surprising novelty, goes beyond obvious needs and expectations and creates an emotional response due to its visual and interactional beauty (enticement). It connects to personal goals through the fascination of words and concepts and promises to fulfill those

goals (relationship). The casual viewer may discover deeper meanings in looking up a word by apprehending the multidimensional and dynamic relationships between concepts (fulfillment).

An Analysis of Perspectives

As suggested in the title of this paper, the reader is meant to come away with the differing views of four authors. What are the connections, and is there a synthesis among the varying views? What was learned during the workshop? Starting with the second question, we learned that the terms *aesthetics* and *computing* are complex and that aesthetic computing is not merely the straightforward application of one field to another but also a self-reflective activity. By juxtaposing two words in a new way, we suggest that the art, design and computing communities must reevaluate the very meaning of the words *aesthetics* and *computing* and not only what it means for them to be combined as a phrase.

To summarize the key perspectives already presented:

- Fishwick stresses the use of aesthetic style and genre within traditional notations for computing artifacts. Aesthetics in both computing and mathematics can be pushed in the direction of artistic styles to augment cognitive aesthetics.
- Diehl discusses the use of visualization and a specific type of aesthetic for software design and execution diagrams. There is already a great deal of research on applying visualization, and one particular sort of aesthetic (optimality), to software design.
- Prophet defines a new type of team collaboration, a kind of “aesthetic computing in action,” in which trans-

disciplinarity leads to new collaborations, influencing the computing process. Applying aesthetics to computing is a matter of process and human integration, not just a static result from the application.

- Löwgren broadens the traditional view of usability to include the aesthetic qualities of the digital design material. As a community, computing professionals must balance cognitive and sensory qualities in design.

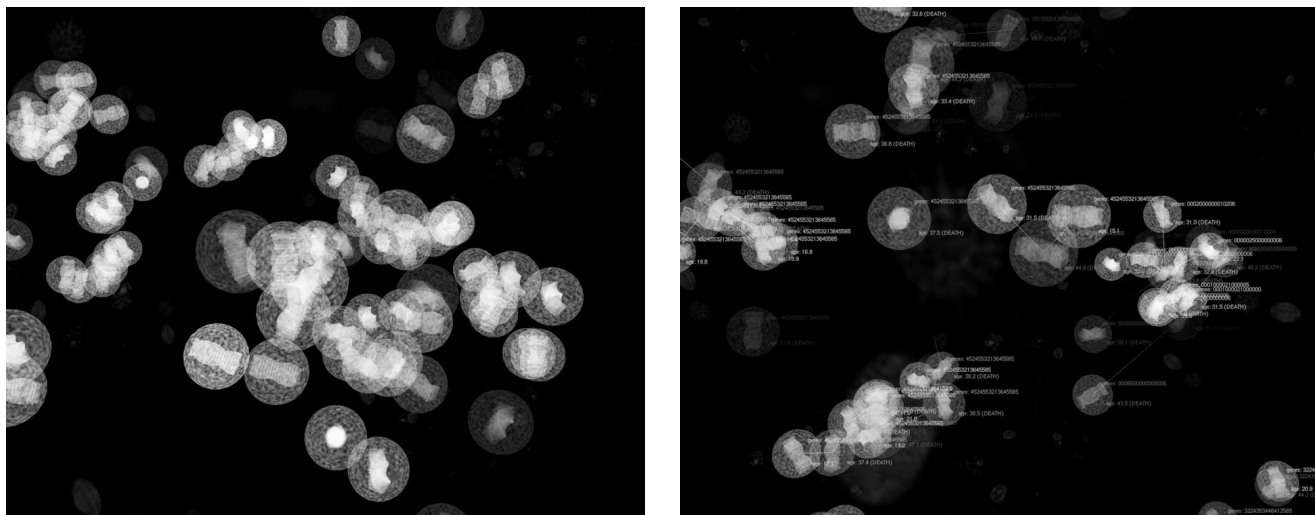
For the most part, there is general agreement among the authors that each section corresponds to a separate axis upon which aesthetics can be seen to apply to computing; however, there is one significant issue raised by Fishwick and Diehl that deserves explanation. Diehl discusses the state of the art in software visualization and emphasizes the current way in which software designers generally view aesthetics—as optimality considerations inherent within the visual design. For example, an aesthetic can be defined to minimize the number of crossings in a 2D graph, or to ensure that all lines leading into a node are equidistant. Diehl also quotes Brooks’s idea that “software is invisible.” Diehl’s views on characteristics of design accurately reflect how the vast majority of computing professionals view aesthetics and computing artifacts—as primarily mental constructs and, if not mental, focused on formal qualities (“formal” is used here as in the artist’s lexicon).

Fishwick takes issue with confining the role of aesthetics to such a narrow interpretation and suggests that future representations for software models should capture more personalized and stylistic forms, given that new technologies have increased our ability to afford 3D

representations without sacrificing abstraction. Mathematics and software are concerned not with visual abstraction but with conceptual abstraction. For example, a stack is a common abstract data structure used in computing. That a stack can be represented in 3D using an aesthetic form that is meaningful to a group of users (that is, a culture) does not defeat the conceptual abstraction. Rather, the 3D concretization of the stack brings to the surface the inherent metaphor that led to 2D diagrammatic stack representations in the first place; the concept of a stack is rife with metaphor—using artistic styles and familiar objects brings these metaphors to life within the same abstract conceptual sphere of knowledge.

The key conflict is cultural—scientists have been trained with visually abstract representations for the sake of material and labor economy. With the advent of new display technologies, it is no longer inconvenient to form novel 3D representations while simultaneously preserving abstraction. There is a need for computer scientists to embrace a wider definition of aesthetics [28], one influenced substantially by the arts. This cultural transition may be difficult but can be gradually facilitated by the kind of evolutionary collaboration described by Prophet, in which artists and scientists work closely together. In such a fashion, all parties become accustomed to a broader definition of aesthetics. The need for this broader definition is reinforced by Löwgren’s discussion of interaction qualities, as well as recent movements in the design community [29,30], where emotion, style and visual appeal are found to be equally as important as function. There are additional re-

Fig. 7. Jane Prophet, *Cell 3D Alive* representations (a, left; b, right). (© Jane Prophet. Rendering design by Rob Saunders.)



sults in the foundational cognitive role of metaphor [31]. If richer designs foster a metaphorical understanding, then such designs ought to be encouraged and investigated. We further suggest that what is relevant for product design is also relevant for the design of more abstract computer programs, data structures and models. Both types of designs reflect a necessary mixture and balance of form and function.

Certainly, optimal designs for achieving a goal or satisfying a particular purpose are still useful when considering the problem of design from the perspective of performance. However, this functional bias of design must be carefully balanced with what one learns by exploring aesthetics in the arts: Design is also about user choice, culture and the freedom to incorporate arbitrary interactive 2D or 3D icons to represent formal structure. Asking, for example, whether it is more appropriate to use a square versus a circle in a design presupposes that there exists an optimal design condition, when in many instances, the design condition is strongly oriented toward user preference and not performance. Designs must be useful, yet this does not imply universal usability for an objective purpose, but instead utility for a segment of the population. This type of free-form design is common in automobiles, architecture and kitchen products; they all must both be usable and contain aesthetic qualities for whoever purchases them. We contend that software and formal structures used in computing are no different in this regard. Representation, in general, has always been driven by the state of technology and the economy. When all one has is clay, then clay marks must be used for representation. As our display and interaction technologies evolve, our representations can also advance, not to replace older representations, but to augment them in the same way that new media have augmented, or remediated [32], old media.

CONCLUSIONS

One of the key tensions in our discussion is the interplay between that which artists produce versus that which computer scientists produce. Artists have an agenda based on a wide variety of styles and aesthetics, from formalism and cultural exploration to capturing social, political or economic aspects of phenomena. Computer scientists primarily are after high-utility artifacts; if something is useless, most mathematicians and computing

professionals will tend to shy away from it. However, there is a line that stretches from “no use” to “full use,” if there can be such unambiguously defined things. To see this, we need to step back to the definition of aesthetic computing: the application of art practice and theory to computing.

There is no reason why this application must be targeted on artifacts of high general utility. By exploring the boundaries and interstices of the “use range,” we think that we can enhance both art and computing. Artists will become more familiar with elements of computing, such as data structures, programs and architectures, and even the core mathematical structures upon which computing is founded. For example, what artists produce [33,34] reflects a computing essence, whether or not the result is of immediately obvious utility. To the extent that a piece makes one reflect, it is useful in exploration, creativity and education, and so even the term “use” or “usability” becomes suspect. Thus, aesthetic computing for the artist can range from software art (a fairly new movement defined by artists producing their own programs or languages) to representational art in which the computing element becomes the subject of the art, rather than the material for the art as in software art. It is this ability to weave through the webs of utility and computing that makes aesthetic computing a unique enterprise.

For the computer scientist, what the artists produce will be fertile ground for representation, interaction design and human-computer interfaces in general. While through software art some artists are becoming computer scientists of a sort, we can also observe the converse situation as some computer scientists become artists.

On a more general level, the encounter between art and computing may be studied in terms of disciplinary relations. Jantsch [35], the originator of such concepts as multi- and interdisciplinarity, views disciplinary integration as an evolutionary hierarchy. If a traditional disciplinary approach is specialization in isolation, then multidisciplinary simply refers to adding different disciplines without any direct cooperation between them.

The next step up the evolutionary ladder is crossdisciplinarity, where one discipline supports the other within the other’s own discipline. Our discussion above, and the identification of fields such as software art and aesthetic computing, might suggest that this is more or

less our current state of progress. An illustrative example is our introduction of the dimension of utility as one of the ways in which the fundamental values of the disciplines of art and computing differ.

The most advanced integrative stage, according to Jantsch, is interdisciplinarity. It involves direct cooperation in both directions where the outcomes typically could not be achieved entirely without any of the disciplines involved. It also entails the formation of new concepts, practices and values that transgress the traditional boundaries of these disciplines. It is our firm conviction that the encounter between art and computing holds the potential for interdisciplinarity in this strong sense, and our ongoing dialogue might represent a step in that direction.

Acknowledgments

The authors are indebted to Dagstuhl, Germany, where many of the ideas in this paper flourished in a synergistic weeklong session. We also would like to acknowledge the Leonardo organization for their co-sponsorship of the event. Paul Fishwick would like to thank his graduate students in the RUBE Project (Minho Park, Jinho Lee and Hyunju Shim), the National Science Foundation, working via grant EIA-0119532, and the Air Force Research Laboratory, working via grant F30602-01-1-05920119532. Jane Prophet’s contribution is part of the *Cell* collaboration and has been written following discussion with Mark d’Inverno, Peter Ride, Rob Saunders, Neil Theise and Katrina Jungnickel. *Cell* research has been conducted with awards from the Wellcome Trust sciart, Shinkansen Future Physical “BioTech” and the Quintin Hogg Trust.

References

1. Aesthetic Computing Dagstuhl Seminar 02291, 15–19 July 2002, <<http://www.dagstuhl.de/02291>>.
2. The Dagstuhl Aesthetic Computing Workshop, “Aesthetic Computing Manifesto,” *Leonardo* **36**, No. 4 (2003) p. 255.
3. M. Kelley, ed., *Encyclopedia of Aesthetics*, 4 vols. (Oxford, U.K.: Oxford Univ. Press, 1998).
4. J. Hadamard, *The Psychology of Invention in the Mathematical Field* (Princeton, NJ: Princeton Univ. Press, 1945).
5. D. Knuth, *The METAFONT Book* (Reading, MA: Addison-Wesley, 1986).
6. E. Gamma et al., *Design Patterns: Elements of Reusable Object-Oriented Software* (Reading, MA: Addison-Wesley, 1995).
7. S. Card, J. Mackinlay and B. Shneiderman, eds., *Readings in Information Visualization: Using Vision to Think* (San Francisco, CA: Morgan Kaufmann, 1999).
8. S. Diehl, ed., *Software Visualization* (New York: Springer Verlag, 2001).
9. J. Stasko et al., eds., *Software Visualization: Programming as a Multimedia Experience* (Cambridge, MA: MIT Press, 1998).
10. T. Kim, J. Lee and P. Fishwick, “A Two-Stage Modeling and Simulation Process for Web-Based Modeling and Simulation,” *ACM Transactions on Modeling and Simulation* **12**, No. 3, 230–248 (July 2002).
11. P. Fishwick, “Aesthetic Programming: Crafting

- Personalized Software," *Leonardo* 35, No. 4, 383–390 (2002).
12. J. Hopkins and P. Fishwick, "Synthetic Human Agents for Modeling and Simulation," *Proceedings of the IEEE* 89, No. 2, 131–147 (February 2001).
13. J. Hopkins and P. Fishwick, "Exploiting the Agent-Based Metaphor in Software Visualization using the RUBE System," *Journal of Visual Languages and Computing* 14, No. 1, 97–117 (February 2003).
14. P. Fishwick, *Simulation Model Design and Execution: Building Digital Worlds* (Englewood Cliffs, NJ: Prentice-Hall, 1995).
15. F.P. Brooks, "No Silver Bullet," *IEEE Computer* 20, No. 4, 10–19 (April 1987).
16. aiCall StackAnalyzer, AbsInt GmbH <<http://www.aicall.de>>.
17. T. Zimmermann, S. Diehl and A. Zeller, "How History Justifies System Architecture," *Proceedings of International Workshop on Principles of Software Evolution IWPSE 2003*, Helsinki, Finland, 1–2 September 2003 (IEEE Computer Society Press, 2003).
18. D. Goodsell, <<http://www.scripps.edu/pub/goodsell/>>.
19. Sol LeWitt, "Paragraphs on Conceptual Art," *Artforum* 5, No. 10, 79–83 (June 1967).
20. J. Kosuth, in G. Guercio, ed., *Art after Philosophy and After: Collected Writing 1966–1990* (Cambridge, MA: MIT Press, 1991).
21. C. Ahlberg and B. Shneiderman, "Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays," *Human Factors in Computing Systems* (CHI '94 Proceedings) (New York: ACM Press, 1994) pp. 313–317.
22. C. Ahlberg, C. Williamson and B. Shneiderman, *Dynamic Queries for Information Exploration: An Implementation and Evaluation. Human Factors in Computing Systems* (CHI '92 Proceedings) (New York: ACM Press, 1992) pp. 619–626.
23. J. Löwgren, "Sens-A-Patch: Interactive Visualization of Label Spaces," in E. Banissi et al., eds., *Proceedings of the Fifth International Conference on Information Visualization (IV2001)* (Los Alamitos, CA: IEEE Computer Society, 2001) pp. 7–12. Web: <<http://webzone.k3.mah.se/k3jolo/Sens-A-Patch>>.
24. A. Henderson and J. Harris, "Beyond Formalism: The Art and Science of Designing Pliant Systems," in K. Kaasgard, ed., *Software Design and Usability* (Copenhagen, Denmark: CBS Press, 2000).
25. T. Freiesleben, M. Knapke and H. Moberg, Hazed Windows, 2001 <http://webzone.k3.mah.se/kid01016/projekt_forskningstema.htm>.
26. J. Khaslavsky and N. Shedroff, "Understanding the Seductive Experience," *Communications of the ACM* 42, No. 5, 45–49 (1999).
27. Visual Thesaurus: <<http://www.plumbdesign.com/thesaurus>>.
28. D. Norman, *Emotional Design: Why We Love (or Hate) Everyday Things* (New York: Basic Books, 2003).
29. Norman [28].
30. P. Jordan, *Designing Pleasurable Products: An Introduction to the New Human Factors* (Boca Raton, FL: CRC Press, 2000).
31. G. Lakoff and M. Johnson, *Philosophy in the Flesh: The Embodied Mind and Its Challenge to Western Thought* (New York: Basic Books, 1999).
32. J. Bolter and R. Grusin, *Remediation: Understanding New Media* (Cambridge, MA: MIT Press, 1999).
33. J. Maeda, *Design by Numbers* (Cambridge, MA: MIT Press, 1999).
34. Processing Language: <<http://www.processing.org>>.
35. E. Jantsch, *Technological Planning and Social Futures* (London: Associated Business Programmes Ltd, 1972).

Manuscript received 15 October 2003.

Corresponding author Paul Fishwick is with the Computer and Information Science Department at the University of Florida.

Stephan Diehl is with the Computer Science Department at the Catholic University Eichstätt, Germany.

Jane Prophet is Director of the Centre for Arts Research Technology and Education (CARTE), University of Westminster, London.

Jonas Löwgren is with the School of Arts and Communication, Malmö University, Malmö, Sweden.

CALL FOR PAPERS

Live Art and Science on the Internet

The Internet has become a venue and medium for art as a means to broadcast ideas to a worldwide audience. *Leonardo* and Guest Editor Martha Wilson seek texts on the subject of "Live Art and Science on the Internet" for a series of special sections in the international journal *Leonardo*, both in print and online.

As artists and others produce live art on the Internet, liveness, presence, mediatization, online activism, surveillance and identity/gender, among other issues, are being explored. We seek texts documenting such work, as well as texts on the history of this field of practice and on the vocabulary being used to describe it. We also seek texts from scientists who have used the Internet to conduct science investigations live on-line.

Guest Editor Martha Wilson and her peer review committee seek *Statements* (500 words plus one image describing one work), *Notes* (2,500 words plus 6 images describing a body of work), *Galleries* (750-word curator's introduction plus up to 10 images by individual artists, each with a 200-word caption) and *Articles* (5,000 words plus 12 images). Texts describing the work of a living artist or scientist must be written by the artist or scientist him/herself, with a co-author if necessary.

This call for papers is open for 2004, 2005 and 2006.

Please send an initial statement of interest with a brief explanation of your project to Martha Wilson: <Leonardo@franklinfurnace.org>. For author guidelines, follow the link "Info for Authors" on *Leonardo On-Line* <www.leonardo.info>.