



System Models for Digital Performance

by reed kram

B.S. Electrical Engineering
Duke University
June 1993

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences
at the
Massachusetts Institute of Technology
June 1998

Copyright Massachusetts Institute of Technology, 1998
All rights reserved

author

reed kram

program in media arts and sciences
May 8, 1998

certified by

john maeda

assistant professor of design and computation
program in media arts and sciences, thesis supervisor

stephen a. benton

chair, departmental committee on graduate students
program in media arts and sciences

System Models for Digital Performance

by Reed Kram

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on May 8, 1998
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

This thesis defines a new dynamic pictographic visual language for describing interactive digital performance systems. We review the origins of system model languages, machine performance spaces, reconfigurable performance spaces, and human-computer interfaces for performance. We discuss a number of novel systems for digital performance created by the author. We then introduce the new, symbolic digital performance language in the form of a living diagrammatic environment called *Modulator*. Finally, we use the *Modulator* system to categorize digital performance spaces, both by the author and by others, and as a framework to examine and explain the creation of computational performance spaces.

Thesis Supervisor: John Maeda

Title: Assistant Professor of Design and Computation

This work was supported by the Things That Think consortia and Italia Telecom fellow program



System Models for Digital Performance

reed kram

the following people served as readers for this thesis

reader

paul haeberli

principal scientist

silicon graphics incorporated

reader

henry jenkins

director of film and media studies

assoc. professor of literature, mit

Acknowledgments

Thanks to John, Kathryn, Elbert, Peter, Thais, Matt, Tom, Chloe, Dave, Rich, Peter, Kris, Andy, John, Scott, Paul, Jared, Elise, Jessica, Matt, Rob, Brygg, Bryant, Sammy, all the X-Factors (of which I am one), Paul, Henry, George, Nicholas, Scraps, Ted, Sweet Creme, Big Fred, Ultra, Love Giant, David, Trish, Poodie, Toots, and Annika and Annorak most of all!

Contents

1 Introduction	
1.1 Motivation	7
<i>Media Language</i>	
<i>Media Structure</i>	
<i>Media Techniques</i>	
<i>Accomplishments</i>	
<i>We Would Like To Make the Performing Machine</i>	
1.2 Why Is This Thesis Being Done At The Media Lab?	17
1.3 Thesis Structure	18
2 Background & Theory	
2.1 System Model Languages	19
<i>Origins</i>	
<i>Artists' Diagrams</i>	
<i>Human-Computer Diagrammatics</i>	
<i>Symbolic "Programming"</i>	
2.2 Machine Performance Spaces	38
2.3 Reconfigurable Performance Spaces	43
2.4 Human-Computer Interfaces for Performance	46
<i>Computational Art</i>	
<i>Digital Tools</i>	
<i>Digital Games</i>	
3 Digital Performance Systems	
3.1 Dimension7	61
3.2 Plastic Jazz	63
<i>Physical Interface Design</i>	
<i>Visual Interface Design</i>	
3.3 Pliant Architecture	71
3.4 Dynamic3	74
3.5 Proposed Interface 1: MidiVis	77
3.6 Proposed Interface 2: Sound City	81
3.7 Transducer	83
<i>Data Abstraction</i>	

4 Modulator Language	
4.1 Hypotheses	89
4.2 Static Representation	91
4.3 Dynamic Representation	96
<i>Sensory Communication</i>	
<i>Modulator Interface Design</i>	
<i>Feedback</i>	
<i>Viewpoint</i>	
<i>Native vs. Transformative Communication</i>	
<i>Communication Channels</i>	
<i>Form of Interaction Space</i>	
5 Analysis	
5.1 Applied	114
5.2 Sensory Throughput	116
5.3 The Performing Machine	119
6 Conclusion	
<i>New Hope for Digital Design</i>	
7 Bibliography	

Chapter 1: Introduction

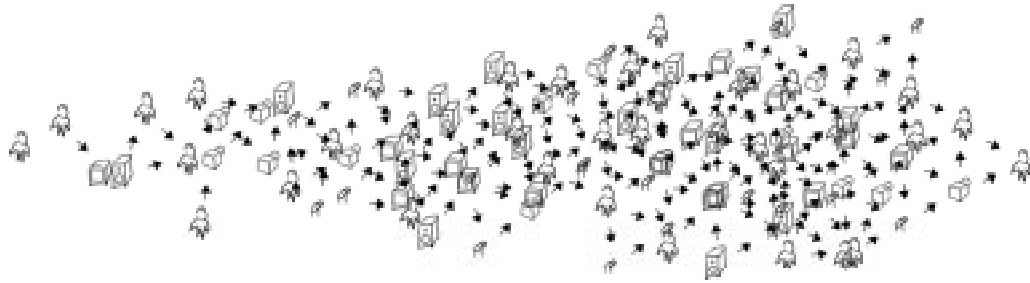


Figure 1.1: A study in overwhelming interface complexity

1.1 Motivation

Media Language

Moholy-Nagy: “The film will outmode painting.”

Nam June Paik: “The cathode-ray tube will replace canvas.”
[from Viola, 1995]

What then for computers? How does computation extend this lineage as defined by the modern artist? It is interesting that despite the ever-prevalent VGA monitor, the computer is not equivalent to the cathode ray tube. Computation is a separate realm which may or may not be connected to a monitor. Its aesthetics may or may not be visual.

The digital medium consists of bits. Bits can be used to represent all sorts of information, from numbers to virtual life. Therefore, designing in this medium should be viewed in a light beyond the manipulation of photographs or making posters, but as having the chance to design at the same level of abstraction as our own imaginations. Of course, this all presents a tremendous challenge to our perceptual and creative abilities and raises the question, “Is the imaginative space of the digital medium greater than the one in our heads?” We don’t know yet.
[Maeda from Aesthetic & Computation Group, 1997]

That said, it matters not that a computer be binary or even electrical. If a computer were made entirely of analog mechanical parts, it would still add, subtract, multiply, divide. Amazing the power, mythology, and complexity born of such mind-numbingly simple building blocks. The computer may be the first of a new breed of media: biological, chemical, physical, wherein the tool and medium are one in the same.

What is a computer? A process perhaps, a connection of communicating elements. Given this definition, then static schematics or diagrams are more computer art than any linear video art.

Certainly one curious example of this phenomenon is the association of the term “media” itself to computation (as in “Media Lab”). Somehow “media” has been adopted to describe the computational media, while in fact “media” has existed since the dawn of time and mechanized computation has not.

“Multimedia” must then be a misnomer on top of a misnomer. It is a combination of two very disparate terms: multidisciplinary and mixed-media. The computational media allows us to explore multidisciplinary work in unique and powerful ways: art, architecture, graphic design, filmmaking, engineering, logic, linguistics in a single medium as never before. We can cross-reference and design between and for these various disciplines dynamically.

Yet this process within a computer is in fact not mixed-media. It is within the single computational media. Too often when we do mix these disciplines in computers, we focus on the most common media previously associated with the discipline: filmmaking = a linear video, graphic design = a rectangular image, architecture = a fixed plan + elevation of a form, linguistics = text parsing, etc. The work created on the computer associated with these disciplines mimics the old media. This is what we get with “multimedia.” As a misnomer, it gives us a bad name: as if we (those that do multidisciplinary work on computers) have something to hide, for very often works that are “multimedia” are in fact the same or extremely similar to works done with other analog technologies that may in fact be better suited to the task.

For the most part this thesis will focus on the mechanics or form of the computer-human interface as opposed to the content of said interactions. Are the actions explicitly “choreographed” or interactive or do they respond directly to

input information? Obviously, a combination of all of these, but this alone could be an entirely separate thesis or six: when is choreography important, when is mediated interaction, etc. How can one “design” something interactive or immediately responsive to a whole range of input information if one doesn't know exactly what information will come in? These issues are all addressed secondarily through the given implemented examples.

Media Structure

“Our present media structure is hard-wired into a very narrow range of forms. We are deeply enculturated into these forms and all-too-often accept them as “only” and “natural.” How often, for example, do we see images that are not flat and rectangular? (How many pictures of yourself have you seen that were not flat and rectangular)” [Naimark, 1981] To this I would add, “how many images of a computer have you seen that do not include a mouse and keyboard.”

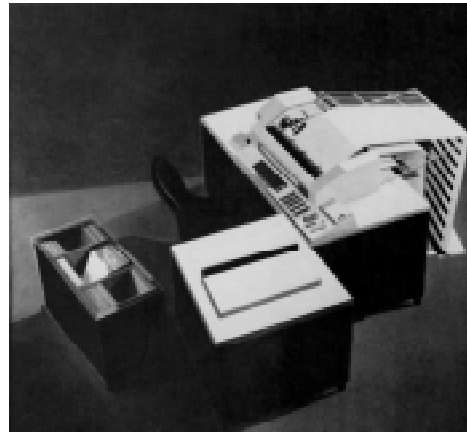


Figure 1.2: Lowell Nesbitt, *IBM 6400*, 1965

We now have the ability to control multiple video projections, audio sources, even haptic devices simultaneously in realtime, something Michael Naimark could only point towards in 1981. Yet our concepts of the expressive possibilities for these new devices remains trapped in accepted status-quo forms which have evolved since Naimark's time.

As Bill Buxton pointed out in his speech at MIT earlier this year, “When I ask people to draw a computer, they draw something like this, (Figure 1.3), almost 100% of the time. I have run this experiment thousands of times and we get the same result almost without fail.” [Buxton, 1998]

To a large degree, our definition of a computer is based entirely on the mechanisms with which we interact. The actual computer itself, the anonymous rectangular “black-box,” seen in Buxton’s representation has almost no importance in the drawing. For the most part, the user would not mind if this physical component disappeared. In fact, thanks to numerous unpleasant experiences with hardware installation into said black box, the user would probably eagerly welcome its disappearance.



Figure 1.3: The personal computer, circa 1998

This work is part of the larger TTT or Things That Think research initiative at the Media Laboratory at MIT. The TTT initiative asks one to think of digital components separated from their traditional housings. Typical TTT projects include novel display surfaces [Campbell, 1996], physical input apparatuses [Smith, 1997], and remote haptic I/O devices [Brave, 1997]. This research attempts to create a forum for examining the design of systems informed by this larger body of TTT research. The proposed systems should be as novel as the input and output devices which motivate this research.

Media Techniques

Visual communication design has not yet advanced in the digital age. At one point graphic designers employed typography and images to communicate using fixed posters and signage. The term “graphic design” has lost its original meaning. We now have the ability to communicate interactively with constantly shifting pixels and audio speakers. The new digital design is a simultaneous experience of vision and hearing and communication. To create a new understanding of light, space and motion on the computer screen one must depart from pre-existing tools to the more rigorous process of the computer's native language, computation. In order to advance, we must find harmony between the disciplines of visual design and computer science. This requires new languages to evolve to explain and express the new design process. This thesis proposes one such language.

Thus far, visual design on computer screens has sought to mimic existing graphic technologies. We present text on a computer monitor as if it were paper. We present movies on a monitor as if it were a television.

History has shown that work in emerging media first adopts the concepts and content of preexisting mediums before entirely new work based in and of the new medium arise. MIT's David Thorburn calls this phenomenon the "horseless buggy principle." In essence, the horseless buggy principle reminds us that there was no reason that the first motorized vehicles should look like horse-drawn carriages. The engineers of these early cars adopted forms from transportation that had come before as opposed to designing entirely new vehicles based around the design constraints of the new technology.

Thorburn points out that time and again, as media arise the first step in their acceptance is this assimilation of previous techniques. The first book printed on Gutenberg's printing press was a bible. Cervantes' *Don Quixote* defined a new literary tradition, the novel, which existed only because of the printing press and could not have come before it. The first films were essentially theatre plays shot as though one were a spectator in the audience. The computational medium is still in the early, adoptive phase of evolution. [Thorburn, 1998]

Our most prevalent methods for interacting with computation are not so different from the typewriter and television. The introduction of the mouse was likely a step in the right direction, as a device designed specifically for the computational medium. Yet the mouse has been widely available for almost 20 years and we have yet to see any other great interface devices arise in that time. Similarly, the applications running on our computers are based largely on concepts borrowed from other mediums and have changed little in 20 years.

The computerized tools for designers ease the creation of print and video work. But the design work being made for computer screens is not inherently different in presentation than work created for print or video. In order to innovate a new method of design and interaction for the screen, the designer must begin to move away from the numerous commercially developed tools that facilitate the authoring of pre-constructed visual elements and pre-determined interactions. The computer designer can only be at one with his medium through building custom programs in harmony with the medium.

The designer must be in full command of his medium. The computer, as a computational medium, can only be truly tamed through the use of computer science. However, the field of computer science has, until now, found little use for graphic design or designers. The field of computer science judges its work in terms of order-n, quantitative technological advancements. This same methodology infiltrates the field of human-computer interaction (as a branch of computer science) in the form of user-studies. Each successive, iterative user-interface is subjected to “human-factors” evaluations with x number of test users. How many great plays or works of art would survive the ubiquitous “user study” at the time of their creation? Thus the field of human-computer interaction remains almost completely unaware of the wealth of human communication inherent to human visual communication design for the simple reason that visual communication design can only effectively be judged qualitatively.

Current movements linking computer science and design generally emphasize a collaboration between experts in these respective fields. In her book directed to the topic of Computers as Theatre, Brenda Laurel reinforces the common myth that the design of human-computer interfaces necessitates the “cooperation” of artists and engineers:

Human-computer activity is like drama in the sense that the primary designer (or playwright) is not the only human source of artistry in the completed whole. In the case of theatre, the director, actors, designers, and technicians who are involved in rendering a performance all make contributions that require artistry. In human-computer activity, there may be a legion of programmers who have designed and architected programs on which a given activity depends, graphic designers who created images and animation, wordsmiths who authored text (or text-generated algorithms), and so on. [Laurel, 1991]

The belief that the creation of a human-computer interface requires a legion of authors is simply untrue. Yet time and again the cycle is repeated: a programmer is asked to fabricate a designer’s vision as closely as possible. This is equivalent to asking a painter to paint a photograph with the utmost accuracy. In each case the result is diluted. It will never reach the heights and the clarity of a master at work in his or her native medium.

The members of our group have worked individually and in close communication with Professor Maeda. We have established a new design process that emphasizes the moment of interaction as a performance. This is the process of simultaneous computational design wherein the craft of computation and a sensory result become fused into a single movement. The Aesthetics & Computation Group design process embodies the age-old feedback between a craftsman and his or her chosen material. This method allows for unhindered creativity in the computational realm. We have made work which moves beyond pointing and clicking on icons to the creation of novel physical interfaces and visual explorations which treat the computer projection as a malleable grid of light. Using the fastest and most complex graphical systems available, we have created dynamic sculptures, pliant architectures, and human interfaces which attempt to predict the next generation of digital design.

Yet how does one describe this new direction? What methods exist to explain methodology, define approaches? Painfully few. We describe computational approaches with logic diagrams and pseudo-code. We explain graphic design with grids and typographic terms.

This thesis aims to define a new language for this new design of digital systems. Similar to Gerstner's "design programmes," this thesis is about "inventing rules of arrangement." [Gerstner, 1964] These combined rules (or language) should serve as a polemic device, as a method for discourse on the subject of aesthetics & computation, and as a technique for future researchers in the field to organize and examine their design process.

Accomplishments

Mathematic and algorithmic design can exist in the same breath as artistic design. In the creation of interactive systems, one must develop mathematical models in tandem with the visual/auditory/sensory presentation. These models must remain parameterized, where parameterized is defined as a methodology by which one is able to alter specific predesigned nodes such that an equation can be manipulated to produce expected results. In this way one is able to lessen the effect of the random, the unexpected, or the unexplained.

Interactive particle physics provides characteristics uniquely advantageous for the design and control of digital performance systems. In this thesis we have advanced a new conception of interactive physics in the interface. The theoretical underpinnings for this direction lie in algorithms commonly used in video games and simulations. Here we adapt these techniques to forward a distinct stylistic feel common to our performance systems.

In addition, we present a novel physical interface, specifically designed for the protean control of three-dimensional forms. Recently, the human-computer interface design community has called for research focussing on physical manifestations for interacting with computers. Our two-handed input device remains consistent in form and function with the computational visual interfaces it controls.

Lastly, we employ the visual language of mathematics to engage a new vision for design in computation. Similar to Valentino Braitenberg's Vehicles, we use straightforward visual diagrams to suggest elegant directions for future design. Though in contrast to Braitenberg's pictures, our Modulator diagrams exist as functioning computational systems: spatial, analytical, alive.

We Would Like To Make the Performing Machine

In his book, “Vision In Motion,” Moholy-Nagy eloquently describes the traces of action:

The sea rolls against a sandy beach; the waves subtly corrugate the sand.
A painted wall cracks; the surface becomes a web of fine lines.
A car moves in the snow; the tires leave deep tracks.
Rope falls; it lies in smooth curves on the ground.
A board is cut; it shows the marks of the saw.
[Moholy-Nagy, 1947]

In each of these cases we are left with the result of a motion. The quality of the mechanism or actor and its interaction with a medium has left a trace. Yet we would know much more about the actor by examining its methodology over time for creating that trace. Imagine we knew little of cars and were presented with only the traces of tire tracks in snow. How could the cars have created these patterns? What mysterious forces caused the curved, black marks? By replaying and reexamining, exploring the creation in time, we would be able to better understand, relate to the car's processes, inner workings, and emotive quality.

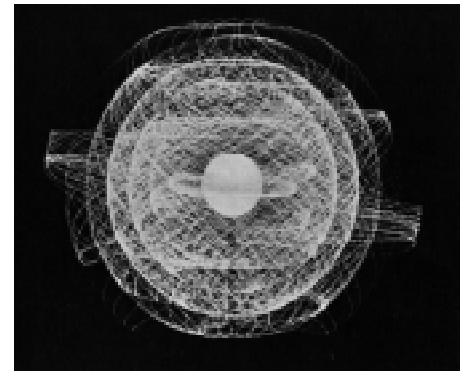


Figure 1.4: Time-lapse photo of a uranium atom

The computational medium provides a hitherto unknown ability to record, to replay, and most importantly, to manipulate time. By focussing on selected moments in time and remolding them, we examine and dissect the expressiveness of our actions. But don't all computer interfaces deal with time? Yes, they do. The use of time generally involves the user waiting for the computer to finish doing something and little else. Some tools currently exist to capture actions or strokes. The GUI pointer provides instantaneous capture. Adobe Illustrator, Photoshop, and After-Effects provide varying levels of capture in time. But these tools focus on the result of the stroke rather than the stroke itself as primary. They cannot modulate the expression of the entry of information over time. Current computer applications are trapped in a narrow realm of interface actions guided by business applications, even those designed to create “computer art.”

The process of the action contains unique information of our emotional, expressive state at that moment. The reinterpretation of this action by a performative computer system initiates a dialogue between the designer and the participant. Recent linguistic research tells us that human dialogue is not mere turn-taking in which one person says something, the other thinks about it and then says something, and so on. Dialogue involves coordination of content and process, synchronization of entrances and exits, adjustments for tempo and dynamics. The two participants must update or revise their common ground at every instant. [Clark and Brennan from Laurel, 1993]

A “digital performance” is a live, interactive event emphasizing the manipulation of time. Here the emphasis is on “the moment” (an activity which occurs at a certain time and for a certain duration). The single greatest ability of the computer for creation is its facility to affect and reinterpret time. Yet very little has been done to take advantage of this ability. The computer screen flashes at least 60 times a second, the computer audio out at 44,000. Like a television, the computer screen is a manipulable grid of light. Unlike a television, this entire grid can be controlled at any moment in time interactively. A television show may be considered a “performance,” though the event or exact moment at which the performance occurs holds little importance. Any performance can be taped and watched in exactly the same way countless times. In contrast an event on a computer screen is seldom thought of as a performance, though for interactive systems the exact moment of activity or participation is of primary importance.

A digital performance needs an audience. John Cage may tell us that “theatre is anything that engages the eye and the ear.” [Cage, 1965] We must be a bit more specific. A digital performance needs a participant or participants. The computer was not the first interactive medium, but it may be the best.

Thus any interactive system that emphasizes the process of action or the event of interaction with a human participant may be thought of as a digital performance, though it may only have an audience of one. Our concept of the performing machine is not created solely with digital components. Our performing machines include humans as parts of the systems. The best way of explaining the digital performance is through examples, which is what we show at length later in this document. In this thesis we will examine performance systems

designed for a single participant and develop methods for understanding the complex webs of feedback in the design of systems for multiple interacting participants.

1.2 Why Is This Thesis Being Done At The Media Lab?

A seemingly preposterous disposition fills the vast majority of institutions teaching “new media” design in the 1990’s: they do not build programs. If one is studying furniture design, the student should build at least a single chair: cut the wood, stretch the leather, then sit on the chair and test how it works. The same can be said for the interactive design, media design, what have you: “build your own.” This thesis is the result of one student’s earnest effort to master the computational medium for expression through building systems with an emphasis on both aesthetic and engineering excellence.

To our knowledge the Aesthetics & Computation Group at the MIT Media Lab is the only organization pursuing this direction at an academic institution. Several schools of design have “media” or “interactive” design departments, but these organizations are primarily focussed on commercially available tools for design. Neither the students nor professors possess the computer science skills to open up the medium to interactions beyond the basics provided by these tools. Isolated individuals in the commercial realm demonstrate exceptional computational skill and inspiring work (Paul Haeberli of SGI and Scott Snibbe of Interval as prime examples). But only at MIT are we lucky enough to have a group of individuals extremely versed in the computational medium actively pursuing pure visual design research for the computer itself. We have now begun to reach a kind of “critical mass” for discourse on the topic of design and computation wherein enough projects are in process such that we can start to extract trends and examine the design process itself. This analysis could only occur at the Media Laboratory.

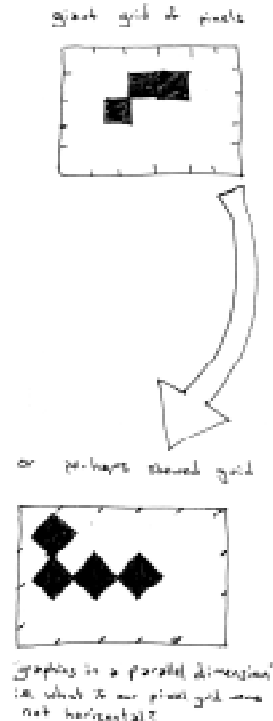
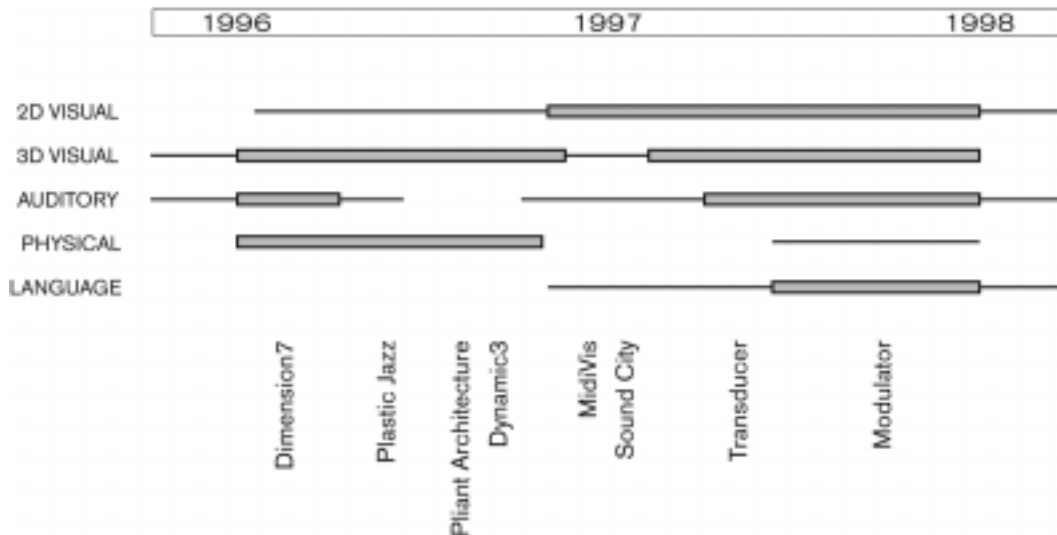


Figure 1.5: Sketch of proposed digital design

1.3 Thesis Structure



This thesis is divided into several sections. Chapter 2 sets the stage for this current work. We outline background media theory defining the context of a digital performance, we study previous examples of pictographic visual languages for defining system models, and we detail historic artistic experiments in which systems of machines and humans perform in various configurations. Chapter 3 examines various systems for digital performance created by the author which led to the Modulator design. Chapter 4 describes the Modulator research language and the functions. In Chapter 5 we use the Modulator system as an underlying approach for the analysis of digital performance models. Chapter 6 concludes with a review of the system and places this work in the larger fields of media study and Aesthetics & Computation.

Figure 1.6: Diagram of the author's works over the past two years. Works spanned 2 and 3-dimensional visual design, auditory, physical, and language constructions. At the base we see the list of works placed at the time at which they occurred.

Chapter 2: Background & Theory

2.1 System Model Languages

Origins

The world, as Norbert Wiener once remarked, may be viewed as a myriad of To Whom It May Concern messages. The significance of this statement becomes apparent when we recognize that everything that exists and happens in the world, every object and event, every plant and animal organism, almost continuously emits its characteristic identifying signal. Thus, the world resounds with these many diverse messages, the cosmic noise, generated by the energy transformation and transmission from each existent and event. [Kepes, 1966]

Biological signals are received, transmitted, amplified, or reduced by each species according to its senses. Man has both wide and limited facilities for accepting and transmitting signals and can learn a great number of signs by which he manages his day-to-day business. Man uses these signs to regulate his functioning and conduct. In addition, man has the unique ability to transform and reinterpret symbols into pictures, sounds, sculptures, and retransmit these symbols for further recognition by other humans. This may also be thought of as “information processing,” a capacity also shared by the man-made creation of computers, which will be discussed later in this document. [Wiener, 1948]

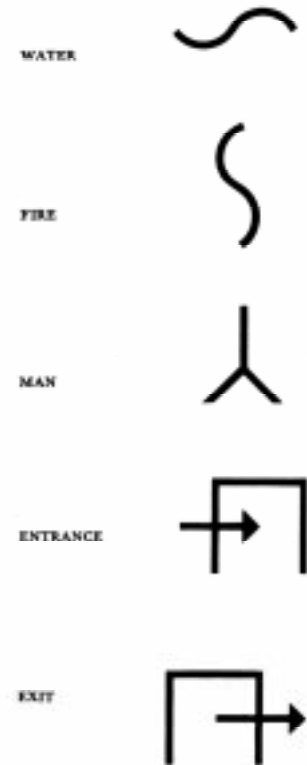


Figure 2.1: Basic symbols

Pictographic visual languages for describing these life-system models appear to have existed as long as drawing itself. From the first cave paintings of neanderthals attacking beasts to electrical engineering logic diagrams humans have long expressed process through drawing.

For humans, the shape of a given symbol may have little to do with the actual appearance of the subject it represents. For example, a sun may not be round and yellow, as demonstrated amply by the word “sun” itself. How then, do we decide the form of a given symbol? There can be no direct answer to this question, only examples.

The form of symbols is, in many ways, dictated by the tools and medium with which the symbols are generated as shown in Figure 2.3. This evolution of Sumerian linguistic symbols is believed to have occurred over the period 4000 to 2000 B.C. As one moves down the rows of the table, each symbol evolves away from the initial, highly representational pictogram. It is easy to see how the constraints imposed by various writing tools affects the forms of these symbols.

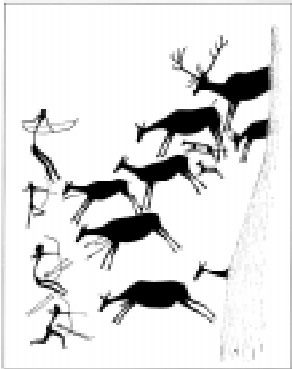


Figure 2.2: Prehistoric hunting scene, Valltorta, Spain

Figure 2.3: Sumerian pictograms



Figure 2.4: Michael Maier's
Nature teaching Nature, 1587

Symbolization does not require connecting lines or arrows to represent communication. A renaissance-era etching illustrates a complex system of religious and societal interplay without any accompanying text (Figure 2.4). [Kepes, 1966]

Graphs are discrete symbolic structures consisting of vertices and edges that connect those vertices. Directed graphs were introduced in the 18th century by the great Swiss mathematician Leonhard Euler. He employed graphs to solve the famous Konigsberg bridge problem. Though graph theory is a very old subject, it has many modern applications. It has found particularly wide use in the description of computational processes. [Rosen, 1995]

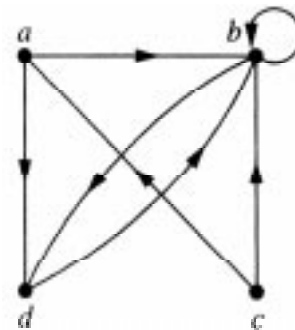


Figure 2.5: A directed graph



Figure 2.6: Marquand's Logic Machine



Figure 2.7: Peirce's letter

The logic diagram representing electrical systems for solving human problems probably dates back to around the 1880s. It begins with a machine built by Princeton professor Allan Marquand for solving problems in formal logic. Marquand's logic machine, through a combination of rods and levers, catgut strings, and spiral springs, was able to display all the valid implications of a simple logical proposition. It took Marquand's former teacher Charles S. Peirce to make the conceptual leap into the electric. In a letter to his former student, Peirce describes how a system of batteries and switches could be arranged to perform significantly more complex problems of formal logic, including algebra and geometry. [Eames, 1973]

Psychologists and neuroscientists found these types of directed diagrams useful for describing cognitive models of the workings of the mind. In Figure 2.8 we see Freud's schematic representation of the primary defense (repression) from his preliminary notebook for *Project for a Scientific Psychology*. [Freud reproduced in Moser, 1996]

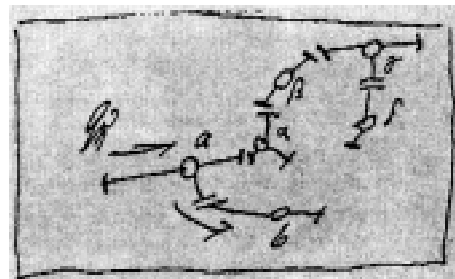


Figure 2.8: Sigmund Freud: Diagram of Repression

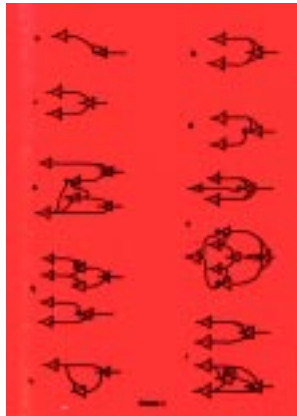


Figure 2.9: McCulloch's neural network notation

In 1945, John von Neumann adopted a notation originally designed to describe the behavior of neural networks by Warren S. McCulloch and Walter Pitts (Figure 2.9) to describe the functioning of his proposed EDVAC machine. Concurrent to the development of this new notation was the remarkable concept of the “stored program,” to be demonstrated for the first time in his novel machine. Two years later, von Neumann created a visual system of boxes connected by arrows as a sample of all the general features of coding for a computer (Figure 2.10). This system is the precursor to today’s flow-diagramming techniques. [Eames, 1973]

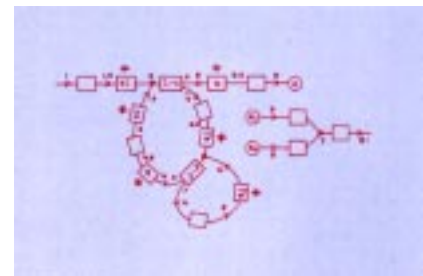


Figure 2.10: von Neumann's “flow diagram”

Artists' Diagrams

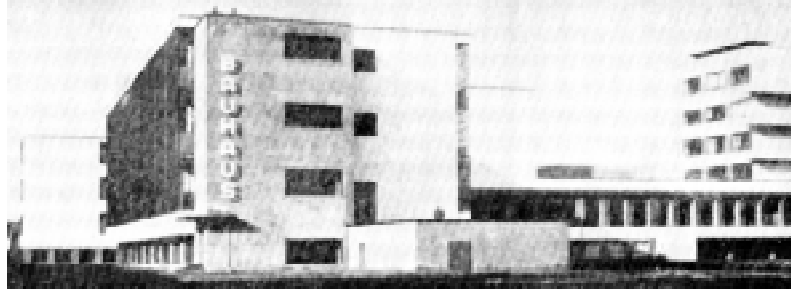


Figure 2.11: The Bauhaus at Dessau, designed by Walter Gropius, 1926

With the dawning of the modern movement in the 1920's, artists sought to incorporate the characteristics of the highly mechanized society around them. Most dramatically at the Bauhaus, the famous German art school, artists examined, even exalted the power of the engineer, the one who built the new tools they used, the bridges, buildings, cars, essentially the world they lived in. This new breed of artist sought to create works that could be mass-produced and still remain true to the original artist. These artists also used the visual language of the modern world, borrowing bits from the new angular environment and from the languages developed by the engineers. The modernists created artworks which seamlessly integrated visual languages much more explicitly than those that had come before (such as the renaissance work previously noted). For the purpose of our investigation, we focus on Paul Klee's diagramming of two-dimensional space, Oscar Schlemmer's diagramming of movement in three-dimensional space, and Laszlo Moholy-Nagy's mapping of theatrical space.

Paul Klee's lecture notes from his time at the Bauhaus are compiled in two lengthy volumes, *The Thinking Eye* and *The Nature of Nature*. In his preface to the first volume, Guilo Carlo Argan declares, "The writings which compose Paul Klee's theory of form production and pictorial form have the same importance and the same meaning for modern art as had Leonardo's writings which composed his theory of painting for Renaissance art." [Klee, 1961]

Within the volumes themselves, one finds innumerable small sketches, text descriptions, and large-scale oil paintings. All of these items are given equal importance; all interspersed. As an example let us examine Klee's examination of "Receptive preparation":

"For my first diagram of this function I choose a rather epic system to be read from left to right. On it I note the values, not in the order of rank, but, in accordance with the irregular displacements of the product, from bottom to top in the following order: third, fourth, eighth, ninth, first, fifth, seventh, sixth, second. The special position of first at middle height should be kept in mind. The sequence and order of rank are not identical."



Figure 2.12: Paul Klee, Product Figure 1



Figure 2.13: Paul Klee, Recept Diagram 2

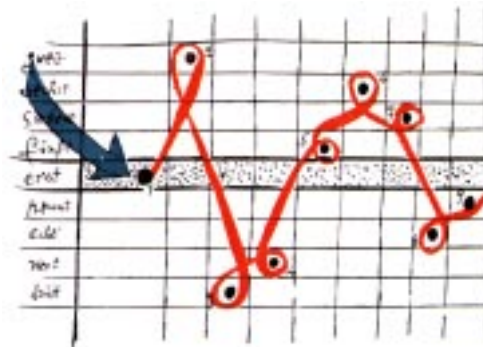


Figure 2.14: Paul Klee, Receptive action

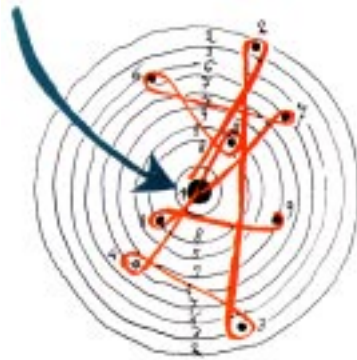


Figure 2.15: Paul Klee, Recept according to Diagram 2 corresponding to Product Figure 1

“For my second diagram I choose something more dramatic. The main value is inside; thus we have a central arrangement. The values are grouped according to their relation to the centre of the product:”

The “end result” of these studies in form creation can be found in a finished painting of a figure, about which Klee notes, “This ‘figure’ indicates interpenetration of outward form and inner nature...Once realized, the idea of dynamic construction from the inside perceives the specifically human content of this form from within.” Though of course, this “finished” work is seen as only the next step on to an additional body of thought and construction.

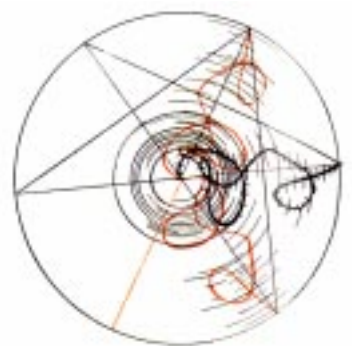


Figure 2.16: Paul Klee, Free movement on constructive foundation

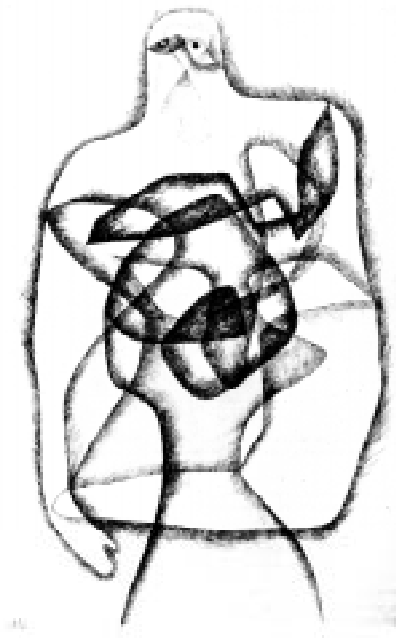


Figure 2.17: Paul Klee, *Figure*, 1931

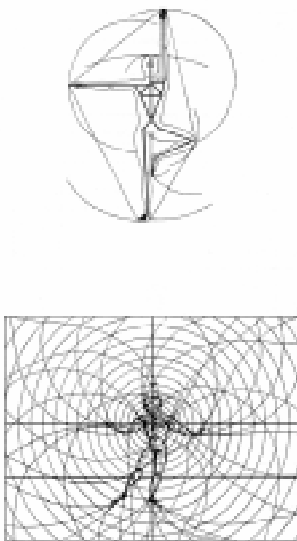


Figure 2.18: Calisthenics of the human body by Oscar Schlemmer

Klee, as an artist, was concerned not with form as an immobile entity, but formation and process. Never before had a painter so intertwined the engineering of design and theory in the modern age.

To Walter Gropius, director of the Bauhaus, Oscar Schlemmer was the school's "Master Magician." Gropius had hired Schlemmer to head the sculpture workshop; but step by step the sculptor broadened the scope of his workshop and transformed it into the Bauhaus stage shop. For Schlemmer, to design for the theater was to design for movement of intercommunicating parts in space. In the process he created a new understanding of human motion and gesture.

Schlemmer employed abstraction and mechanization: abstraction for “the construction in bold outline of a new totality” and mechanization as “the inexorable process which now lays claim to every sphere of life and art.” He was obsessed with identifying new symbols. Schlemmer thought it a “mark of Cain in our culture that we have no symbols any more and - worse - that we are unable to create them.” [Gropius, 1961]

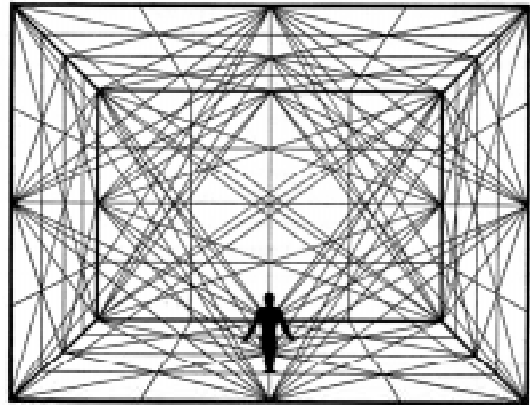


Figure 2.19: Oscar Schlemmer, laws of human body in cubical space

More often than not, Schlemmer would test out his diagrammatic theories through actual performances. In figure 2.21, we see a number of poles attached to an actor to emphasize the movement of his limbs.



Figure 2.21: Schlemmer performance accentuating movement of limbs

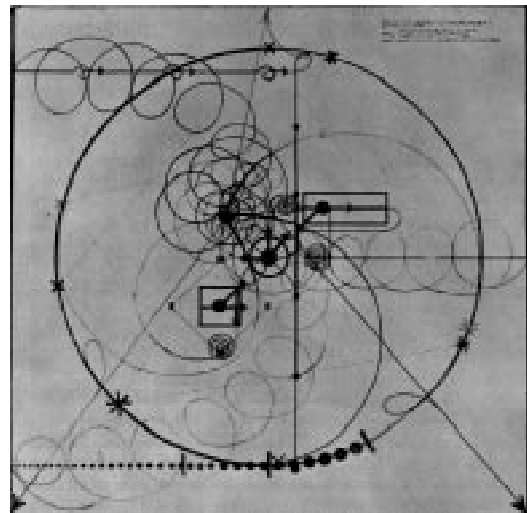


Figure 2.20: Diagram of the Gesture Dance by Oscar Schlemmer

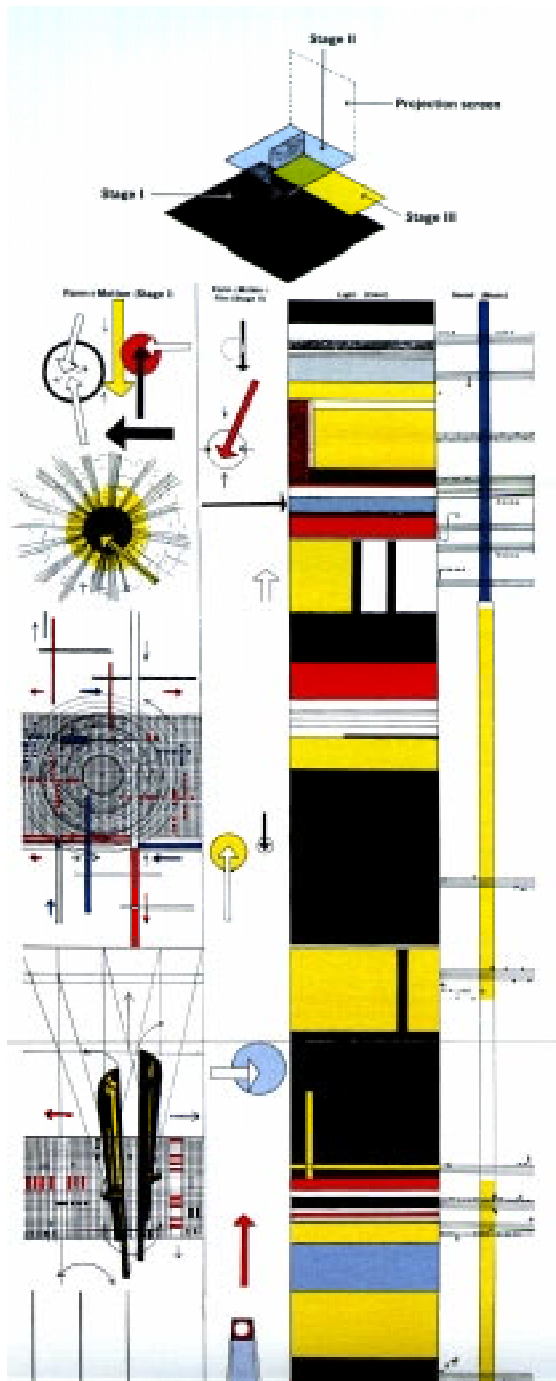


Figure 2.22: MECHANIZED ECCENTRIC by Moholy-Nagy

In Figure 2.22 we see the compositional sketch for a “MECHANIZED ECCENTRIC,” by Laszlo Moholy-Nagy. Moholy-Nagy was a master at the Bauhaus as well and can be considered a pioneer of truly “multi-media” creation. Beginning his career as an abstract painter, Moholy branched into sculpture, photography, typography, advertising art, film, and theater; his ultimate goal being the manifestation of a “vision in motion,” a new concept of space.

Many of Moholy-Nagy’s skills shine in this beautiful diagram of space, movement, light, and sound over time. At the very top of the score we see an isometric view of the proposed stages on which the action for this performance takes place. Stage 1, the lower stage, is for larger movements and forms. Stage 2, the upper stage, is for smaller, more restricted movements and film projections. On Stage 3, the intermediate stage, one finds mechanized instruments, percussion, megaphones, sound effects, noisemakers. The score itself is read from the top down. The leftmost column shows the actions on stage 1: “arrows plunge, louvered shutters open up, disks rotate, electric apparatus, grid systems of colors shoot up, down, back, forth...mechanized men.” The second column shows the form, motion, and film sequence of stage 2. The third column shows light effects over time and the fourth signifies the musical score (the long vertical stripes indicate siren sounds).

This work by Moholy-Nagy serves as an excellent transition to the pictorial and artistic representation of sound and/or music. On

some level, this may be seen as the most basic symbolic transformation as each of the twenty-some letters of our alphabet stands for a particular sound. It is this invention, that of written language, which has allowed men and women to record and build upon their achievements. A great many artists have focussed on the explicit transformation of sound notation to artistic abstraction. Here we see musical interpretations given visual form by John Cage and Toschi Ichiyanagi (Figures 2.23 and 2.24).



Figure 2.24: Toschi Ichiyanagi, Score for music for electric metronome



Figure 2.25: Still from Walt Disney's *Fantasia*, 1940



Figure 2.23: Musical notations by John Cage

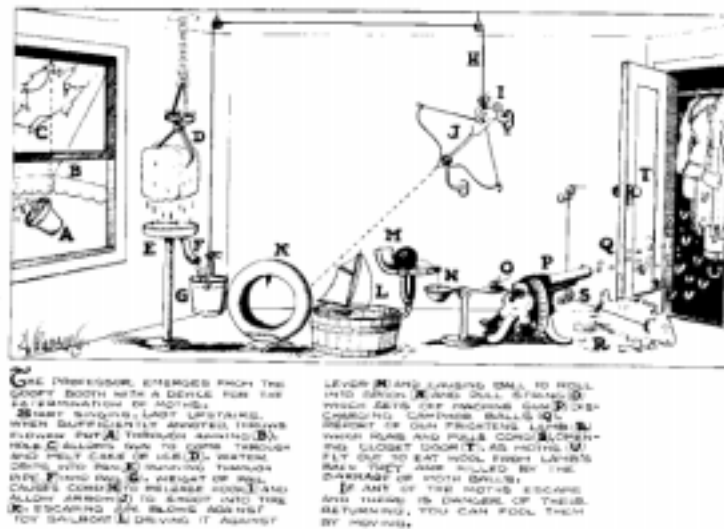


Figure 2.26: Cartoon diagram of process by Rube Goldberg, circa 1916

Perhaps the most popular artist to emphasize diagrammatic process in his work is the cartoonist Rube Goldberg. Over the course of his career, Goldberg developed countless comic strips and characters in innumerable publications; but he remains best-known for his machines meticulously labelled and rendered for demonstrating sequential actions (Figure 2.26).

Quite clearly all of the previous artistic examples walk the line somewhere between absolute form and pure function. They may accurately represent a process in color, space, sound, time, or some combination thereof; but more likely they provide dramatic expression of the sense of the process.

Human-Computer Diagrammatics

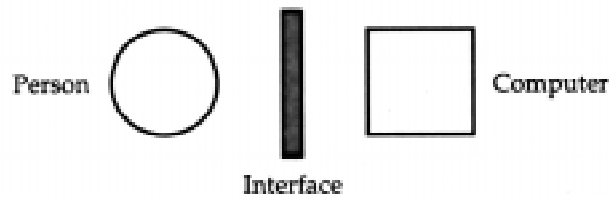


Figure 2.27: The pre-cognitive-science view of the interface

Brenda Laurel's *Computers as Theatre* provides an excellent overview of current prevailing models of the human-computer interface. In Figure 2.27, we see a schematic representation of the interface. The vertical band in the center of the diagram depicts the interface: the screen, keyboard, hardware, software, etc. For the interface to function, each participant must have some knowledge of the other. The computer must have a limited set of actions to expect of the human and computer user must know something about the functioning of computers. This gives us the “mental model” in Figure 2.28.

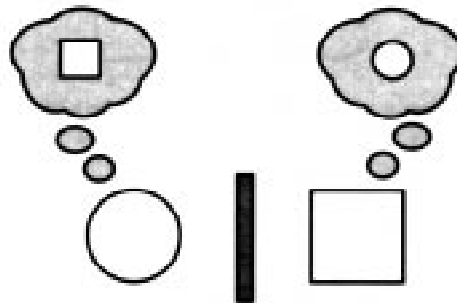


Figure 2.28: The mental models view

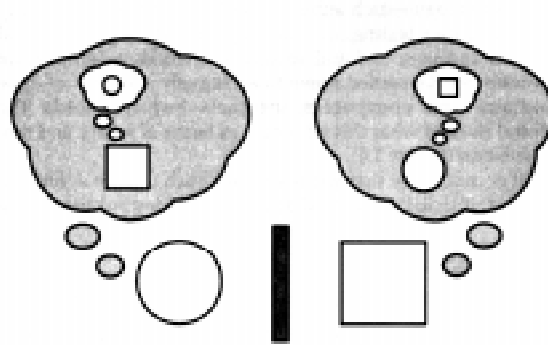


Figure 2.29: The “horrible recursion” version of the mental models view

“However, in order to use an interface correctly, you must also have an idea of what the computer is expecting you to do. If you are going to admit that what the two parties ‘think’ about each other is part of what is going on, you will have to agree that what the two parties think about what the other is thinking about them must perforce be included in the model” (Figure 2.29). [Laurel, 1993] This model raises some of the main problems facing those who would construct proper cognitive diagrams of the human-computer interface. We will attempt to address some of these issues later in this paper. For now, one may accept, as Laurel does, that the ramifications of this type of “horrible recursion” model are unbounded and we must move on.



Figure 2.30: A simple model of the interface, circa 1989

A more stable model of the interface is shown in Figure 2.30. This is, very probably, the most common method of conceptualizing the human-computer interface. Here, the interface links the human and computer, conforming to the needs of each.



Figure 2.31: Laurel's
"interface as theater" model

Laurel goes on to propose her own model for the interface, based upon her conception of the interface as theatre play in which human and computer participants are actors (Figure 2.31). Here the human-computer interface becomes a "virtual world" populated by human agents, computer-generated agents, and representational structures (windows, desktops, icons, etc.). I would argue that this model, though certainly on the right track, is perhaps no less vague or oversimplified than the first in this series (Figure 2.27).

In summary and to return to our original overview of symbolic representation, if we can make one general requirement of symbolic systems it is this: symbolization appears to require at least two interacting subjects immersed in an environment common to both. In this study we use this established history of diagrammatic communication as the point of departure for our "living diagrams."

Symbolic "Programming"

We now investigate the creation of symbolic "programming" languages. We examine symbolic programming languages of the graphical, physical, and theoretical varieties. In this case, "programming" does not refer specifically to computer programming. Though each of the languages referred to are certainly derivative of the computer age and its corresponding flow diagrams described above.

In his book, *Vehicles*, Valentino Braitenberg employs the visual language of engineering to explain cognitive functions. "This is an exercise is fictional science, or science fiction, if you like that better. Not for amusement: science fic-

tion in the service of science. Or just science, if you agree that fiction is part of it, always was, and always will be as long as our brains are only minuscule fragments of the universe, much too small to hold all the facts of the world but not too idle to speculate about them.” [Braitenberg, 1984] In this spirit Braitenberg proceeds to describe a series of mechanical constructions that eloquently demonstrate structures of the brain.

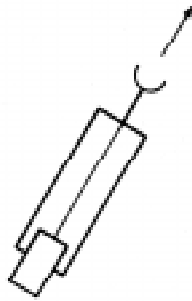


Figure 2.32: Braitenberg
Vehicle 1

His first vehicle, the simplest, has one sensor and one motor (Figure 2.32). The more of the thing the sensor is tuned to accept, the faster the motor turns. This vehicle is quite stupid, since it can't turn outside of external disturbances, but even still one may still think it *alive* since one seldom sees nonliving matter behave as this vehicle does.

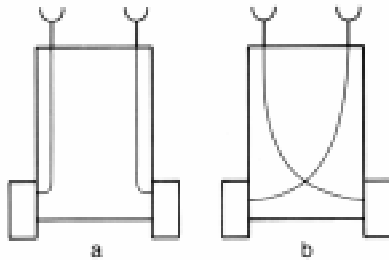


Figure 2.33: Braitenberg
Vehicle 2, with two motors
and two sensors

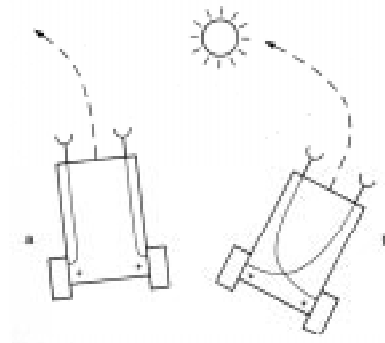


Figure 2.34: Vehicles 2a and
2b in the vicinity of a source

The second vehicle employs two motors and two sensors (Figure 2.33), essentially two of the first vehicle strapped together. As before, the more the sensors are affected, the faster the motors go. Note the subtle differences in wiring between the two versions of this vehicle. Figure 2.34 shows the dramatic effect these two wiring configurations have when given two vehicles whose sensors are tuned to respond to light intensity. Vehicle (a) turns away from a light source. Vehicle (b) turns towards it, not unlike a fly to a light bulb. “Decussation” is a bunch of nerve fibers linking sensors on one side of our body to the half of the brain on the opposite side. Scientists have long puzzled as to why the optic nerves cross to the opposite side of the vertebrate brain.

Later vehicles introduce threshold control, evolution, abstraction/generalization, vision, shape recognition, and memory. All the while, Braitenburg is far less concerned with the actual construction of the vehicles than he is with the motivation behind the construction. With such simple, elegant examples Braitenburg is able to propose possible answers to questions that have confounded brain scholars for decades. We cannot hope that our document will have the same effect on computer-human understanding. We can, however, emulate Braitenburg’s methodology of construction as theory.

In 1988, Paul Haeberli introduced ConMan. ConMan is a visual programming language for interactive graphics. About ConMan, Haeberli says, "We want to use the facilities of the modern interactive medium more effectively to give the user more expressive power and freedom to construct and modify applications in a flexible way. Why isn't application development more like making a bacon, lettuce, and tomato, cucumber, salami, avocado, Jell-O, and sushi sandwich? Can't we use the interactive medium itself to help us?" [Haeberli, 1988] ConMan uses a data flow metaphor. A user builds applications by interconnecting components in a directed graph on the screen. In this project, Haeberli makes a dramatic statement away from the myth of the monolithic, self-contained application and towards flexible, open-ended systems of intercommunicating parts.

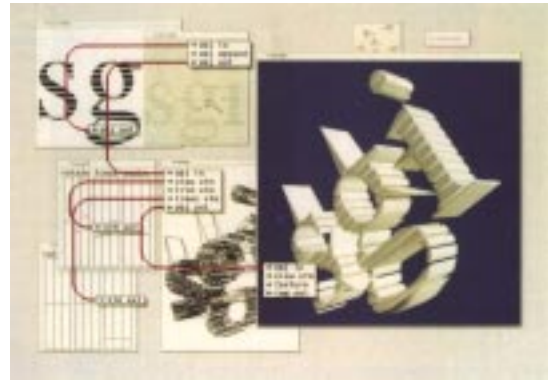


Figure 2.35: Screen from *ConMan* by Paul Haeberli

Recently, the human-computer interface community has sounded the call for the development of more "physical" interfaces. Along these lines John Maeda proposes a novel physical programming language in his book, *Dynamic Form*. Maeda's "solid programming" is facilitated through a number of blocks, each of which has a single input and a single output. A given block can be tagged with cues for a single "fuzzy logic" if-then statement, such as "if cold then high power" or "if heavy then none." To form more complex parallel relationships, one may snap blocks together in the x and y directions and thereby combine sets of rules together in parallel. By adding input sensors and output actuators, one may build physical logic systems. [Maeda, 1993]

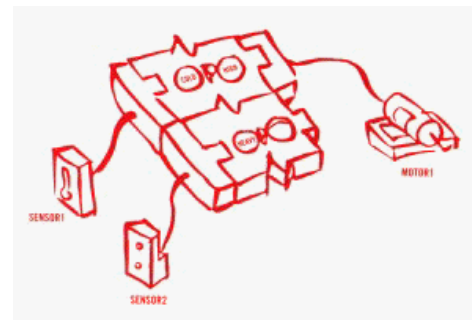


Figure 2.36: Physical programming language by John Maeda

2.2 Machine Performance Spaces

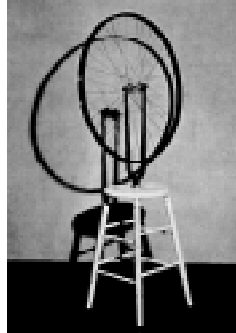


Figure 2.37: Marcel Duchamp, *Bicycle Wheel*, 1913: perhaps the first physically interactive artwork

Here we investigate some of the many historic artistic experiments in having machines “talk” and “listen” to each other in different ways. Artists explored the creation of intercommunicating parts and dynamic form long before input/output (I/O) devices became prevalent.

Though the concept for his work had little to do with the search for movement in art, Marcel Duchamp's notorious *Bicycle Wheel* may have been the first kinetic sculpture (Figure 2.37). Duchamp was interested in the presentation of a new reality by taking a familiar object and exposing it within the context of the art world. In the process, he unearthed the everpresent mechanization of the modern world. His later *Rotorelief* explored this theme more directly (Figure 2.38). The *Rotorelief* consists of a painted disc connected to an electromotor. When in motion, the flat disc appears to transform into a three-dimensional conical form turning in space.

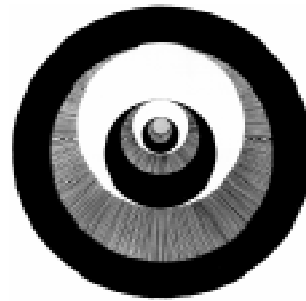


Figure 2.38: Duchamp's *Rotorelief*, 1925

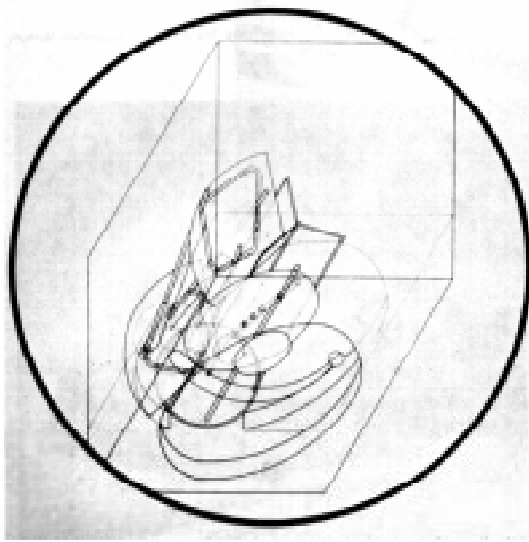


Figure 2.39: Sketch for the *Light-Space Modulator*, Moholy-Nagy

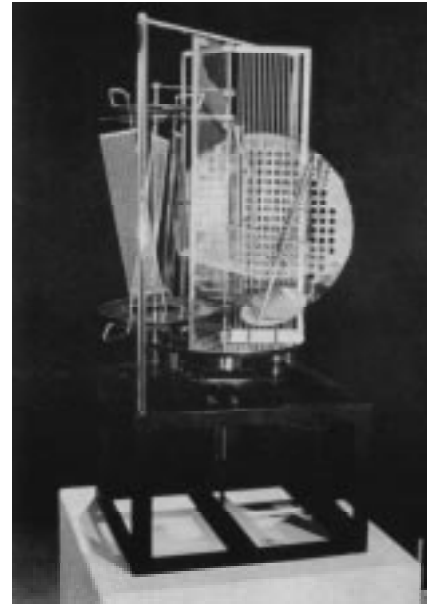


Figure 2.40: The *Light-Space Modulator* by Moholy-Nagy

The namesake and perhaps spiritual guide for our current project is the *Light-Space Modulator* by Laszlo Moholy-Nagy. The *Light-Space Modulator* is an electro-mechanical kinetic sculpture built of glass, chrome, wire, and rods (Figure 2.40). When plugged in, the device turns, spirals revolve, a ball spins down, beams of light and shadow play on the walls and ceiling. The sculpture was created not for the metallic forms themselves, but for the reflections produced by these structures and their impact on the surrounding architecture and space. Moholy made a film which starred the *Light-Space Modulator*, entitled “Light Display: Black and White and Grey.” Through kinetics, he sought to sculpt what he called “virtual volumes,” the trajectory or outline shown by an object in motion.

Moholy worked with the piece off and on for the 9 years between 1922 and 1931. He employed an assistant, a craftsman to help build the *Modulator* and won his future wife’s affections by showing it to her. She later relates how the work became like a problem child as they toted it with them wherever they went. It is now housed in the Busch-Reisinger Museum of Harvard, a mere mile away from MIT!

By Moholy's definition, any structure which changes or transforms matter of one sense through materials may be considered a Modulator. This definition will be shown to have particular significance in the development and interpretation of our own "Modulator" system further on.

Later in his career, Moholy-Nagy abandoned his light architecture projects. He did this not because he was finished with his explorations nor because he felt that the area no longer held great promise. He explains his frustration thus:

It would be easy to give an obvious answer and to say that the physical dependence on capital, industry and the workshop is an unmovable hindrance to the development of light architecture, which holds out no immediate promise of practical application, producing only the emotions denied from color in space. While a painter in his studio, possessing only a few tubes of color and a few brushes, can be a sovereign creator, the artist in "light-play" easily becomes the slave of technical considerations as well as of his material. Technical consideration can, indeed, be given too great an emphasis, especially considering the general fear lest scientific knowledge and controlled technique should dominate art.

It is a legend of cowardice that says that intellectual perception does harm to the artist, that he requires nothing but feeling and intuition for his work. As if we knew nothing of Leonardo, Giotto, the cathedral architect, Raphael, or Michelangelo, whose creative power increased with the increase of his knowledge and the development of his skill!

However, after these fears have been overcome, as they soon can be in the intense interest and concentrated effort required by the task and the exaltation of spirit which it induces, there still remains the paralyzing difficulty of presenting what has been done and demonstrating what has been achieved. There scarcely exists a building in which the creation of the "light-artist" could be made accessible

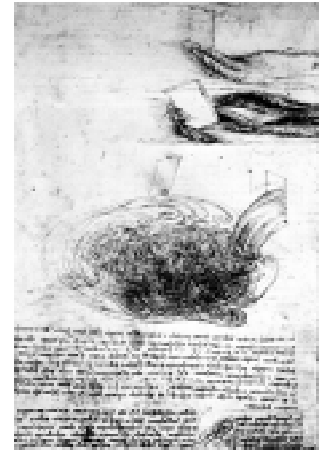


Figure 2.41: A study of streaming water from one of Leonardo da Vinci's sketchbooks

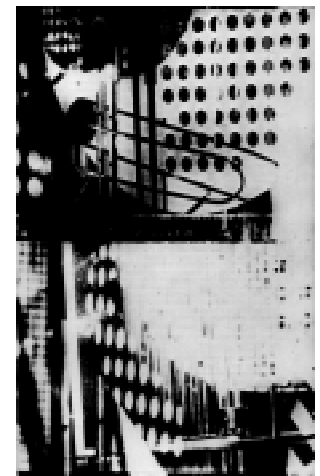


Figure 2.42: Still from Moholy-Nagy's *Light Display: Black and White and Grey*

to the public. The dream that has been realized is put away into storage and there it remains until it fades away in the insignificance of its own isolation.[Kostelanetz, 1970]

When Moholy-Nagy wrote these words in 1936, the modern digital computer was barely a twinkle in von Neumann's eye. Yet just the same, Moholy vividly outlines the advantages and issues of the digital designer of the 1990s. We are now able to quickly sculpt the light architecture first realized in Moholy's creations dynamically and in living color. However, the technical considerations of the modern light sculptor remain just as daunting as in Moholy's time.

Nine years before Alexander Calder's first mobiles, students in Moholy-Nagy's classes at the Bauhaus were creating dynamic sculptures, balancing unequal forms. However, these efforts remained somewhat clumsy as these works sought out some unseen power source. Moholy tried remote electrical and magnetic controls. It was up to Calder to discover random currents of air. [Kostelanetz, 1970]



Figure 2.43: Alexander Calder with one of his mobiles

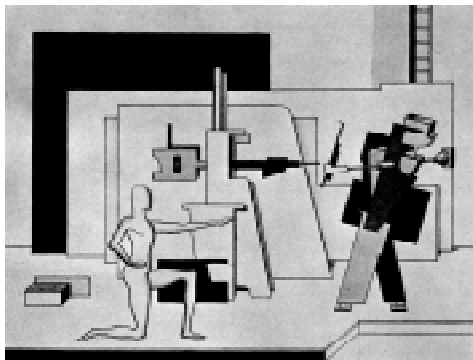
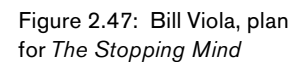
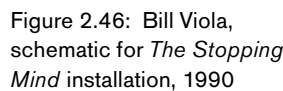
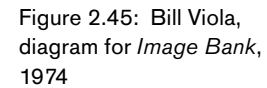


Figure 2.44: Kurt Schmidt, *Man + Machine*

The Stopping Mind is a video installation for projected images and sound based on the seemingly innate human desire to stop time. Four large screens hang from the ceiling and are suspended in space in the center of a dark room. Four unique but related images are projected on each of the four screens. The images are still, the room is silent. Suddenly the images burst into frantic motion and the room is filled with cacophonous sound. After a few seconds, the images freeze and the sound stops.



2.3 Reconfigurable Performance Spaces

As many artists created kinetic experiments with objects in space, others in the areas of stage design and theater architecture focussed on dynamic environments in which to perform. The designers of the Bauhaus developed some highly refined, though unbuilt studies for reconfigurable theaters.

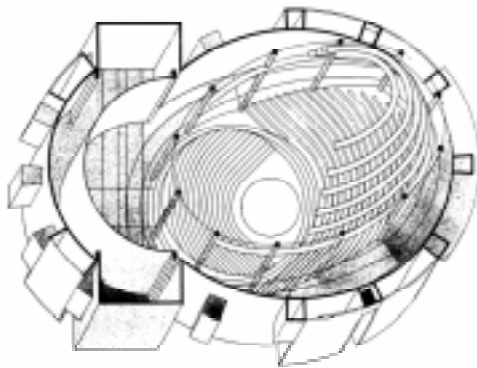


Figure 2.48: The *Total Theater* by Walter Gropius, view from above

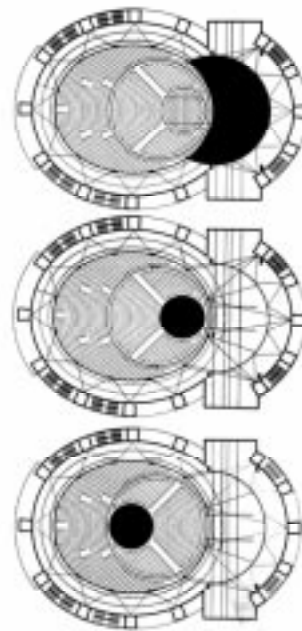


Figure 2.49: Stage arrangements for *The Total Theater*

Walter Gropius' plan for the *Total Theater* from 1926 (Figures 2.48 & 2.49) allows for the stage and seating to transform into three arrangements. The center section rotates for deep stage, proscenium stage, and center stage configurations. Gropius also allowed for twelve projection screens to be placed between the columns, thus surrounding the spectators with light performance.

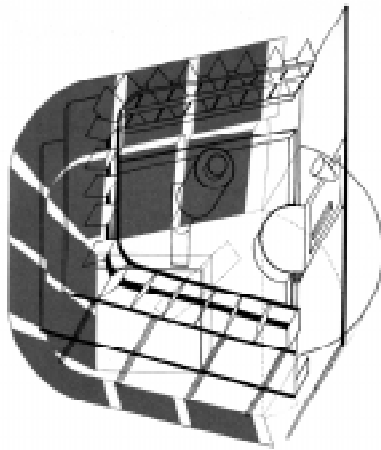


Figure 2.50: *U-Theater* by
Farkas Molnar

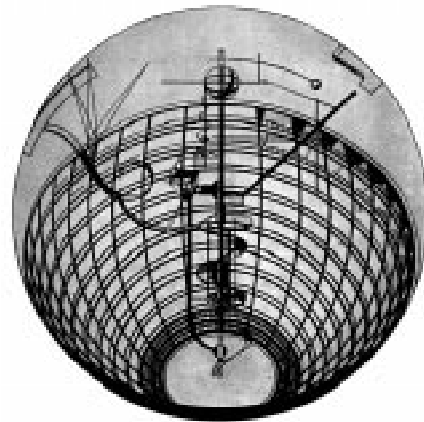


Figure 2.51: *The Spherical
Theater* by Andreas Weininger

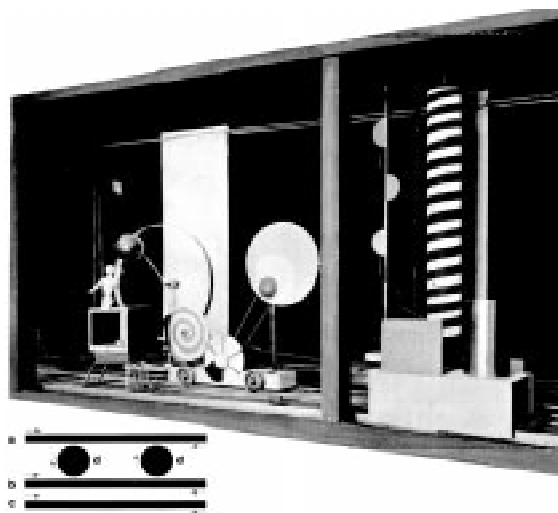


Figure 2.52: *Model for a
Mechanical Stage* by Heinz Loew

Students of the Bauhaus designed stunning immersive theaters as well. In Figures 2.50 and 2.51 we see contrasting theater designs by Farkas Molnar and Andreas Weininger which were clearly influenced by Oskar Schlemmer's free definition of theater space.

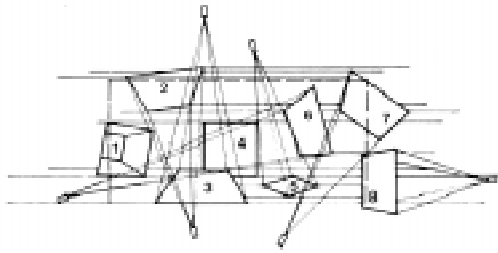


Figure 2.53: Josef Svoboda, early frontal diagram of *Polyekran*, 1958



Figure 2.54: Performance of *Polyekran*, scenography by Josef Svoboda, 1958

In Josef Svoboda's *Polyekran* set-up for the 1958 Brussels World's Fair the artist employs the use of eight projection screens arranged about the performance space. Svoboda does not attempt to hide the screens. Instead he highlights them, creating architectural space with the screens alone.

In 1967, a group of performers in New Orleans led by Richard Schechner produced a unique version of Eugene Ionesco's *Victims of Duty* in which an entire studio was transformed into the living room of the play's main characters. During the performance the room was populated not only by the actors, but by the audience as well. *Victims of Duty* was the first American "environmental" theater production, the first to intermingle audience and performers in a single designed unit.

[McNamara, 1975]



Figure 2.55: *Victims of Duty*, Richard Schechner environmental theater production, 1967

Cerebrum, an “intermedia and human sensorium,” takes this evolution of the performance environment one step further. Cerebrum was set up in New York in the late 1960’s as something between a “theater,” a “gallery,” and a “nightclub.” A participant purchased three hours of time to practice “leisure, decision-making, interpersonal responsibility, body awareness, and sensory perception.” Participants donned white, diaphanous gowns. Guides led around the participants and facilitated interaction with other participants and with the mediated environment. Eclectic music played, scented fog wafted, and food was passed around. Some scripted events affected all participants simultaneously, other unscripted happenings were created by the participants themselves. The environment mutated dynamically during each “performance.” [Youngblood, 1970]



Figure 2.56: *Cerebrum*, New York City, late 1960’s

2.4 Human-Computer Interfaces for Performance

In this section of our investigation of historical precedents the reader may rightfully get somewhat depressed. After examining a series of free expressions of form, time, and structure, we are now in a place to review computational works. For the most part, these digital creations exhibit little of the glory seen in our previous examples. Yet the field of human-computer design is now over thirty years old and its history deserves examination. This history provides an appropriate backdrop for the motivation behind our current research.



Figure 2.57: The first issue of the EAT newsletter

The early days of creation using digital means were marked by dramatic collaborations between highly skilled engineers and well-known artists such as the EAT group.

After receiving his Ph.D. in electrical engineering from the University of California at Berkeley, J. Wilhelm (Billy) Kluver joined the technical staff at Bell Laboratories to work on the physics of infrared lasers. While at Bell Labs, Kluver contributed his engineering skills to works by Jean Tinguely, John Cage, Robert Rauschenberg, Jasper Johns, and Andy Warhol. In 1967, Kluver, with artist Rauschenberg, founded Experiments in Art and Technology (EAT). The stated mission of this highly influen-

tial group was to develop “an effective collaborative relationship between artists and engineers...Engineers who have become involved with artist’s projects have perceived how the artist’s insight can influence his directions and give human scale to his work. The artist in turn desires to create within the technological world in order to satisfy the traditional involvement of the artist with the revelant (sic) forces shaping society.” [from Davis, 1973]



Figure 2.58: Pepsi-Cola Pavilion, Osaka, Japan, World's Fair, 1970

The EAT approach is perhaps best exemplified by the *Pepsi Pavilion*, their contribution to the 1970 World's Fair in Osaka, Japan (Figures 2.58 & 2.59). This 90-foot-high domed environment was the result of a complex web of collaborating artists, engineers, architects, and scientists, orchestrated by Kluver himself. The pavilion featured the largest spherical mirror ever made, which produced reflections of viewers on the dome ceiling, a light-sound environment specifically made for the spherical environment, and a man-made water cloud, which floated gently above the dome.

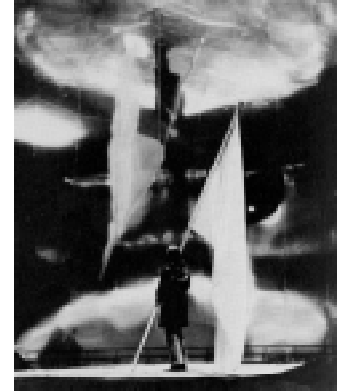


Figure 2.59: Interior, Pepsi-Cola Pavilion

In the early 1970's, Artist Charles Csuri and engineer Ivan Sutherland developed programs for intricate linear abstractions. Forms billowed, expanded, and contracted on a video monitor and could be controlled interactively with a light pen. Sutherland is also well known as the inventor of the first color computer graphics display as well as the first three-dimensional computational space through the use of tiny cathode-ray tubes mounted on the viewer's head. Perhaps most significantly, Sutherland's *Sketchpad* system of 1963 may be thought of as the first direct-manipulation human-computer graphical interface. [Davis, 1973; Sutherland, 1963]



Figure 2.60: Artist Charles Csuri using a light pen to draw animated objects



Figure 2.61: Engineer Ivan Sutherland wearing head-mounted display, 1970



Figure 2.62: James Seawright, *Network III*, 1971: Spectator walks over a carpet underlaid with pressure plates which activate a computer-controlled pattern of lights overhead



Figure 2.63: *The Architecture Machine*, 1970

One particularly interesting computational project from this same era comes from the MIT Architecture Machine Group led by Nicholas Negroponte. In his book, *The Architecture Machine*, Negroponte calls for the invention of “robot architects” in the name of an environmental humanism between man and machine. As a functioning example of this thesis, the group from MIT created a project entitled *Seek*. *Seek* is a plexiglass construction in which a group of gerbils rearrange a series of small blocks. A robotic arm then patiently rebuilds symmetric structures related to the gerbil’s implicit wishes. The project was

featured in the *Software* exhibition of 1970 at the Jewish Museum of New York City. Though hardly performative, this ingenious project provides a dramatic metaphor for the implied relationship between man and machine. [Davis, 1973]

When the MIT *Architecture Machine* broke down at the Jewish Museum in 1970, Thomas Hess reflected the widespread sentiment of tension between artists and engineers in an editorial in *Art News*. He described the shipwrecked gerbils, trapped by the broken arm of the computer, covered with their own excrement. “Artists who become seriously engaged in technological processes might remember,” he concluded, “what happened to four charming gerbils.” [from Davis, 1973]

Out of this atmosphere came the movement to create tools to enable all artists to work with computers, no matter their skill with computer engineering. In his book *Art and the Future* from 1973, Douglas Davis writes, “There are strong indications that computers will yet program themselves, and that everyone will have direct access to them, either by telephone or by miniaturized computer units, some no larger than transistor radios. In that state, the new tool will be as accessible to the artist as the brush, pen, or camera is today.” [Davis, 1973]

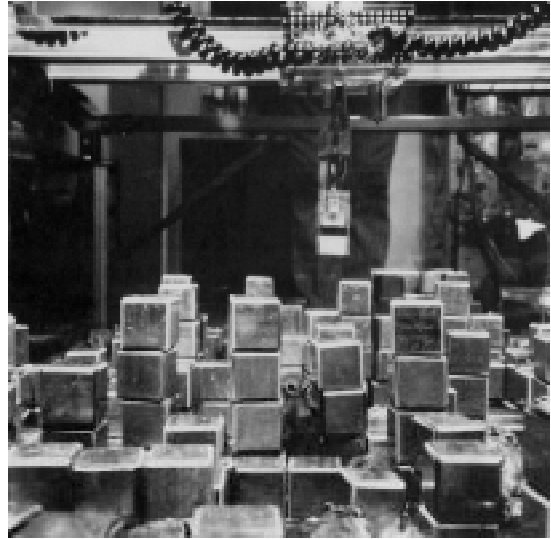


Figure 2.64: *Seek*: live gerbils, aluminum blocks, and computer-controlled robotic arm



Figure 2.65: The Apple IIc personal computer, 1983

The introduction of the personal computer and a great number of commercially available tools aimed at artists and designers seemed, at least on the surface, to address this situation. Anyone, with the click of a button could create “computer art.” Artists, it seemed, were again free from the weighty concerns of the mathematicians and engineers. As a result of this innovation, creation on computers was divided into two very distinct camps: the tool makers, chiefly engineers with a few designers, and the tool users, chiefly artists and designers.

Now in the late 1990's we wonder why so many works made using the computer look alike. "...the general conclusion is that the implicitly constrained nature of modern digital painting and sculpture tools has trapped many designers within the same stylistic boundaries as other designers in tremendous proportions, an unprecedented phenomenon in the history of design." [Maeda from ACG, 1997] In the 1990's, the creation of the tools themselves appears to be one of the only avenues for freedom of expression using programming.

Current tools for real-time audio or visual performance using computers involve obtuse controls, either heavily GUI'ed or overstylized, making it difficult for the audience to understand exactly what the performer is doing. Currently the process of editing sounds or manipulating three-dimensional structures on a computer remains a frustratingly rigid process.

Digital Tools

Considerable work has been done in the creation of computational environments for designing three dimensional form. *Autostudio* by Alias|Wavefront Incorporated is one of the most advanced commercially-available, three-dimensional modelling systems. In *Autostudio*, the visual display is divided into four quadrants, each a view into the current state of the three-dimensional model: from above, from the side, from the top, and in perspective. The user adds a shape by clicking on the icon for various primitives: cube, sphere, cone, and cylinder. The user may then apply a wide range of predefined transformations to the object or objects to build desired constructions and set keyframes to arrange animations of these constructions. All of this leads up to a final "render," either a still frame or a series of frames which make up an animation.

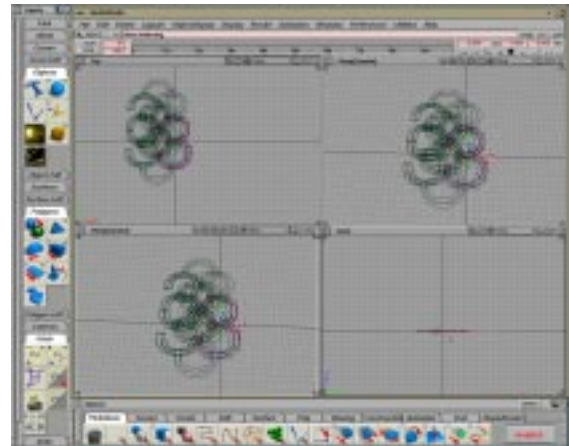


Figure 2.66: Using
Alias|Wavefront *Autostudio*

Three-dimensional modeling tools have evolved from earlier computer-aided design (CAD) tools for engineering and architectural design. These applications have sought to extend the visual vocabulary of the traditional designers drafting table mixed with direct analytic geometry transformations (“rotate,” “scale,” “translate,” etc.). Even given such rigid constraints, one may be amazed to watch a highly proficient user in control of *Autostudio* or a similar application. These so-called “demo jocks” can, in an instant, create whale-like forms with undulating tails drifting through space or complex series of ovular ants marching down a pole. All the while, the user may spin around the world or peer deeper inside, examining the computational space. However, the interfaces are not designed for this type of performance - the live control of the environment. The products succeed based on their ability to create “realistic” final depictions. Recent efforts to build modelling software suggest some movement in alternate directions.

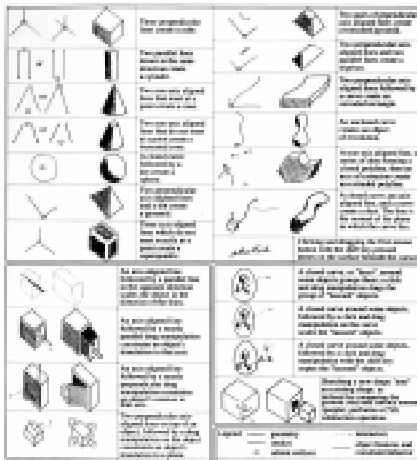


Figure 2.67: Gestures for creating and manipulating shapes in Robert Zeleznik's *Sketch* program

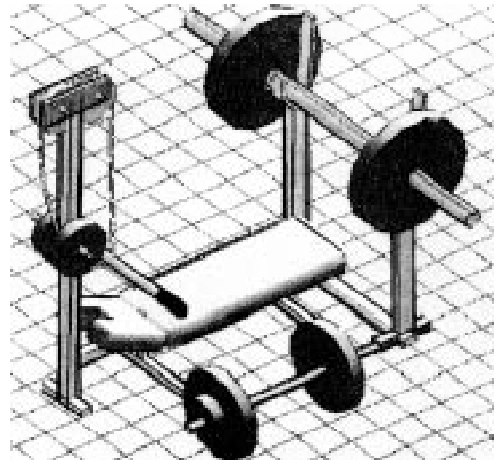


Figure 2.68: Sample construction in *Sketch*

Research has emerged of late in the use of gesture recognition in the manipulation of three dimensional form, perhaps best in the *Sketch* system by Robert Zeleznik [Zeleznik, 1996]. The *Sketch* system attempts to rethink the common three-dimensional modelling program. Instead of a highly precise, numerical environment, *Sketch* presents a space wherein the user is able to rapidly sketch

out ideas in three-dimensions. The *Sketch* screen does not employ any common GUI tools, instead all actions are based on gesture recognition and on the context of the three-dimensional form being acted upon. The most notable aspect of this approach is the sense of comprehensibility that the system appears to have, in spite of its gesture complexity.

Mr. Zeleznik visited MIT this past year to demonstrate the wizardry of his *Sketch* program. In Zeleznik's hands, he could quickly craft all manner of three-dimensional forms: tables, lamps, windows with windowshades, pianos with keys, etc. The computer screen seemed to leap at his touch, all based on rapid motions with the mouse and the occasional touch of a key. The effect was truly amazing. When one of the graduate students in the group watching his demonstration asked to try out the system, he graciously offered his seat. However, even with its maker leaning over the his shoulder coaching, the graduate student could barely make a cube with *Sketch*.

Though the *Sketch* interface is without buttons or menus or dialogue boxes, it is also very difficult to learn at first. Or rather, in order to learn the *Sketch* system, one must use dramatically different methods to those employed when learning traditional graphical user interfaces. The *Sketch* paradigm does not provide for the common exploratory learning of most commercial software products. If one tries to attempt to explore with *Sketch*, perhaps by tossing the mouse around or clicking each mouse button in various combinations, seemingly random events occur on the screen. Upon seeing *Sketch* live, the immediate impression one gets is that the system is particularly tuned to Robert Zeleznik's gestures. This is probably not the case. A more likely situation is that the process of learning *Sketch* is more similar to learning to hit a baseball or ride a bicycle than memorizing what icon stands for "extrude."

In Figure 2.69 we see a typical screen from a common musical editor. In this program, *SoundEdit 16* by Macromedia Inc., time flows from left to right across the display. The user adds sound clips of either the midi or sampled varieties which appear as horizontal bands. By selecting a given band with the mouse, then clicking on one of the pull-down menus at the top of the screen the user is presented with a dialog box. He or she can then change one or more of the parameters of the given sound, such as the speed of the clip. The user clicks “ok” and the change is then reflected on the display. At any time, the user can press the “play” button on the control panel to monitor the current state of his or her progress. The user then hears the present arrangement of the sounds and a line works its way from left to right across the display to denote the position of the “play head.” This same interface or one with slight modifications is found in the majority of commercial audio editors available today, such as Cakewalk *Pro Audio*, etc.



Figure 2.69: Screen from Macromedia *SoundEdit 16*



Figure 2.70: Screen from Cakewalk *Pro Audio*



Figure 2.71: Traditional Vinyl DJ setup: two turntables + 1 mixer



Figure 2.72: *Mixman Studio* software interface

In the last two years, several commercial software packages have been introduced which attempt to emulate preexisting analog devices for musical performance. *Mixman Studio*, by Mixman Software, takes as its visual model the traditional disk jockey setup consisting of two turntables and a cross-fader between them (Figure 2.72). The product's stated goal is to allow users to create their own "mixes" built of sound samples. Though the *Mixman* interface looks like two large vinyl records (Figure 2.71), the user soon learns this is, in fact, a false representation of what the interface actually does. The *Mixman* design allows its creators to craft up to 16 samples which can be played simultaneously. To play a given sample, the user clicks on one of the small, glowing "bulbs" radiating outwards from the center of the vinyl record graphic. Click the bulb once, the sample turns on, click the bulb again, the sample turns off. Ultimately, *Mixman* remains a hollow shadow of its former, analog self.



Figure 2.73: The Roland *TB-303* Bass Line analog synthesizer



Figure 2.74: Interface screen from ReBirth *RB-338* computational emulator

The ReBirth *RB-338* emulator takes the recreation of analog devices one step beyond the work of Mixman. This commercial product, released in 1997 by Propellerhead Software of Sweden, emulates in exacting detail the look and functions of the Roland *TB-303* Bass Line analog synthesizer. The *TB-303* was introduced in 1981 as an emulator for another analog device, the electric bass. Its creators hoped that single musicians or bands in need of a bass player would program the synthesizer to play simple, repetitive bass lines in place of an actual musician. The problem was that the *TB-303* sounded nothing like an electric base. Thus very few units were sold and the *TB-303* was, at the time of its creation considered a failure.

Then, about nine years later, came the introduction of techno music. This musical genre, bred from the clubs and underground parties of urban youth and fueled by simple, repetitive synthesized beats, claimed the *TB-303* as its instrument of choice. Though its interface consists of a myriad of tiny dials and levers and its programming is highly obstruse, the Roland *TB-303* obtained near mythical proportions among techno musicians. As techno music has now reached the mainstream, the demand for this device, of which only a few thousand were produced, has skyrocketed.

The Rebirth *RB-338* hopes to fill this demand for the original Roland synthesizer by recreating its look and sound computationally. Using digital signal processing the Rebirth *RB-338* sounds remarkably similar to its analog counterpart. Its graphical interface looks almost identical to the Roland synthe-

sizer. A click of the mouse on a lever changes its position. A click on a dial and it rotates accordingly. All the oddities of programming the original Roland have been left intact.

Though ultimately again, the Rebirth *RB-338* system remains significantly less satisfying than its analog parent. As before, the interface mechanisms were not designed for a graphical environment. The original Roland synthesizer gained popularity not merely because of its sound, but because of its shape, its weight, the unique feel of its levers, dials, and buttons, and certainly not least a certain amount of nostalgia for antiquated packaging and technology as well. Thus, one must develop unique interface concepts from the medium itself as opposed to adopting the interfaces of preexisting devices.

Digital Games

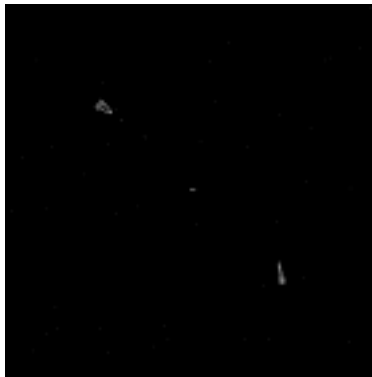


Figure 2.75: Reproduction of *Spacewar*, the first computer game

The first computer game was created in 1961 by a group of hackers at MIT. The group was sitting around a DEC PDP-1, the first minicomputer connected to a display, trying to figure out an interesting use for some pattern-generating software they had created. They decided that they could make a two-dimensional environment in which a person could maneuver objects. Spaceships seemed to be the obvious thing to maneuver and the game of *Spacewar* was born (later to become the popular arcade game, *Asteroids*). [Steadman in Jones 1995; Laurel, 1993]

Why was *Spacewar* the “natural” thing to build with this new technology? Why not a pie chart or an automated kaleidoscope or a desktop? Its designers identified *action* as the key ingredient and conceived *Spacewar* as a game that could provide a good balance between thinking and doing for its players. They regarded the computer as a machine naturally suited for representing things that you could see, control, and play with. Its interesting potential lay not in its ability to perform calculations but in its capacity *to represent action in which humans could participate*. [Laurel, 1993]

Stated in these terms, computer games may be the most performative, expressive interfaces in the digital medium. A computer game serves to engage its user in participatory activity. Very often these interactions include immediate dialogues between machine and user. Games often dramatically control time scales: thousands of years are compressed to hours in a civilization adventure, milliseconds become minutes in a molecular life simulation. To this day, the vast majority of games are made with low-level computer programming. As the speed of computers has advanced, the computer game has always pushed the limits of interactivity. For this reason, computer games generally integrate new forms of physical input and output first (trackballs, steering wheels, flight sticks, three-dimensional joysticks, haptic feedback joysticks, virtual reality displays, etc.)

Certainly the interactive computer game is entirely native to the computational medium. The various genres associated with gaming have evolved directly from the experiential man-machine interface.

Not only was *Spacewar* the first computer game, it was also the first “destruction” game. In a destruction game the participant attempts to “kill” as many things as possible in as short a time as possible without being “killed” yourself by a computational or additional human opponent. Today, more than thirty years later, these games remain as strong as ever as evidenced by the popularity of games such as *Doom*, *Quake*, *Mortal Kombat*, and *Descent* (Figure 2.76).



Figure 2.76: Screen from *Descent* - “destruction”

Destruction games are extremely dependent on achieving maximum speed with the human-computer interface. For this reason, the designers of these games often employ the most advanced computational techniques. In games such as *Descent*, the participant flies a nimble spacecraft through fully-rendered three-dimensional environments at 60 frames-per-second. Amazingly crafty, alien spacecraft hide behind corners, sneak up behind, band together in teams, and generally create general nuisances of themselves attempting to “kill” the player.

In an “exploration” game, the user embarks on a mission of discovery in a crafted environment. The first of these role-playing games was the text-based *Adventure*, written at Stanford in the 1960s. Exploration games are “interactive” through their use of a series of puzzles. The player must talk to synthetic characters, find and use objects, and learn about the settings in order to trigger the next stage of the plot. Other games of this type include *King’s Quest*, *Ultima*, and *Myst* (Figure 2.77). [Steadman in Jones, 1995]

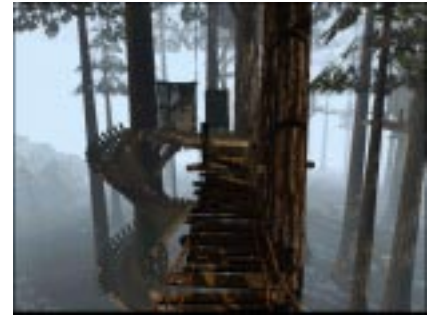


Figure 2.77: Screen from *Myst* - “exploration”

British mathematician John Conway invented what can be considered the first “creation” game. In his *LIFE*, introduced in 1970, the user plays with an open-ended model of systemic development. Creation games are characterized by environments in which users are encouraged to build things or guide the development of autonomous structures. Other examples of works in this genre include the many multi-user dungeons (MUDs), *Civilization*, and *SimCity* (Figure 2.78). [Steadman in Jones, 1995]



Figure 2.78: Screen from *SimCity 3000* - “creation”

Early video games were made by one or two lone hackers. In the 1990’s, people within the computer gaming community like to think of the computer game industry as the “New Hollywood.” Indeed, in some ways this may be an apt description. These days, major video games cost hundreds of thousands, if not millions, of dollars to make and employ producers, programmers, artists, musicians, actors, etc., over several years.

Of the most common complaints leveled at video games: that they are too shallow, too violent, or too anti-social, I agree with none. My own criticisms with the state of computer games are twofold: first, the computer game industry remains trapped in the same genres which were first introduced thirty years ago (SimCity being a rare exception). Second, as previously mentioned, computer software will remain a second-rate artform until notions of authorship can be established. Several game designers are now credited with unique personal vision, *Ultima's* Lord British, *SimCity's* Will Wright, and *King's Quest's* Roberta Williams. However, this is not enough; three directors does not a film-festival make. [Steadman in Jones, 199]

Though these games do not have the feel of a singular artistic vision, they do have something equally strong. The games have a dedicated audience thousands strong who feel intimately connected with environments they themselves have explored, inhabited, created, or destroyed.

Chapter 3: Digital Performance Systems

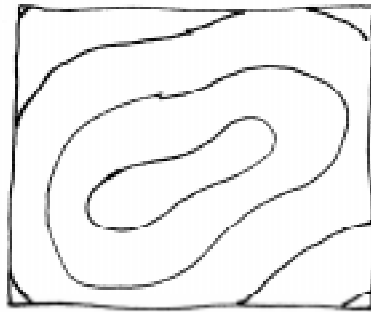


Figure 3.1: Sketch of audio/
visual performance sense

The Modulator Initiative has evolved through the analysis and creation of performance systems. In this section we examine a number of digital performance spaces which lead to the development of the present system.

3.1 Dimension7

This thesis is informed in large part by work done by the author before arriving at the MIT Media Laboratory. In this section we examine one such project integrating computational artforms into the context of a traditional musical performance.

The *Dimension7* project was carried out in collaboration with Om Recordings artists Soulstice in San Francisco in February 1996. The music group Soulstice consists of two keyboardists and a drummer.

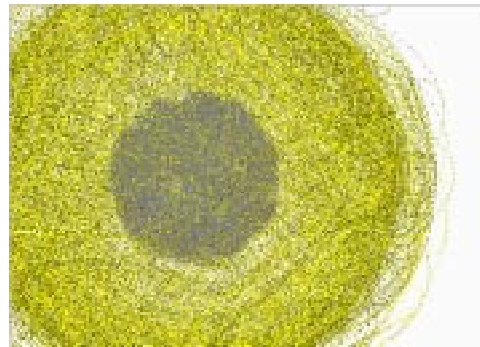


Figure 3.2: Still from
Dimension7 system

For the *Dimension7* project, the musician's keyboards and drums were connected to a Pentium personal computer via a midi interface. The *Dimension7* project was executed in very tight control of the computational medium, entirely written in low-level c code. This custom software transformed the musicians actions on their instruments into unique three-dimensional visual forms. In addition, the system allowed these forms to be transformed dynamically live in performance using a keyboard and mouse by an additional performer. The resultant visual forms were projected on a large screen above the performers on stage. Visual fireworks fired from the edge of the musician's instrument with intensely infinite color.



Figure 3.3: *Dimension 7* interactive animation formed by musician's keyboard

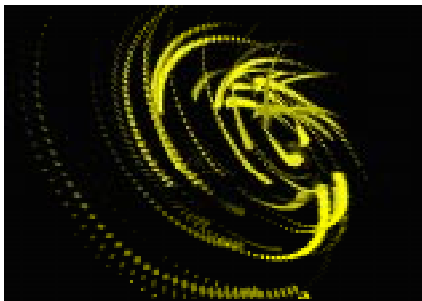


Figure 3.4: *Dimension 7* animation formed by musician's drumming

The musicians found that these projected reinterpretations of their actions pushed their performances in directions they could not have anticipated. For example, the drummer, in seeing his beats transformed into visual patterns, found that he discovered new audio rhythms while exploring visual patterns.

3.2 Plastic Jazz

The *Dimension7* work, though conceptually well ahead of its time, is in execution quite uncontrolled. Shapes of various sizes and colors shoot into space and pile one on top of each other. Upon arriving at the Media Laboratory we attempted to rethink some of the unbounded nature of the *Dimension7* project. We recognized two deficiencies of the *Dimension7* project which we believe led to the unbounded sense of this piece:

First, there is no definitive formal language for the *Dimension7* project. Form and color were used to great effect, but appeared arbitrary. At the Media Laboratory, we resolved to limit our set of forms to only the most basic, discover the nature of these forms in motion, and then work outwards

Second, in *Dimension7* the motion of objects in space is tied to conventional three-dimensional mathematical constructs and thus, not appropriate for the expressive nature of the work. Objects rotated in perfect circular orbits or traversed straight lines. To address this issue, we set out to develop custom input devices and motion algorithms for precise, yet expressive motion control in three-dimensional space.

With these goals in mind, we embarked on the *Plastic Jazz* project. *Plastic Jazz* strives to give a performer the ability to craft malleable instruments of form in space. Similar to a jazz musician combining and layering musical compositions, this work allows a visual performer to pick and layer instruments of form dynamically.

Physical Interface Design

In choosing an input device for our new system we looked at several options. As with the *Dimension7* project, incorporating an existing musical instrument appears to be a seductive prospect. By using a musical instrument, the audience

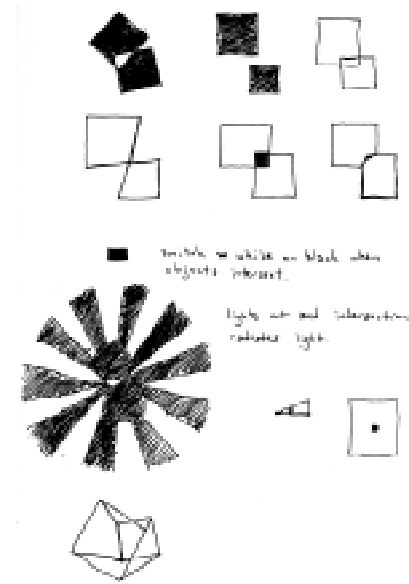


Figure 3.5: Studies of intersecting squares

may immediately connect the process of musical performance to our new conception of visual improvisation. However, as shown in the commercial audio editing systems reviewed above, one must strive to avoid forcing an existing interface into a new medium. Also, the goal of the *Plastic Jazz* project is not to visually represent the process of playing a musical instrument nor the sound emanating from an instrument. Therefore we abandoned assimilating a preexisting musical instrument into our system.

The conventional mouse input offers a number of dramatic advantages as an input device for performative interfaces. The mouse is extremely easy to monitor, it is connected to almost every computer, and provides an extremely fine grain of detection. Lastly, a great many people are highly proficient in their control of this motion-capture device thanks to innumerable hours spent controlling desktop metaphors.

The advantages of the use of the mouse as a motion-capture input for performance are perhaps best exemplified by John Maeda's *Flying Letters*. Here the everyday mouse gives the user the ability to dynamically configure moving compositions, where poetry emerges not merely from word meanings and configurations, but from the elegant transformation of human motion.

The mouse has one primary difficulty as an input device for three-dimensional systems: it can only record position in two dimensions. In addition, a primary ambition of *Plastic Jazz* is to draw very explicit connections between the form of the physical and visual interfaces and the human and computational motion uniting the two. Thus, we designed and built a custom physical interface.

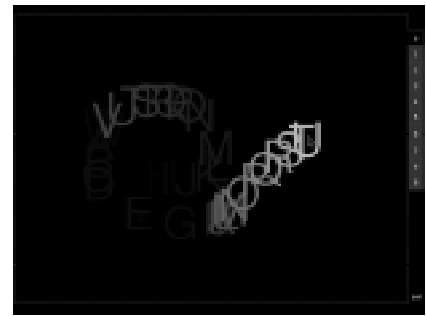


Figure 3.6: John Maeda's
Reactive Square

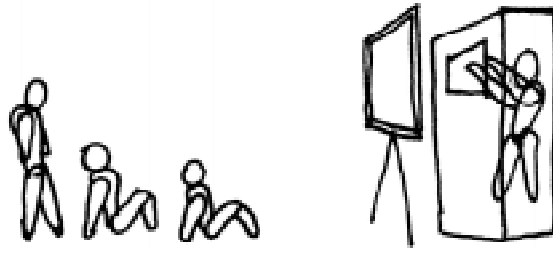


Figure 3.7: One common mode of performance, the puppet show

In considering the design of a new interface for controlling three dimensional forms we kept returning to the notion of a puppet-master, able to precisely direct her animated characters through rods and wires connected to both her hands. To get this level of control of our forms, we resolved to make a two-handed interface which could detect accurate position and rotation in three-dimension space.

We chose to construct two hand-held cubes which employed the *Flock of Birds* sensors from Ascension Inc. The *Flock of Birds* is a magnetic position and rotation detection-device commonly used for motion-capture applications. The *Flock of Birds* functions through the use of a single hub which emits a strong magnetic field. Individual passive sensors then detect their position and orientation within this magnetic field. The *Flock of Birds* sensors are highly accurate when within their magnetic field and have an update rate 60Hz.



Figure 3.8: Custom physical interface

In the construction of our cubes, we hoped that each cube would have a weight such that the user knew she was holding something with a purpose. We aimed for the weight of a baseball as opposed to a golfball. Our initial tests with steel, and then aluminum blocks revealed that the *Flock of Birds* sensors are extremely susceptible to interference when near metal. Thanks to this discovery, we quickly ruled out metals of any kind and moved on to hard wood.

Each cube is constructed using a single block of maple wood cut to shape and painted matte black. A rectangular space is milled out of one side of the cube to permit the *Flock of Birds* to fit inside the cube. A cover of thin plastic is attached to this side of the cube using two screws to hold the sensor in place. A single red button is mounted on a second side of each cube. These two screws and single button, though made of metal, did not appear to affect the accuracy of our sensors.

Visual Interface Design

As previously mentioned, Paul Klee developed some of the most rigorous studies of the true nature of form in this century. His particular work examining overlapping, intersecting, and interpenetrating forms gives particular insight in the study of three-dimensional computational interfaces. In Figure 3.9, Klee shows quite clearly the transformation of the zero-dimensional point to the three-dimensional body. Klee tells us:

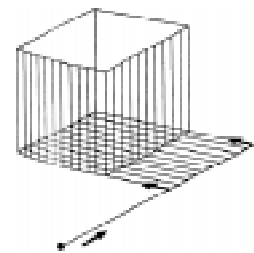


Figure 3.9: Demonstration of evolution from point to line to plane to body, Paul Klee

Point. The point [is the] primordial element, all pervasive.
Line. A point discharges its tension towards another point. The causal principle is [that each element will inherit the] reciprocal tension. Essence of a dimension. One-dimensional element.



Figure 3.10: Point and line, Paul Klee

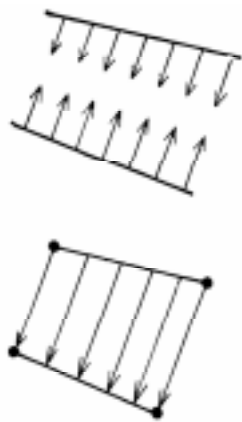


Figure 3.11: Tension and discharge of tension from line to line, Paul Klee

Plane. Tension between line and line results in a plane. Essence of two dimensions. Two-dimensional element.



Figure 3.12: Body, Paul Klee

Body. The line moves and produces the plane; the plane moves and the body comes into being. Essence of three-dimensions. Three-dimensional element. The cube is a balanced synthesis of three definite dimensions and as such [is] the normative symbol of corporeality. [Klee, 1961]

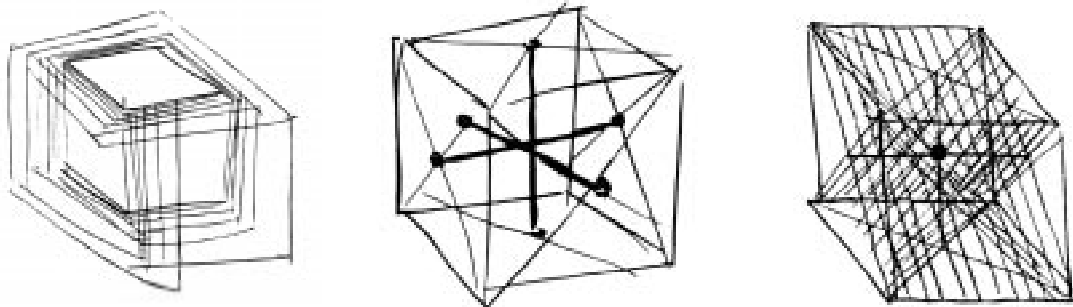


Figure 3.13: Spatial, three-dimensional and transparent; Inward, three dimensional and body inside; Purely inward, body innermost, Paul Klee

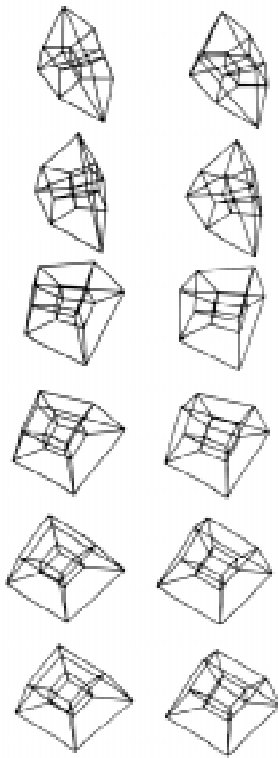


Figure 3.14: Successive stereo pairs of four-dimensional hypercube, Michael Noll

Thus the tension between elements of one dimension form the basis for the construction of elements of the next. What can we then expect from the tension between three-dimensional bodies? Klee suggests some possible avenues for exploration with his next drawings detailing the interpenetration and division of common territory, studies in two-dimensional movement, and spatial tension internal to transparent three-dimensional forms (Figure 3.13).

For the most part, we are left to our own devices. Klee was without implements for sketching beyond the two-dimensional faculties of pencil and paper. In fact, even today our faculties for examining overlapping, moving, transparent bodies are quite limited in the third dimension. The designer may use clay to sculpt static works or glass to construct transparent structures. Though these mediums remain limited in their scope for exploring the further dimensions.

Today's supercomputers and tomorrow's desktop machines have tremendous capabilities for the display of two and three-dimensional forms in real time. A great deal of work has been done in the realistic creation and display of forms, yet as seen in the examples noted above, the navigation and manipulation of these spaces remains frustratingly

rigid. We lose the joy of pliability in our interactions with the computer when lost in a cacophony of visual iconic references. There is no grace. Mastery of the digital medium provides the means by which one may control the entire space of interaction dynamically, freed from the prevailing trends of interface design.

Emulating Paul Klee's methodological work along these lines, we created a computational environment for examining the tension between three-dimensional bodies. To this end, we built the visual interface shown in Figure 3.15. In the *Plastic Jazz* interface, the user examines the interplay between two cubical forms which are under their control. At the center of the screen, one finds the two forms of primary concern. At the base of the display, the user sees an array of cubical forms partially obscured by a translucent grey bar. Each of the cubical forms has a predefined length, width, height, color, and transparency.

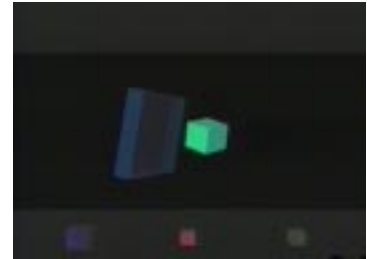


Figure 3.15: *Plastic Jazz* graphical interface

The external appearance of the physical cubes mimics the geometric forms on the screen. The movement and angle of our cubes in physical space maps directly to the equivalent motion of the cubes in three-dimensional computational space. As the computational cubes travel and rotate, three-dimensional forms appear which designate their previous positions and angles, thus accentuating the changing locations, orientations, and interrelationships of the primary forms. These “history” forms are marked off every 0.5 seconds and their transparency increases as they age to 3 seconds, at which point they disappear (Figure 3.17).



Figure 3.16: *Plastic Jazz* physical input

We incorporate two different modes of interaction for the buttons on the physical cubes. In the first mode, a press of the button on the side of the cube inflates the corresponding cube on the visual display.



Figure 3.17: *Plastic Jazz* forms in motion, intersecting

In the second mode, a press of the button on the right cube allows the user to examine a different form with unique size, color, and transparency characteristics. One of the two primary computational objects associated with the physical cubes floats down to the bottom right of the row at the base of the screen, the objects on the bottom row slide over to the left, and a new cube is brought to the center of the screen. The position and orientation of this new cube are now associated with one of the physical cubes.



Figure 3.18: Selecting new forms for examination, *Plastic Jazz*

A press of the button on the left cube causes the system to take a three-dimensional snapshot of the objects in motion. This sculptural photograph of the current state drifts to the upper left of the grey bar at the top of the screen. The snapshot is also saved out to disk in the form of a three-dimensional description file. This file can be read into any number of commercial 3D modelers for later examination.



Figure 3.19: Three-dimensional snapshot/ sculpture capture



Figure 3.20: *Plastic Jazz* installation

The Plastic Jazz system was shown in a dramatic installation in the Media Lab “cube” space. As shown in Figure 3.20, the video display was set on a black stand at eye level for a standing spectator. This configuration hints at a metaphysical console video game. The two cube input devices were hung with black wires from the mesh ceiling of the space 30 feet above. Each movement of the input cube both transforms the objects on the screen as well as the 30 foot wire support, thus connecting the interface to the environment.



Figure 3.21: A moment with the *Plastic Jazz* system

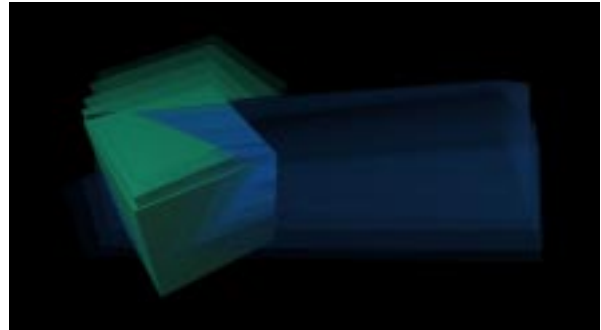


Figure 3.22: *Plastic Jazz* sculpture

3.3 Pliant Architecture

Now that we had developed an environment for understanding the intersection of three-dimensional computational forms, we felt it necessary to examine the possibilities offered by the pliable form. We aimed to create computational form able to bend to the performer's will, while still maintaining a primary conception and unique characteristics of the given form. Whereas before we had investigated the tension between two mobile, yet unchanging forms, we then considered the single transforming body.

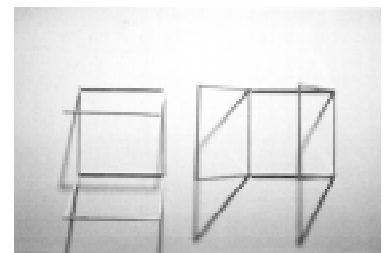


Figure 3.23: Gianni Colombo's *Elastic Relief*

Other artists have explored kinetic form through the use of electromotors. Gianni Colombo built subtle reliefs in which surfaces slowly ripple and cubes deform. These installations explore the way light transforms objects and spaces. He also constructed "elastic reliefs," (Figure 3.23) which can be transformed by the spectator as well as flexible environments with elastic stretched from wall to wall and ceiling to floor. [van Berkum, 1986]

To induce the maximum spectator participation, we resolved to incorporate simulated physical algorithms into our research system. The Media Laboratory has a long history of projects that have advanced the understanding of physics in the graphical interface. One particularly beautiful example of such a project is Karl Sims' *Particle Dreams* which evolved from work while Sims was a masters student at MIT. In this amazing animation, Sims creates and transforms between computer-generated three-dimensional shapes, whose outlines are defined by points in space. The motion of each of the points which make up the larger shapes is controlled by elementary particle physics algorithms. At the time this animation was made in 1990, each frame was rendered using a Connection Machine CM-2 supercomputer. Thanks to the speed of computers in 1998, we are able to do such calculations in realtime. [Sims, 1990]



Figure 3.24: Karl Sims,
Particle Dreams, 1988

We embarked on a similar study to that of Sims though with the added concern that the overall forms transmute interactively. To emphasize the dynamism of a form in transformation as opposed to the dynamism of a static form, we again chose to examine a cubical body. As an initial study, we arranged a series of small cubes in a regular pattern to suggest four sides of a larger cube. Each side of this larger cube was made of a grid of 6 cubes by 9 cubes. This gives us a total of 180 discrete objects in motion.

Each small cube in the *Pliant Architecture* was given a unique particle physics data structure relating to its distance from an imaginary point in the center of the larger cube. The distance from the center point is directly proportional to the mass of each small cube.

A movement of the physical input device is transmitted to the interactive particle physics model of the small cubes. Thus, small cubes nearer the center of a given side of the larger cube react more quickly to the movements of the input device than those close to the edges.



Figure 3.25: Entering data with *Pliant Architecture*

As with the *Plastic Jazz* project, our focus in *Pliant Architecture* is to emphasize transformed human movement in three-dimensions. To this end, we employ one of the two physical input devices developed for *Plastic Jazz*. In *Pliant Architecture*, a gesture with the physical cube causes the appearance of the overall cube to bend as shown in Figure 3.25.

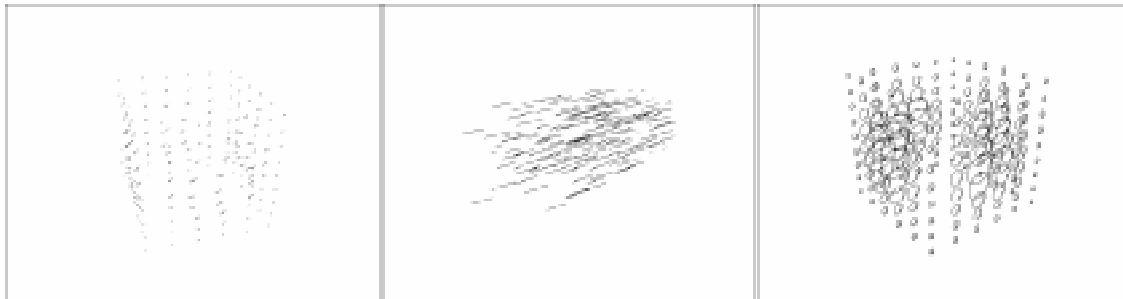


Figure 3.26: Recorded motion sculptures, paths of cubes moving in space

In addition to the live, interactive piece, we did a number of reinterpretations of the captured kinetic sculpture. In the first of these studies, we recorded the paths followed by the small cubes as a human participant manipulates the physical cube. These motion paths form are then plotted in three-dimensional space as independent structures. Figure 3.26 shows a number of these “recorded motion sculptures.” Each one embodies the unique characteristics of the brief moment of human motion which created the sculpture.

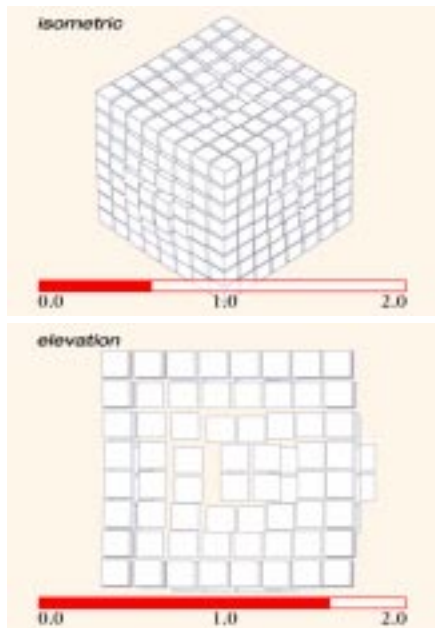


Figure 3.27: Isometric and elevation of the Pliant Architecture in slow motion

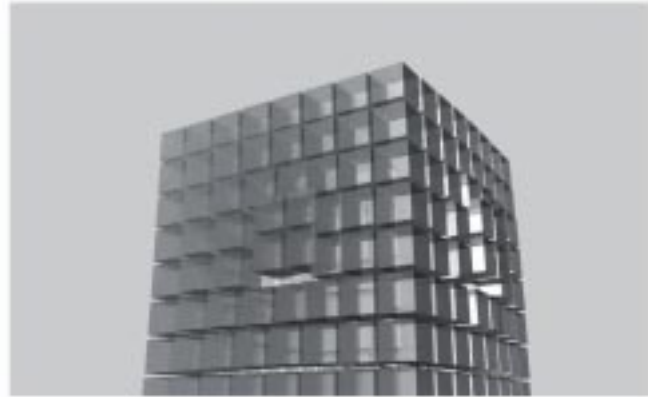


Figure 3.28: Still from final animation of Pliant Architecture

In the second reinterpretation, we created a linear animation from two seconds of recorded motion. The mutation of the overall cube structure is replayed and slowed to 1/30 of its original speed. To achieve this dramatically slowed movement, we applied a 3D spline interpolation to the original data set. The gradually bending form was imagined as a future building in our animation.

3.4 Dynamic3

Can the use of one's computer file manager be a performance? This is precisely the proposition of the Dynamic3 project.



Figure 3.29: Dynamic3 staging area for interaction

In this study, two physical cubes control a malleable, hierarchical data structure represented as abstract objects on a virtual stage. By using interactive particle physics, virtual objects react in ways that correspond to our existing assumptions about our physical world.

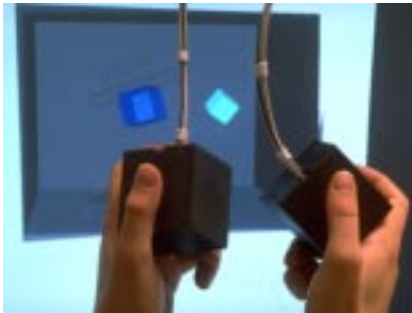


Figure 3.30: Dynamic3 physical and graphical interface

The two physical cubes control the position and orientation of their virtual counterparts. A change in position of one of the physical cubes exerts a force on the parameterized particle physics model. By variably weighting various data objects, we introduce a new paradigm by which the dynamic/motive reaction of information structures gives subtle cues as to the contents of the object.

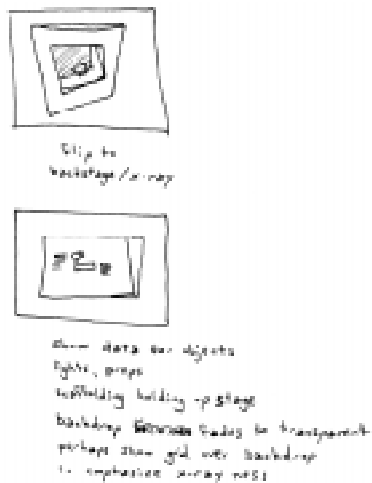


Figure 3.31: Sketch of design for stage action in Dynamic3

Our approach emphasizes the subtleties of interaction. Not what is seen, but what is felt. To amplify the expression, we present a physics-based computational model. In addition we use a fluid physical interface. Through repeated exploration, the user is able to feel differences in reactivity and modulate his or her decisions accordingly, without sacrificing aesthetic or metaphorical continuity.

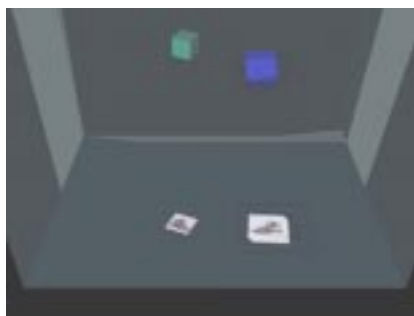


Figure 3.32: Dynamic3 "data shadows"

Each of the virtual structures can have child objects. These child data structures are represented as wireframe substructures which trail smoothly behind their parent. Mounted on each physical cube is a single, small button. This button serves as the control for changing states of the system. By pressing a button, sub-objects in the hierarchy transfer from one hand to the other. Each child is also a structure which can in turn contain subordinate children.

A firm gesture towards the rear of the virtual stage allows the user to view more detailed information about the contents of the abstract data objects. The camera tracks and pans to a position behind the translucent rear stage. From this position, the camera acts much like an x-ray machine, revealing the information structure within each data object. The objects' shadows transform into projections of their corresponding data. In this way we maintain the primacy of our original abstract forms, while seamlessly examining their concrete data shadow.

Modern data graphics can do much more than simply substitute for small statistical tables. At their best, graphics are instruments for reasoning about quantitative information. Often the most effective way to describe, explore, and summarize a set of numbers - even a very large set - is to look at pictures of those numbers. [Tufte, 1983]

In this piece, we view interface mechanics and interface design as a single process. The physical underpinnings of the interface architecture allow for the possibility of functionally and aesthetically graceful interactions.

3.5 Proposed Interface 1: MidiVis

With the deficiencies of these systems in mind, we sought to outline a series of interfaces for a truly performative interactive musical space. Building upon the Dimension7 projects, our first interface focussed on the common midi protocol for musical intercommunication. This proposed interface, which was not seen through to functioning implementation, we called MidiVis.

The MidiVis system seeks to unite the processes of playing musical instruments and editing the resultant midi tracks into a single, continuous performance. This performance may involve a single user manipulating prerecorded midi tracks, simultaneously playing a musical instrument and manipulating the midi track computationally, or manipulating the midi tracks of other musicians playing their instruments. In addition, it was our goal that multiple users could manipulate separate portions of the midi tracks simultaneously.

Figure 3.33 shows a typical screen of the proposed MidiVis graphical interface. Here we see the midi tracks for three instruments become sculpture. The representation for sound in the MidiVis system is tightly aligned with the midi format (form from medium). In the midi protocol, sounds are defined by notes for a given instrument. Each note is described with a precise pitch, amplitude, and duration.

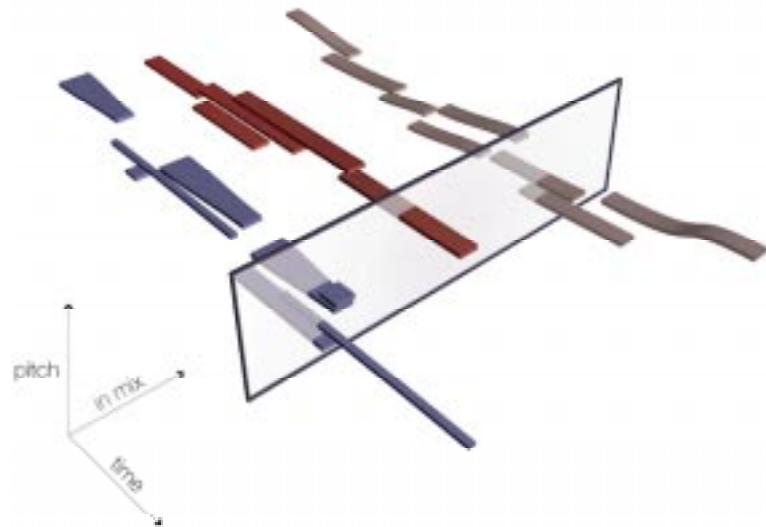


Figure 3.33: Perspective view of *MidiVis* sound interface

In the MidiVis scheme a given midi instrument is represented as ribbons flying through space. Each ribbon depicts a single midi note. The length of the note represents its duration, the width of the note its amplitude, and the position of the note in the Y direction tells its pitch. When the system is live, the musical strips move in the direction specified by the tag “sound movement”. Figure 3.33 shows the groups of notes passing through a window representing the current moment in time, in other words the current sound one hears.



Figure 3.34: *MidiVis*, side view

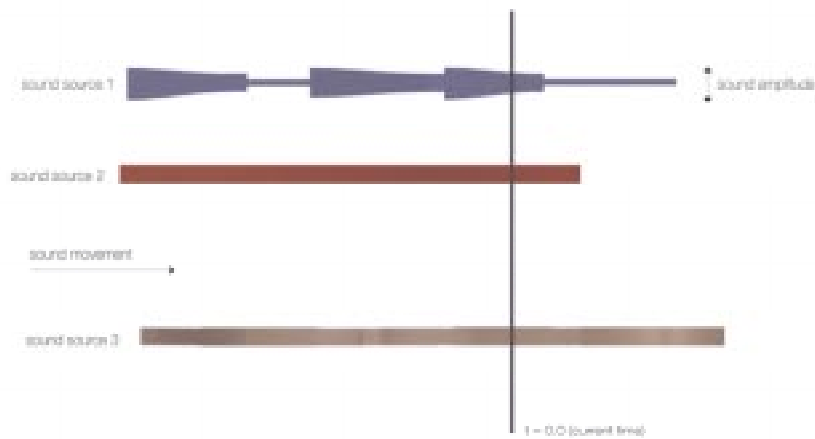


Figure 3.35: *midiVis*,
top view

A user may shift her camera to view the system from the side (Figure 3.34), thus obtaining a view of the MidiVis system similar to more traditional sound editors: time flows from left to right across the display, note pitch increases from bottom to top. Alternatively, the camera may be positioned from above (Figure 3.35). The user now sees a similar but somewhat altered view: time still flows from left to right, but now the amplitude of a given instrument can be seen in the width of each colored bar.



Figure 3.36: *MidiVis*,
front view

Viewing the front of the system (Figure 3.36), the user can see each note strike the “current time” window, not unlike watching the keys of a player piano in action. Note that from each of these views the participant is presented with a unique, *meaningful* representation of the transforming musical structures.

By specifying regions in space one affects both the visual and auditory output. In Figure 3.37 we see a filter which causes a two groups of sound ribbons to be pulled together. This action causes the pitch and amplitude of each note to be affected according to this new representation. These changes register immediately. The interactor may affect sound regions before, after, or during the current moment in time. Although we only depict a single filter here, one can easily imagine that by specifying spherical, conical, or cylindrical regions, entirely different effects would ensue. For example, a cylindrical filter may blow a gust of soft wind from beneath a the notes of a given instrument, tossing their ribbons upwards and thus increasing the pitch of the notes.

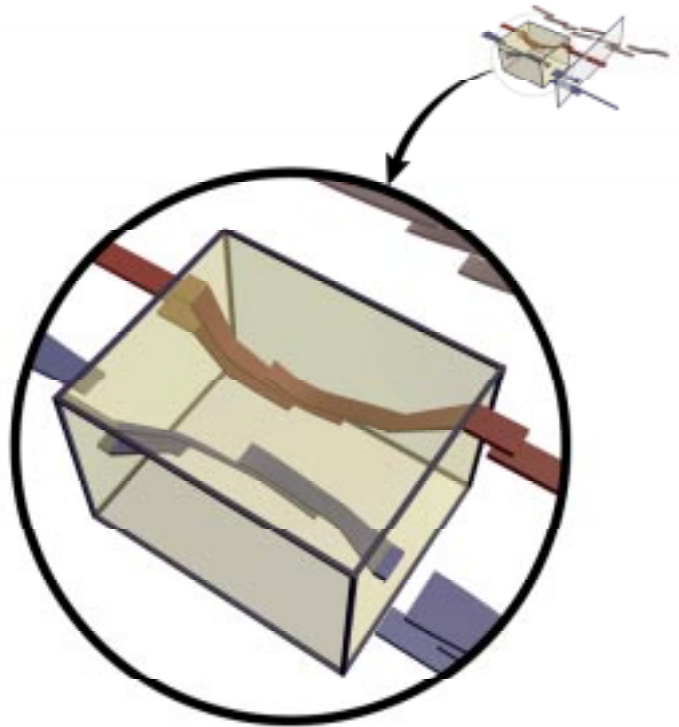


Figure 3.37: "Dynamic filter" in *MidiVis*

Our design sketches of the MidiVis proposal begin to demonstrate a harmony between modern human-computer interface techniques and a freer visual interpretation of musical structure.

3.6 Proposed Interface 2: Sound City



Figure 3.38: Early sketch of Sound City project

For our second prototype sound and visual interface we focussed on digitally sampled audio as opposed to midi audio. Again we employed three-dimensional visualization techniques for exploring visual representations of sound space.

Digitally sampled audio is of an entirely different character from the midi protocol representation. As previously described, midi sounds are defined by notes of set length, pitch, and duration for a given instrument. Midi data may only be recorded from an instrument that subscribes to the midi protocol (commonly electronic keyboards, drums, and synthesizers) and can only be played back on said instruments or a computational emulator. On the other hand, digitally sampled audio can represent almost any sound source, even those that have nothing to do with instruments: a soprano singing, cars honking, children playing, etc. Various researchers have studied the possibility of extracting midi information from a digitally sampled source. Unfortunately, for now this only seems to work for a single instrument playing a known song. [Martin, 1996]

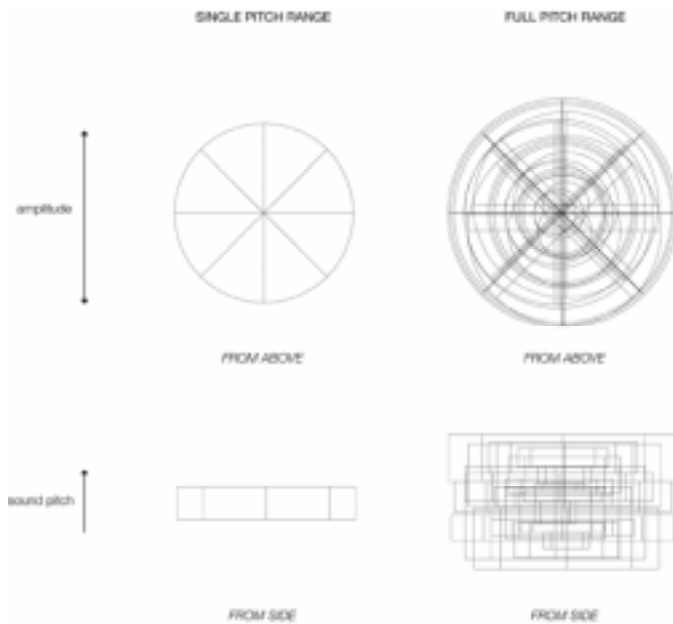


Figure 3.39: *Sound City* schematics for sampled sound representation

This additional path of communication raises issues of implementation on both the audio and the visual end. The audio system must have more robust digital signal processing capabilities to tell the visual system how to represent the audio streams. A given audio source is split into a range of pitches.

In turn, new visual representations must develop with the additional data communication. Each pitch range is represented as a single cylinder. The amplitude of a pitch can be seen in the diameter of its corresponding cylinder. A complete sound stream forms a solid structure of combined cylinders. The sound structures bend and transform based on the changing audio source:

The additional path of communication allows for significantly more elegant visual constructions. This city of cylindrical forms becomes a living, interactive diagram of the realtime music editing. By adding beat-tracking and sound analysis code developed by Eric Sheirer of the MIT Media Lab, the sound structures would intercommunicate based on the synchronicity of their musical rhythms:

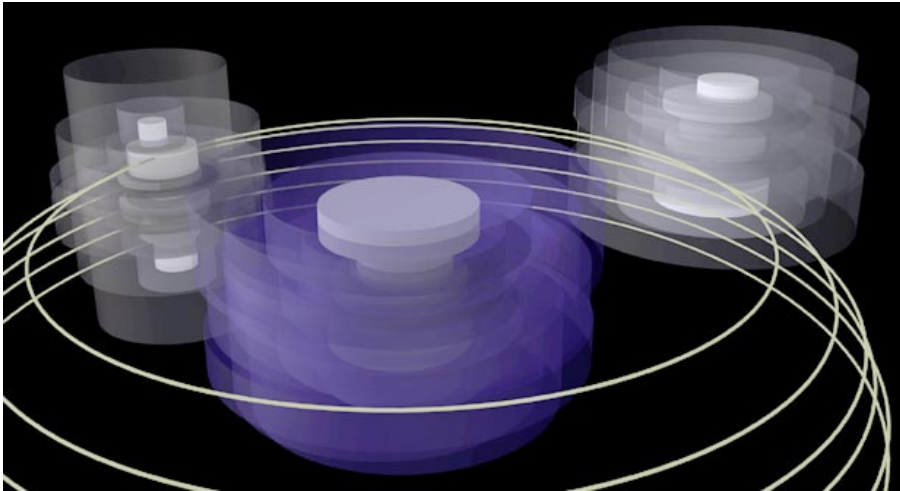
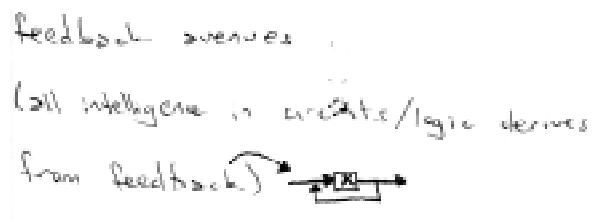


Figure 3.40: Proposed
Sound City interface

In the above image we see some of the same pictographic building blocks from the design language transferred into a sketch of a proposed performance space. This can be seen as an advantage of the language. It is an intuitive step to transform the sketch to a system model diagram.

3.7 Transducer



Transducer is a digital system for live, audio-visual performance specifically designed to explore issues in parallel with the development of the Modulator language. The system allows a performer to build constructions of sampled audio and computational three-dimensional form simultaneously. Each sound clip is visualized as a “playable” cylinder of sound that can be manipulated both visually and aurally in real-time. The Transducer system demonstrates a

creative space with equal design detailing at both the construction and performance phase.

The intent of this system is not to act as a traditional tool for editing audio, nor as a three-dimensional modeler. Transducer asks one to envision a space where the process of editing and creating on a computer becomes a dynamic performance which an audience can easily comprehend. The content of this performance may be sufficiently complex to elicit multiple interpretations, but Transducer enforces the notion that the process should itself be a fluid and transparent expression. We akin the simple interactive functions of Transducer to that of a DJ's mixer, encapsulating the magic and freedom of disk-jockey performance with concise visuals that are clearly in tune with the music, though entirely unique to the medium with which the work is created.

The performer and audience of the Transducer system view a video projection which illuminates a screen hanging from above and listen to audio from two speakers. The performer acts upon the system with a single-button mouse, though there are no menus or buttons in the Transducer system. Like Robert Zeleznik's *Sketch* system, all actions are based on gesture recognition and on the context of the three-dimensional form being acted upon: all interface actions occur on cylindrical sound objects

Currently Transducer consists of two computers: a Silicon Graphics octane and an Intel PentiumPro 200. The Silicon Graphics computer handles visual computation and output and the Intel handles audio computation and output.

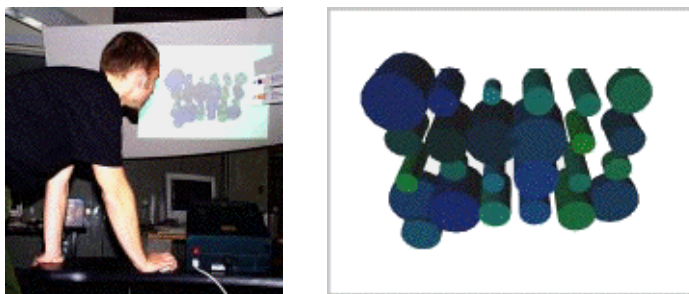


Figure 3.41: User interface and "palette" screen in *Transducer*

At first, the system presents a palette of cylindrical objects. As the user moves his or her mouse over each of the cylinders, he or she hears a sampled sound stream associated with that object. Each of the objects has a representative color and shape corresponding to the sound stream associated with it.



Figure 3.42: Live transition in *Transducer* interface

By clicking the left mouse button while over the object, the user selects the sound object to be manipulated. The palette of objects unfolds and drifts behind the camera (this transition can be seen in Figure 3.42), the sound associated with the chosen object begins to play, and the selected object moves to the “manipulation zone.” In this area, four types of mouse actions control the sound object as shown in Table 1.

MOUSE ACTION	EVENT
click + drag vertically up or down	increase/decrease sound frequency, incr./decr. height of sound
click + drag horizontally towards or away from object center	t/b sound amplitude, move object towards/away from center, t/b object transparency
short click on object	stop playing sound return object to palette
click anywhere not on an object	bring up palette of all sound objects with active sounds visible behind

Table 1: *Transducer* Interface Commands

By clicking the left mouse button while over the object, the user selects the sound/object to be manipulated. The palette of objects drifts behind the camera, and the selected object moves to the “manipulation zone.” While in this area, by clicking on the object and dragging the mouse up or down, the user both stretches and contracts the object and increases and decreases the frequency of the associated sound. By clicking on the object and dragging the mouse to the left and right, the user affects the transparency of the object and the amplitude of its sound stream. Clicking anywhere not on the object brings up the palette of all sound/objects with the sound visible behind. Additional sound/objects can be previewed and any number of sound/objects can be brought into the manipulation zone.

In this way a single user or performer is able to build simultaneous visual and audio constructions in realtime. The user can examine interrelationships between multiple, diverse sound sources and a corresponding visual form.

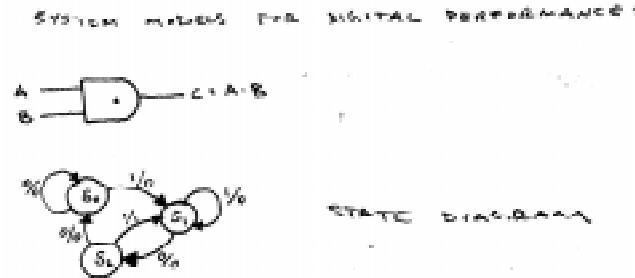
Data Abstraction

The form of each object gives clues as to the data (or sound information) contained within. Each of the sound clips is rated along six axes according to the characteristics of the sound stream: jazz factor, ambient factor, speed factor, vocal factor, hipHop factor, and germanRisque factor

In addition, the motion of each object is modeled with a unique physics model. Thus two objects react differently to user input based on the internal “mass” and “drag” of each, as in the Dynamic3 project.

By parameterizing each aspect of the performance, we can create continuous (rather than discrete) relationships between the different dataspace (audio and visual). Since our interface to the system is primarily gestural and our primary focus is on the ability to immediately conceptualize expression, we are not concerned with the precise representation of sound or visual structure. Rather, we are concerned with an acceptable simultaneous approximation of both, able to be realized in real-time. Along these lines, the system makes extensive use of interactive physics algorithms for scene transition and effects.

Chapter 4: Modulator Language



In his essay “The User Interface is **The** Language,” Michael L. Dertouzos suggests three chief requirements for interface languages of the future:

1. The language should incorporate communication between multiple users and multiple computers over networks
2. The language should include capabilities for controlling multiple tasks
3. The language should have input/output capabilities for multiple devices

“In short, future languages should include integrated access to this broader environment [beyond the desktop computer], for the simple reason that these capabilities will be present and should be controlled by everyone.” [Dertouzos in Myers, 1992] To illustrate his vision for the future computer language, Dertouzos gives us Figure 4.1, for which the caption reads “Target of new languages: networked multiprocessors that are easy to use.”

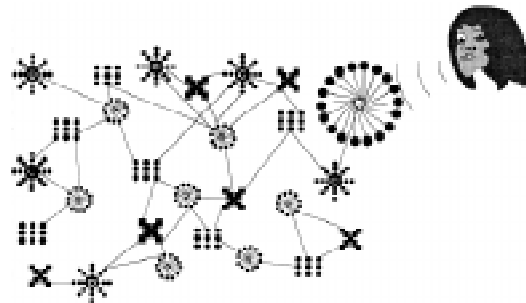


Figure 4.1: Dertouzas' language for interface design: networked multiprocessors

In the Modulator Initiative we propose a vision for understanding the future human-computer interface similar to Dertouzas' proposition, though with some subtle but dramatic differences.

In Dertouzas' figure, the user communicates with a network of interconnected multiprocessor machines. If the language truly is to become the interface, not only the multiple processor, but the processor itself must disappear to the user. We suggest that instead of interacting with processors, people interact with sensors. The interface language of the future should consist of networks of sensory input/output devices.

In the future, computers will not live in boxes. Computation will roam free in a multitude of distributed, interconnected processes. Our relationship with computation will be based entirely on dynamically configurable interfaces of interchangeable parts.

In the Modulator Initiative we present a unified visual language of interface design based around the concept of interaction as performance. The language consists of seven characters: visual input, visual output, audio input, audio output, tangible input, touch (or haptic) output, and the human.

With these simple building blocks we have constructed a three-dimensional living diagram for exploring future interface design issues. Using this system we can categorize preexisting interactive systems, build prototypical interface system models, and examine issues of feedback and communication inherent to interaction in the next millennium.

In this section we define the basic structure of this new visual language. Since this is a thesis detailing design process, we will structure our examination of the language chronologically, and in the process tell the story of how it was made.

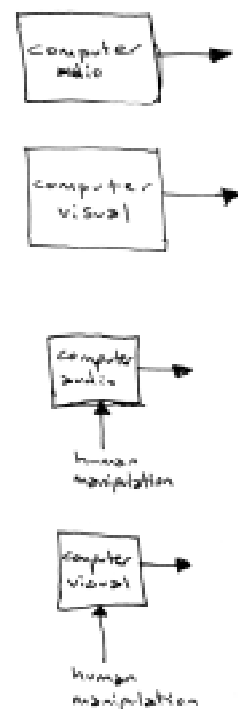


Figure 4.2: Sensory inputs and outputs

4.1 Hypotheses

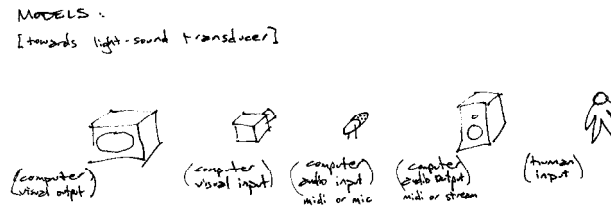


Figure 4.3: Initial sketch of interface actors

The Modulator Initiative evolved as a process of rethinking earlier digital systems for sound and visual performance. Though much of the previous audio/visual work for performances was, from an engineering standpoint, completely controlled and perfectly crafted, the vast majority of work felt either completely limited to the point of boredom or frustration or entirely unbounded, psychedelic. To us this situation appeared to be not a problem of form or content *per se*, but of inappropriate feedback.

Norbert Wiener, in his introduction to feedback and oscillation, details a simple but poignant example of the types of feedback we will examine:

Let us consider a signal tower on a railroad. The signalman controls a number of levers which turn the semaphore signals on or off and which regulate the settings of the switches. However, it does not do for him to assume blindly that the signals and the switches have followed his orders. It may be that the switches have frozen fast, or that the weight of a load of snow has bent the signal arms, and that what he has supposed to be the actual state of the switches and the signals - his effectors - does not correspond to the orders he has given. To avoid the dangers inherent in this contingency, every effector, switch or signal, is attached to a tell-tale back in the signal tower, which conveys to the signalman its actual states and performance...

Notice that in this system there is a human link in the chain of the transmission and return of information: in what we shall from now on call the chain of feedback. It is true that the signalman is not altogether a

free agent; that his switches and signals are interlocked, either mechanically or electrically, and that he is not free to choose some of the more disastrous combinations.[Wiener, 1948]

Every interactive human-computer interface may be thought of as a chain of feedback with a human link. The chain of feedback of most human-computer interfaces is no more sophisticated than the stop-and-start example provided by Wiener: We type in a word on the keyboard. We cannot assume that we typed in the word correctly or that the word was recorded properly by the computer. The computer screen shows that the we typed the correct letters and that the computer followed our orders.

Computational systems that link together audio and visual streams for realtime interactive performance have significantly more complex chains of feedback. Yet adequate methods for describing such systems have not kept pace with the technological development of the systems themselves.

Our goals for embarking on this project were the following:

1. By creating a pictorial visual language for describing interface g able to better analyze and characterize human-computer interfaces a: interfaces.
2. This new system might serve to impact the field of information repr tion at large by dynamically depicting live processes in a novel wa

4.2 Static Representation

In its initial state, the language had two main grammar types: actors and actions. By nature of being interactive, the computer both listens and speaks, i.e. there is an “input” and an “output.” An actor was either a human or an computational element. A computational element was one of the following: “visual output,” “visual input,” “audio output,” or “audio input.” There was one single action called “communication.” According to this language, an interface may be defined by a visual combination of actors and actions. We can then begin to map the chains of feedback in human-computer systems.

Our first step in the evolution of this language involved the construction of a program for making two-dimensional, static representations of human-computer systems using our pictorial language.

Initial sketches for each of the actors and for “communication” were drawn in Adobe Illustrator using a pen stylus. A custom plug-in for Illustrator was created in c++ to export and import these drawings to and from java. Finally a custom piece of software was composed in java for constructing independent system models. Using these techniques, we designed the following large (approximately 5 feet by 3 feet) poster:

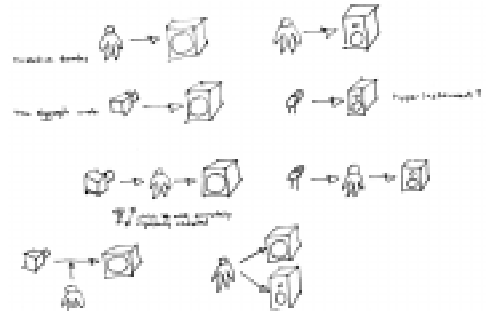


Figure 4.4: Early definition of language elements

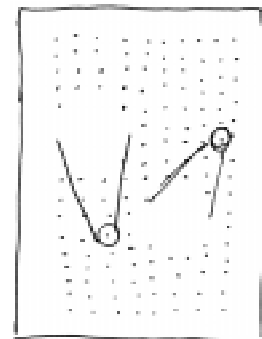


Figure 4.5: Method for comparing/contrasting individual models out of a large sample

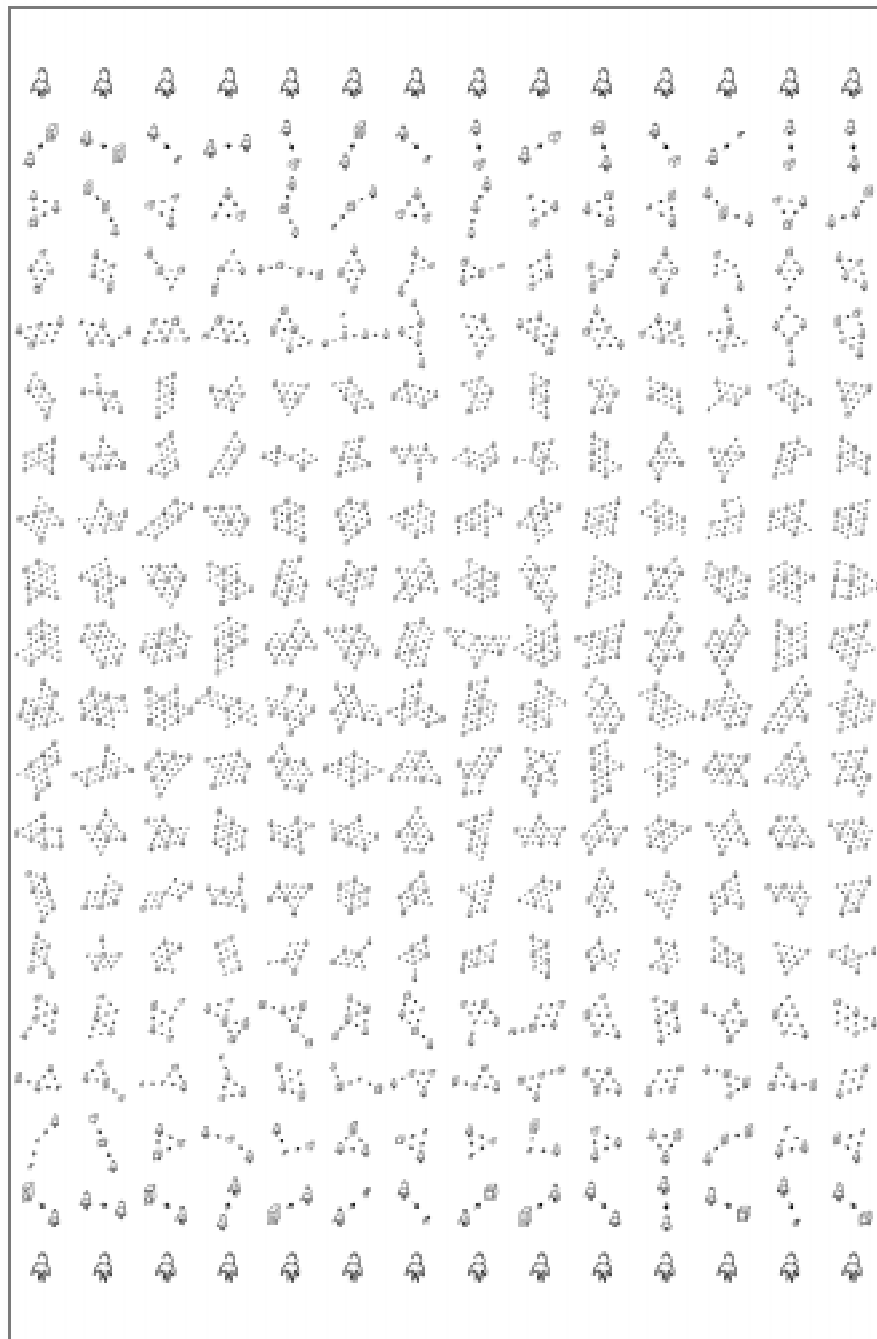


Figure 4.6: Poster study of proposed system model language

On the vertical axis we see models of increasing and decreasing complexity. With each row an additional actor is added to the system. On the horizontal axis we see assorted iterations with identical complexity. In this way we are able to visualize a vast field of interacting actors and actions.

Each of these systems was created at random and without regard to whom or what each actor was communicating with. In this way, things get a bit out of control. However, as with Braitenberg's vehicles, one may find that on occasion it is time to "give chance a chance." [Braitenberg, 1984]

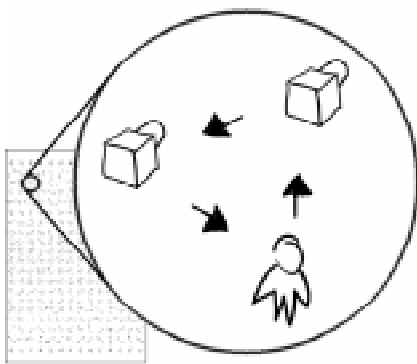


Figure 4.7: Questionable network: human communicating with video inputs

Many of these individual system models are almost completely nonsensical. For example, let us examine this model wherein a human actor is watched by a video input: fine. This first camera is then communicating to a second video input: possible, but questionable. Finally, the second video input is communicating back to the human: certainly wrong. A human can get very little out of watching a video camera.

We also find a human communicating to an audio output (the equivalent of talking to a stereo speaker), an audio output communicating to a visual output, and a visual output communicating to an audio input: all highly impractical if not unrealizable systems.

However upon close examination, one may find that an equal number of these system models create entirely possible, even surprising interactive diagrams. One of the first things one notices in these systems is the organization and status of the human actors. We see a system with two people communicating one to the other. We then discover three people communicating, the first to the second, second to the third, and third back to the first.

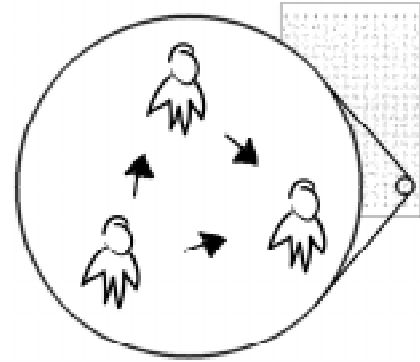


Figure 4.8: Viable network: humans communicating with each other

Very early on in its development, we determined (as did Wiener) that a symbolic representation for the human could and should be a primary participant in these diagrams. Though we could not have predicted the dramatic impact this single decision would have on the characteristic of our resultant diagrams. In this two-dimensional plane of interacting agents, equal importance is given to a system of intercommunicating humans and intercommunicating machines or any combination thereof.

Brenda Laurel succinctly states the importance of human depiction in interface representation: “The human is an indispensable ingredient of the representation, since it is only through a person’s actions that all dimensions of the representation can be manifest. To put it another way, a computer-based representation without a human participant is like the sound of a tree falling in the proverbial uninhabited forest.” Laurel goes on to restate the importance of this seemingly self-evident point: “...reconceptualizing what computers do as representing action with human participants suggests a design philosophy that diverges significantly from much of the contemporary thinking about interfaces.” [Laurel, 1993]

The intention of this poster study was not to act as a final system model generator. Rather, the poster served to raise many questions about the method of constructing living diagrams and the rules which might govern their functioning.















		DESTINATION						
								
SOURCE		✓		✓		✓		✓
		✓		✓				
			✓		✓		✓	
		✓				✓		
			✓		✓		✓	
		✓						✓
			✓		✓		✓	

Table 2: Rules of communication for interface actors

At this point we were able to begin to formulate a network of feasible and infeasible communications. Table 2 shows this network. Communication sources run down the left (or vertical) axis of Table 2. Communication destinations run across the top (or horizontal) axis of the table. If the corresponding intersection between a given source and destination contains a check in its box, then this is a permitted communication path.

The human actor can communicate freely with other humans. The human can also output to a visual input or audio input. The human is not allowed to output to a visual output or audio output as he needs some method of getting his transmission to these sources.

Alternatively, the visual output can communicate with the human. Humans do watch video screens, after all. The visual output can also send information to the visual input as a video camera can be pointed at a video screen. The visual output is not allowed to communicate with the audio output or audio input. No valid communication path can be established to these actors. The audio output functions in a similar manner to the visual output.

The visual input can send communication to both the visual output and audio output. In the same way, the audio input communicates to the visual and audio outputs as well.

Perhaps the most obvious component lacking from the poster study is a method of input other than a microphone or video camera. For example, where do the common mouse or keyboard fit into this systemology? Certainly our system language must at least encompass the most common interfaces! This deficiency will be addressed as we expand the notations for our system language.

4.3 Dynamic Representation

Sensory Communication

Human sensory perception can be divided up into five categories: “sight”, “hearing”, “touch”, “taste”, and “smell.” Using our senses, humans communicate with each other and with our environment. Through these five senses, human beings negotiate their internal processes with those of the outside world. Thus, the main focus of communication for our interface study will be sensory. This emphasis on the senses brings our analysis of computational interface mechanics in line with the same interface mechanics which drive human communication. In this way, we have a very distinct criteria with which to define our interface actors. Each actor shall represent a sensory input or output:



Figure 4.9: The human actor

Visual output is the most common output method for computer systems; the most prevalent devices for visual output being the everpresent CRT as well as the video projector. Several research projects have attempted to extend the possibilities for visual displays beyond that of the emissive light display. Joe Jacobsen and his group at the Media Laboratory endeavor to create new types of paper displays that can be transformed using electrical currents. [Campbell, 1996] Andrew Dahley’s “Ambi-



Figure 4.10: The visual output actor

ent Fixtures” present several novel methods visual display using custom-made mechanical devices. [Dahley, 1998]

Visual input devices are far less common than their output counterpart. Some examples of visual input devices include video cameras, three-dimensional location sensors, and infrared cameras.

With the explosion of the “multimedia” personal computer in the early 1990’s, more and more people have access to audio output and input devices for human-computer interaction. However, for the most part these devices serve small, supporting roles in the human-computer dialogue. Obviously, the most common audio output device is the loudspeaker and the most common audio input device is the microphone.



Figure 4.11: The visual input actor

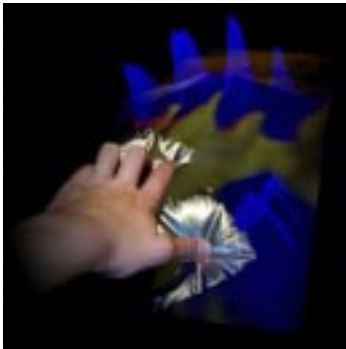


Figure 4.12: Tom White’s “liquid haptic” touch input device



Figure 4.13: *InTouch* touch I/O device

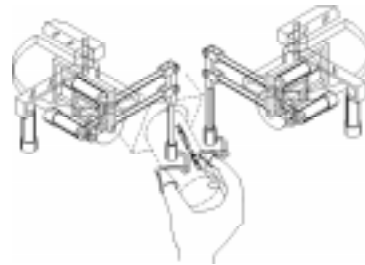


Figure 4.14: *PHANToM* touch I/O device

Though we commonly place most interface design importance on our visual sense, it is our sense of touch with which human beings modify and manipulate the world around us. The sense of touch may be split into two categories: the tactile sense, through which humans sense shapes and textures; and the kinesthetic sense, through which humans sense movement or force in muscles or joints. [Hannaford in Barfield, 1995] For the purposes of our study, we link these two under the single “touch”

umbrella and give this sense unique input and output actors. Though we leave it as an interesting intellectual study: the output of “tactile” sensation.

A “touch” input may be a mouse or keyboard, of course. It can also be a joystick, a trackball, a touch screen, or a liquid interface such as Tom White’s bladder [White, 1998], among others. The *InTouch* system developed by Scott Brave and Andrew Dahley [Brave, 1997] as well as the Phantom [SensAble, 1993] demonstrate examples of touch systems which incorporate both input and output mechanisms. Though in each case a computer mediates the input and output systems. For our purposes we separate these dual-purpose input-output devices into their input and output components as individual actors.



Figure 4.15: The touch output actor



Figure 4.16: The touch input actor

Other interface agents which may interact with humans in alternative ways: chemical or biological, sensing internal changes in the body. Some of these newest interfaces pose unique challenges to interface designers. Many of these may be grouped under the umbrella of “touch” interfaces as they require some mechanisms which touch a participant in some way (bio-feedback sensors must touch the skin, the pill monitoring body functions must be inside one’s stomach, etc.)

The careful reader will notice we have still neglected two of the five basic human senses. In the 1920’s, director David Belasco experimented with the use of realistic odors in the performances of his plays; apparently he gave up the practice when he discovered that the smell of frying bacon completely distracted the audience from the other aspects of the performance. In the 1960’s, Morton Heilig created a stand-alone arcade machine called *Sensorama*. *Sensorama* provided stereo-scopic images, haptic feedback, as well as environmental smells of New York City. [Laurel, 1993]

In addition, one can quickly point to contemporary environmental artworks which employ smell as a large factor in the creation of powerfully active spaces. Based on these examples, the sense of smell holds dramatic potential for enlivening reactive environments.

In addition to these examples, the previously mentioned “Cerebrum” project employed extensive use of the sense of taste to create a complete “sensory-kinetic multimedia experience” in the late 1960’s. [Youngblood, 1970] Though for now we will leave the senses of smell and taste for further research.



Figure 4.17: The audio output actor



Figure 4.18: The audio input actor

Modulator Interface Design

At this point in the development of the theoretical underpinnings driving our system it seemed a natural time to advance a new computational research platform for examining our evolving language. Thus, the Modulator project was born. We began constructing this three-dimensional system based upon the visual language evolved from the two-dimensional poster project. The user is initially presented with an array of small cubes aligned in a hexagonal pattern. This array serves to denote the X-Z plane or “ground.”

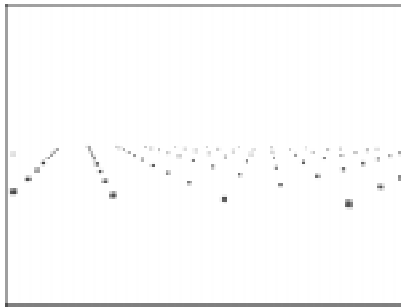


Figure 4.19: The Modulator system before actors enter

By clicking and dragging on the middle mouse button, the user can freely control their view (or “camera”) into this world (Figure 4.19). A drag to the left or right repositions the camera’s position about the origin ($X=0$, $Y=0$, $Z=0$). All the while the camera (and thus the user) is looking at the origin. A quick mouse movement to the left or right, brings the camera closer to the origin as it flies by to its new position. A slow drag with the mouse moves the camera about slowly, keeping its distance from the origin.

Each click of the right mouse button brings a new actor into the system. The new actor may be one of the seven types defined above. The actor enters from a large distance away from the origin at an arbitrary angle. As it enters, the actor rights itself and moves quickly towards the origin.

These actors look identical to those of the two-dimensional poster except that they are now given a short depth in the Z direction. In this format, if one were to rotate the actor 90 degrees away from the viewing plane the shape would appear as a vertical line. In order to avoid a group of lines which should look

like representations for interface actors, we rotate each actor as the camera moves so that it remains perpendicular to the viewing plane.

The movement and actions of each individual actor are governed by some very simple rules:

1. An actor tries to move towards the origin.
2. Each actor must remain at least a distance d away from all other actors.
3. An actor may communicate with another actor if the distance between the two is less than r (where r is greater than d).

When an actor moves within distance r from a neighboring actor, the Modulator establishes a communication path if a connection can be made according to the rules previously defined in Table 2. These communications are represented as three-dimensional directed arrows. If two neighboring actors can both output to each other, then a two-headed arrow is shown.



Figure 4.20: Living diagram in modulator system

After an actor has entered the system it can be moved around to a new position. By clicking on an actor with the middle mouse button and dragging the user can move about the actor. The actor is moved according to the mouse's position mapped onto a plane perpendicular to the direction of the camera's viewing angle. As the selected actor is moved about, the other actors move out of the way, to maintain their minimal distance.

A right click on any of the actors causes that actor to gradually pull apart from the other actors in the system model and then fly off in the distance.

Feedback

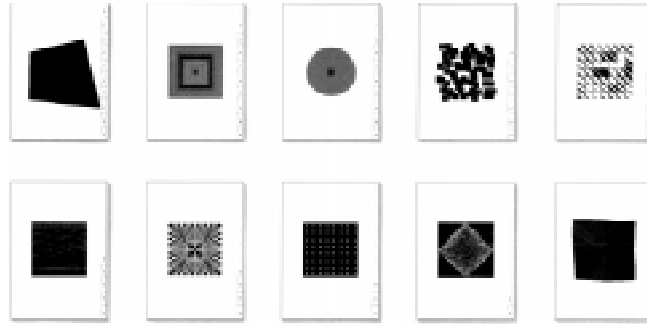


Figure 4.21: John Maeda's
Reactive Square

Intelligence requires feedback. For example: our nervous system virtually runs on feedback alone. Our brain cannot tell our leg exactly where to move to, it must receive continual feedback from sensors/sensations from the leg telling the brain position and status. [Wiener, 1948] The understanding and control of feedback is of primary importance in the design of interactive systems.

To assist in our analysis of the feedback capabilities in the *Modulator* system it is perhaps appropriate to enlist John Maeda's *The Reactive Square*, the project through which I was first convinced of the sincerity of Professor Maeda's movement. *The Reactive Square* is a printed book and it is a computer program.

About the book Maeda says, "Originally this book was to be called Ode to Malevich, as I was influenced by the early Russian Suprematist Kazimir Malevich and his courage to abandon all decoration in pursuit of the simplest of forms, such as a single black square. I began thinking about black square which would exist only on the computer, and how the medium of the computer could allow tremendous potential as a starting point. But I was not sure in what manner and mode these squares should be interactive." [Maeda, 1997] Eventually, Maeda found the answer. He wanted his young children to interact with the computer not by struggling with a mouse or keyboard, but by talking or singing. Therefore, *The Reactive Square* is ten studies of a square which responds to sounds received by the computer's microphone. In a lesser artist's hands, this

could be intensely boring, a poor screen saver at best. Under Maeda's control, it is not: when one speaks into the computer's microphone, the graphic square leaps in unexpected, yet completely controlled ways.



Figure 4.22: System model of a single user interacting with Maeda's *Reactive Square*

From the standpoint of our system diagrams, this digital performance is fairly straightforward to model. For a single person "playing" the reactive square, the performer communicates by making sounds (speaking, knocking pots and pans together, etc.) which are picked up by the computer's microphone. This signal is then transformed into a visual representation which can be viewed by the human thanks to the computer's visual display. This system can be modeled with the two-dimensional sketch in Figure 4.22.

As previously stated, every interactive human-computer interface may be thought of as a chain of feedback with a human link. Here the chain of feedback is easy to discern. Audio data leaves the human, becomes visual data and returns. Each step in the chain has a part in the overall flow of information through the system. Maeda's design skill allows the feedback loop to remain highly controlled given wide fluctuations in the within the user's range of input.



Figure 4.23: Modulator diagram of two users performing *The Reactive Square*

Now, let us for a minute propose that two of Maeda's children want to play *The Reactive Square* simultaneously. Our system model diagram grows significantly more complex. The new model can be seen in Figure 4.23. Consequently, our examination of feedback in this system expands in complexity as well. Instead of the single chain of feedback experienced by the single human in our previous system model, we now have five feedback loops to describe the experiences of the two participants, as seen in Figure 4.24.



Figure 4.24: Feedback loops for two people using *The Reactive Square*

The first two sketched diagrams in Figure 4.24 show feedback loops not unlike those from our first examination of *The Reactive Square* system. Each person interacting with the system can see the visual representation of their audio input to the system directly. The third feedback loop is that of purely human communication. The participants each discuss their use of the system and modulate their actions. In the fourth and fifth diagrams we note that not only do the humans talk to each other, but by communicating to the other human one causes that other person to alter their actions in the system as a whole. Person two may tell person one to yell loudly; person two can then see this action transformed into the corresponding visual output.

As we began to examine more complex interactive systems with our sensory-based language, the analysis of chains of feedback grew exponentially more complex with each additional actor.

Therefore, we sought methods for determining the feedback loops experienced by human participants in a given interactive performance. We employed directed-graph theory to build a custom algorithm for listing these feedback chains. We then integrated the three-dimensional Modulator application with our previous two-dimensional post-script system.

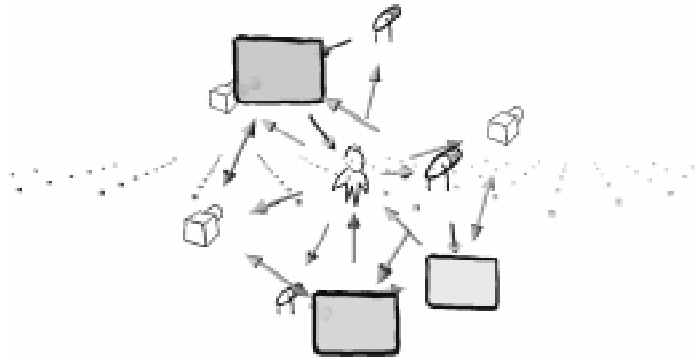


Figure 4.25: Modulator diagram of a single human in a complex media environment

In this way, by pressing the “f” key while using the Modulator system, one may get a printed list of each chain of feedback for each of the human participants in the current system model.

To demonstrate the use of this facility, we shall examine the slightly more complex media environment shown in Figure 4.25. This system includes three audio inputs, three visual inputs, and three visual outputs. On first glance, this may seem to be a large number of input and output devices. Indeed, this system uses more I/O devices than the average home computer. However, if we recall Bill Viola’s *Stopping Mind* with its four video projectors and four loudspeakers or his *Image Bank* with six video cameras and six monitors, the system in Figure 4.25 seems less extreme. [Viola, 1995]



Figure 4.26: One feedback loop experienced by the human

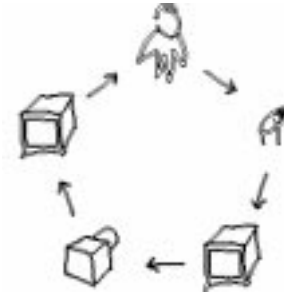


Figure 4.27: The same feedback loop expressed as two-dimensional diagram

In Figure 4.26, we see one of the feedback loops experienced by the system participating in this digital performance. Figure 4.27 depicts the corresponding two-dimensional representation of the this same feedback loop. Finally, in Figure 4.28 we see the complete list of chains of mechanical feedback felt by the human actor in the Modulator diagram of Figure 4.26.



Figure 4.28: Computed feedback diagrams for the single human in the system

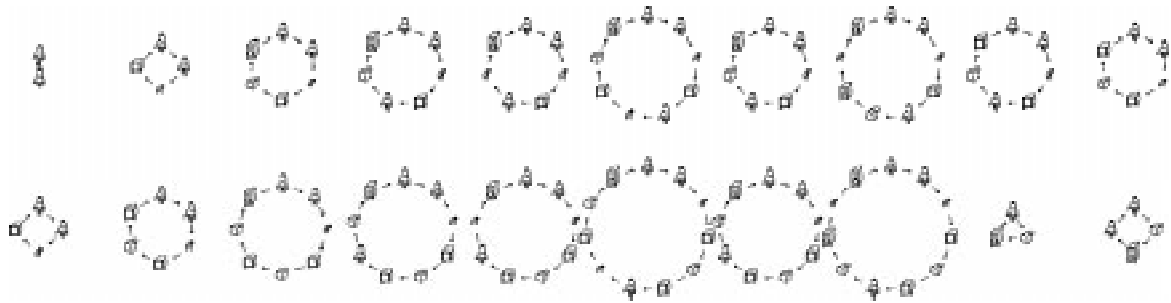


Figure 4.29: Analysis of feedback for a single human in a more complex interactive system

Using this capability, our Modulator system can describe feedback chains in interactive systems of unlimited complexity. As with the recent movements to display and make sense of large sets of information. The Modulator system allows a user to remake and examine large and complex sets of real-world processes.

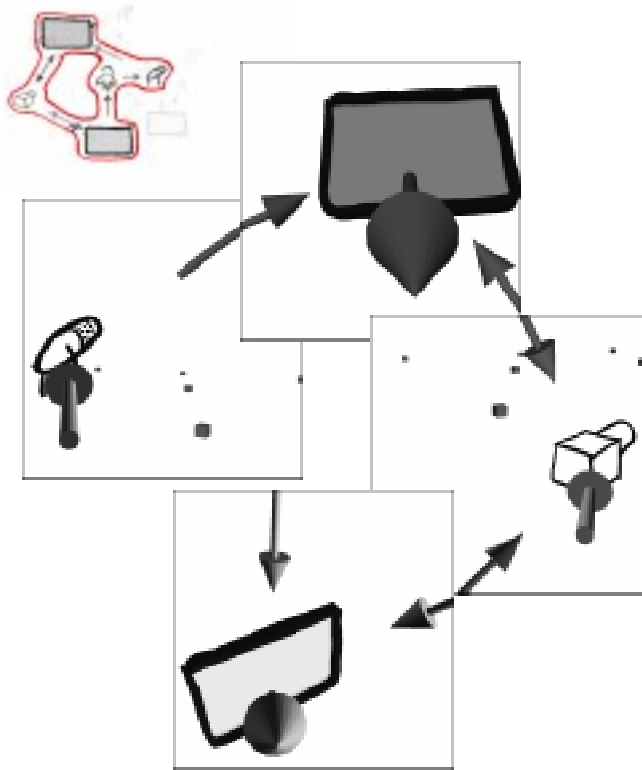


Figure 4.30: Feedback loop seen from human actor's point of view

Viewpoint

One distinguishing aspect of three-dimensional environments (both real and virtual) is that the user may assume several unique viewpoints into a space. As Modulator is a three-dimensional system, we make use of this by allowing the user to play the role of any of the human actors in the system.

A click of the left mouse button on a human actor causes the camera to sweep in and “put on the clothes” of the chosen actor. In this position, a click and drag on the middle mouse button allows the user to rotate his or her viewpoint to survey the surroundings. The user now sees the current diagram with the eyes and hears the dia-

gram with the ears of this human actor. The user sees neighboring communication paths entering and/or leaving this actor and hears those sound outputs which connect. A click with the left mouse button anywhere not on an object causes the camera to return to a global viewpoint.

Native vs. Transformative Communication

We are now in the position to define two classes of interface communication: “native” and “transformative.” When a method of communication passes through natural elements it may be considered “native.” Where a communication passes through electronic or computational components it will be called “transformative.”

An audio wave emitted from a speaker which passes through air and is then heard by a human is natural. The same audio wave received by a microphone which is then sent to an video output may be considered transformative, as it must pass through a level of electronics to change into a visual signal of some kind. We can thus rewrite our source and destination table, taking into account our new techniques for classifying interface mechanics (Table 3).















		Destination						
								
Source		N		N		N		N
		N		N				
			T		T		T	
		N				N		
			T		T		T	
		N						N
			T		T		T	

Table 3: Rules of communication including native and transformative designation

Now we see a “N” for native communication and a “T” for transformative. At every point in this table where a “T” occurs, we may insert some level of computational mediation or mutation.

At this point, one may raise the argument that in real life people can communicate with one another digitally, yet this type of communication is not directly represented in the Modulator model. Truly people can communicate using bits as a part of a communication chain. But if one person is standing next to another, the only way they can communicate directly with bits is if they call out “1,0,1,1,0...” to each other. A second, more effective way they might communicate using bits is if they somehow get their human means of communication into some digital device. The second method requires a device which captures their human sensory communication and transforms it into computational communication. This is the crux of “native” and “transformative” communication.

Communication Channels

Based on our concept of sensory communication and previous definition of “native” and “transformative” communications, we have derived one final classification system for communication in the interface, the notion of “channels”.

“Native” communication may pass through the air or physical objects with visual, auditory, or tactile means. We then define four channels for native communication: visual channel, audio channel, tactile channel, and multimodal channel. A human communicating directly with another human may use a combination of visual, audio, and tactile communications. This agglomerated type of communication we call “multimodal”

In our current system we are concerned with digital media, therefore “transformative” communication may only occur using a “digital channel”. Certainly one may also think of a digital channel as “electrical,” as at its most basic level digital computation is composed of electrical pulses. In addition, many functions of digital computers can be replicated with purely electrical components. And yes, finally and as previously stated, one might construct a mechanical computer. All the same, we will use the term “digital channel” keeping in mind these other possibilities under the larger “digital” umbrella. The “transformative” channel also opens up interesting possibilities for the introduction of entirely different forms of transformation than those offered by digital media. Here one may envision a biological or chemical channel for changing sensory input to additional states.

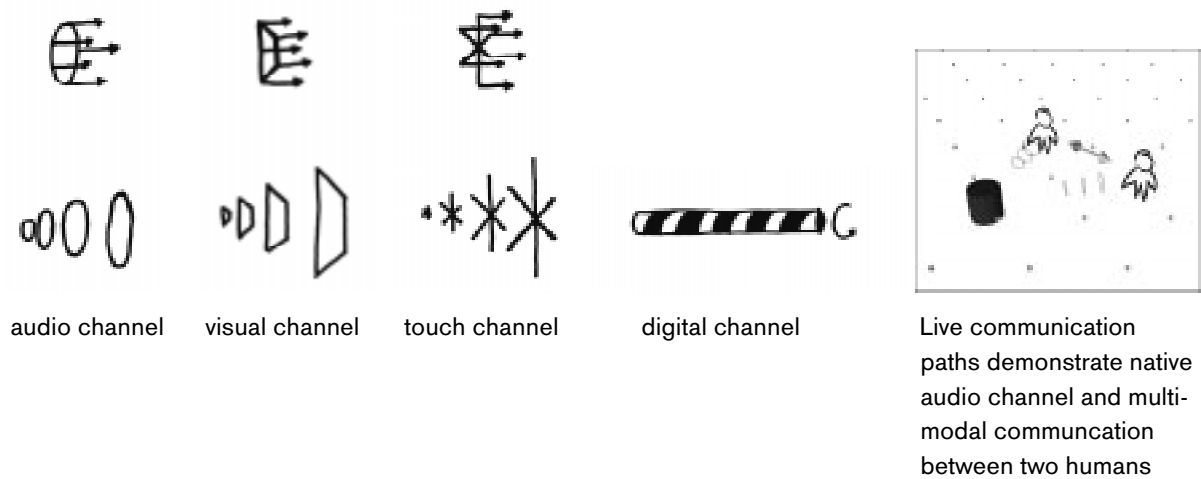


Figure 4.31: Communication paths between actors in the *Modulator System*

Thus with this new definition, a communication path between interface actors may be one of the following: visual channel, audio channel, tactile channel, multimodal, or digital channel. We now modify our three-dimensional Modulator system to accommodate these communication categories. We designate communication in the visual channel as rectangles moving in space, the audio channel is represented as circles (Figure 4.31) and the tactile channel as a series of short lines emanating outwards. The multimodal channel remains as a directed arrow. The digital channel is represented as a spiral, barber pole-type form spinning in the direction of communication.

Form of Interaction Space

In his research “Exploring an Organic Information Architecture,” Matthew Grenby asks us to imagine an ancient “data” oak tree on a hill:

From a distance, you can appreciate the over-all form of the tree: the strange symmetries of the branch network, the shape and color of the canopy. As you come nearer, you notice the leaves rustling in the breeze... You see faint traces of charred bark, indicating that at one point this tree survived a fire in the field... We know a significant

amount about this particular tree. No text. No numbers. All this information has been gleaned from a quick examination of only the formal qualities of an object: its shape, not its numbers.” [Grenby, 1998]

According to Grenby humans have deep innate relationships with shapes. We can detect subtle changes in shapes with which we are familiar. We develop attachments to particular shapes. Grenby claims that, by reshaping information and processes into discernible forms we enable users to employ their facilities for comprehending shapes, developed over the entire course of their lives, for the purpose of quickly differentiating characteristics of said information.

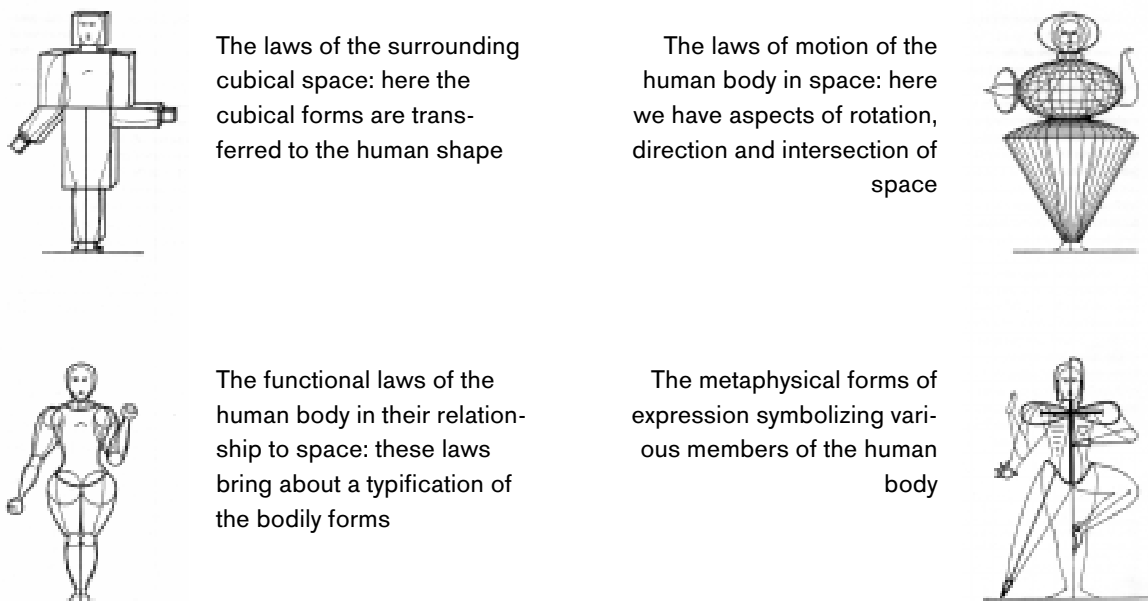


Figure 4.32: Oscar Schlemmer's conception of stage costume

Oscar Schlemmer might agree with Grenby's theory, were he alive today. In his figures and drawings, Schlemmer transformed information and processes of the human figure into form. In the upper left of Figure 4.32, the cubical forms define the human shape: head, torso, arms, legs. In the lower left, Schlemmer represents the functional laws of the human body, the egg shape of the head, the vase shape of the torso, and the ball shape of the joints. Next we find the laws

of motion of the human body in space: the aspects of rotation, direction, and intersection of space. In the last piece of this series, Schlemmer allows himself more creative freedom. Here he details the metaphysical forms of expression symbolizing various members of the human body.

Clearly simple qualities of forms tell us a great deal about the information and processes which cause those resultant shapes. Edward Tufte explains further that visual comparisons between a number of such forms allow us to quickly compare and contrast information, perhaps much more efficiently than an equivalent system based entirely on numerical information may allow. Shapes allow local comparisons within our eyespan.



Figure 4.33: The *Modulator* system creates a form about the diagrammatic elements

A press of the spacebar causes the Modulator system to create a three-dimensional shape about the current space of interaction. Figure 4.33 shows one such shape. Using such a visualization, we can quickly categorize human-computer interface configurations based on the organization of individual actors and communications.

Chapter 5: Analysis

In this section of the thesis we analyze the language in terms of its relationship to the Aesthetics & Computation movement. As various human languages have unique phrases which give us keys to the personalities of the culture which gave birth to the language, this new language exhibits features which allow us to examine the larger state of design and computation.

5.1 Applied

In this section we will apply the new language to works by the author to demonstrate the functionality of these linguistic building blocks. We define the mechanics of these works using the language we have established.

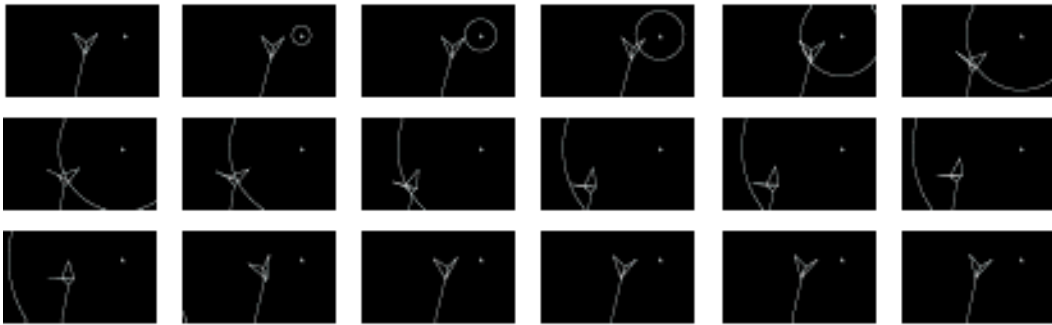


Figure 5.1: User clicks once in the kite interface, kite responds as if blown by gust of wind



Figure 5.2: Using the kite interface

Let us take a simple example first. This piece, designed in java for the world-wide web demonstrates the simple beauty of interactive physics. A human clicks the mouse and triggers a buffer of wind. The kite, an autonomous entity, reacts in a natural manner as it is modeled using dynamics algorithms.

We see a single human who can communicate with a computer visual via a physical input device. This visual output

communicates back to the human. Like Maeda's *Reactive Square*, the model is essentially a feedback loop through which the human and the computer visual each modulate their output based on the latest communication from the other.

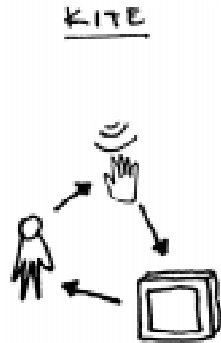


Figure 5.3: System model for the *Kite* piece, a single feedback loop

We then move on to additional select performance systems built by the author. The additional complexity of these systems serves as an ideal testbed for our *Modulator* language.

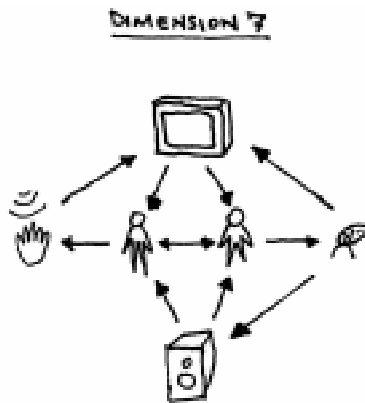


Figure 5.4: Sketched system model of the *Dimension 7* environment

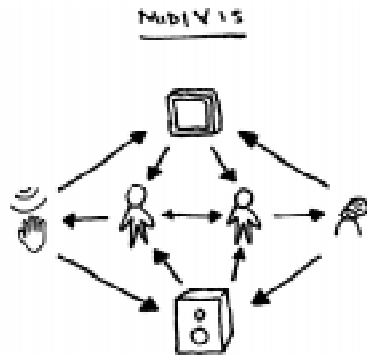


Figure 5.5: Model of the *MidiVis* system

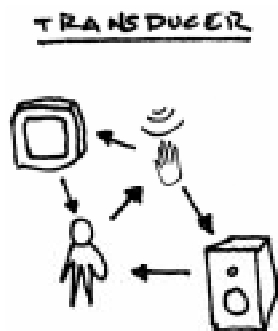


Figure 5.6: Model of the *Transducer* system

5.2 Sensory Throughput

During World War II, Norbert Wiener found himself called upon to join in the war effort. At that time, a soldier controlled antiaircraft guns through the direct application of precomputed range tables. Professor Wiener and Julian Bigelow embarked on a project to dramatically improve the accuracy of these guns in any conditions. Their solution was to turn the antiaircraft artillery into self-regulating robots in which the human links in the control chain were treated as though they were pieces of feedback apparatus. Though their designs were not perfected, the theoretical framework for their endeavor was circulated in the classified document: *The Extrapolation, Interpolation and Smoothing of Stationary Time Series*, which was later declassified and published as a book of the same name.[Eames, 1973]

Our systems are not so unlike those that Wiener dealt with. As previously discussed, our conception of the performative interface involves an emphasis of time and action in the interface functions. We may define a performative interface as more similar in actual interface functioning to many video games than to static audio/visual devices for computers. How many video games involve the (simulated) control of weapons against highly mobile foes? In this way, these video games employ similar simultaneous electro-mechanical control of body and mind faculties to Wiener's anti-aircraft weaponry.

After the war, Wiener transferred his efforts from weapons of destruction to the interplay of human and mechanical minds. In 1947 he writes: "The best we can do is to see that a large public understands the trend and the bearing of the present work, and to confine our personal efforts to those fields, such as physiology and psychology, most remote from war and exploitation. As we have seen, there are those who hope that the good of a better understanding of man and society which is offered by this new field of work may anticipate and outweigh the incidental contribution we are making to the concentration of power..." [Wiener, 1948]

Although we do not attach the same dramatic implications to playing war-like games and the actual process of war, we do seek to employ the technologies and methodologies "after the videogame" for our design concerns.

On the communication engineering plane, it had already become clear to Mr. Bigelow and myself that the problems of control engineering and of communication engineering were inseparable, and that they centered not around the technique of electrical engineering but around the much more fundamental notion of the message, whether this should be transmitted by electrical, mechanical, or nervous means. The message is a discrete or continuous sequence of measurable events distributed in time - precisely what is called a time series by the statisticians. [Wiener, 1948]

This raises a very interesting point for our interactive performance systems: how can one measure the flow of information or measurable events in our system models? Certainly we can attempt to count the number of bits that pass between communicating parts in the native and transformative pathways:

Let us first focus on the *Kite* piece, a concise, functioning feedback loop. The mouse input for the *Kite* piece has X and Y positions (16 bits) as well as a single button (1 bit), each of which is updated 60 times a second. This gives us 1980 bits/second through the touch channel. The visual display is monochrome, 640 pixels wide by 370 pixels high. This visual display is updated 60 times a second. Thus, the visual channel has 236,800 bits/second of throughput. The transformative channel must therefore “step-up” the lower-bandwidth touch channel to the higher-bandwidth visual by a factor of 120 times.

In addition, we can attempt to generalize bandwidth for native channels. The native visual and audio channels may be the easiest to discern. If one’s video display runs at 60 Hz, then we may say the display transmits 60 samples/sec * 640x480 pixels/sample * 24 bits/pixel = 442,368,000 bits per second. The same may be said for the data rate of a video camera. A CD-quality audio line carries 44,100 samples/sec * 16 bits/sample * 2 channels = 1,411,200 bits per second.

The touch channel is much more elusive to determine quantitatively. Using Fitts’ law, Langolf determined that the fingers receive 38 bits/second of information, the wrist 23 bits/second, and the arm 10 bits/second. [Langolf, 1976]

Though calculations of this type serve as interesting devices for contrasting various interactive approaches, they must be considered highly suspect in regards to overall data throughput for our system models. Just as we have given our generalized visual throughput eleven million times the throughput of Langolf’s touch bandwidth, one may construct visual outputs with a single LED or haptic devices with many precise outputs to various parts of the hand simultaneously.

Therefore our language study must remain primarily a polemic device, a touchstone for discussing aspects of interface design as opposed to a precise quantitative model of human sensory throughput.

5.3 The Performing Machine

It has long been clear to me that the modern ultra-rapid computing machine was in principle an ideal central nervous system to an apparatus for automatic control; and that its input and output need not be in the form of numbers or diagrams but might very well be, respectively, the readings of artificial sense organs, such as photoelectric cells or thermometers, and the performance of motors or solenoids. With the aid of strain gauges or similar agencies to read the performance of these motor organs and to report, to “feed back,” to the central control system as an artificial kinesthetic sense, we are already in a position to construct artificial machines of almost any degree of elaborateness of performance. [Wiener, 1948]

Here Norbert Wiener signals the development of artificial-intelligence robotics. In 1948, he envisioned a machine which would consist of a number of synthetic sense organs (inputs) and motors (outputs) which served as a “nervous system” for a machine of any degree of complexity. Using this type of simplified view of the robot, we may consider our performing machines as functioning “robots” as well, though our definition differs in three rather dramatic ways.

First, there is no definition of “internal” and “external” for our robot. With a traditional robot, sensors primarily monitor the state of the robot’s external environment and report this information back to the “brain.” The traditional robot has a shell casing of some kind demarcating regions internal and external to the robot and between the robot and the human. A Modulator robot is a sequence of intercommunicating senses and sensors monitoring states internally and externally.

Secondly, there is no single “brain” or “central processing unit” in our model. Rather, there are various parts in the chain (which we call “transformative”) that can be affected by discrete processors/processing components.

Finally, our conception of a robot includes the human as a primary component of the nervous system. There can be no performance without a human audience. The digital performance derives its strength from the abilities of its human participants and the design of its functioning digital parts.

After a talk, Marvin Minsky was once questioned by a particularly livid audience member, “Is it true that you have said that machines will take over the human race?” To which Marvin replied, “Yes, it is true. Machines have already taken over the human race. See the digital watch on your wrist, or your cellular phone, or your personal digital assistant? Someday in the near future all these features and many more will appear in special glasses. Certain people will own these glasses. When a person with these glasses goes to a party or a business meeting the glasses will tell them the names and other important information about other people they run into. The people who own these glasses will have a distinct advantage over those that do not. Eventually, the majority of people will own these glasses. This process will continue. Machines will take over the human race because the machines will be us.” [Minsky, 1996]

The Modulator view of robotics may in fact be closer to that of the “wearable” view than that of traditional robotics. The wearable definition of computation explicitly includes a human participant as a part of its construction. In a wearable computer, the human being has become a robot, synthesized with computationally augmented senses. [Starner, 1997]

In contrast to Kurt Schmidt’s *Man+Machine* or “man meets machine,” our version of performance suggests “man as a part of machine.” Our version of the performing machine does not assume human form, but rather borrows the modernist tradition of “form from material.” [Gropius, 1961] Though this is perhaps where the Modulator robot differs most dramatically from Starner’s wearable machine.

In Starner’s system, computation is used to augment, emphasize, or heighten the awareness of a participant’s senses. In contrast, the Modulator vision of computation does not any explicit addition of computational firepower to human sensory perception.

Rather, in the Modulator language, the computation is still quite separate from the human participant, though they are both part of a larger system. There is no notion of an abstracted “computer” as an intelligent interacting entity similar to a human. Instead, we have only computational inputs and outputs. When constructing system models, the author must insert the “human interface” in or as a flow of communication.

This linguistic structure highlights a key aspect of the Aesthetics & Computation movement: for now the computer cannot create an interesting design on its own. It must be taught by a computational designer how to react to given interactions to present a very defined result. Computational components must be controlled. The author of a digital performance system must understand possible ranges of interface feedback and design their components accordingly.

Chapter 6: Conclusion

To begin this thesis, we discussed the origins of system model diagrams. From the first cave paintings to digital-era logic diagrams, humans have long expressed process through diagrams. With the dawning of the modern movement, certain artists began to develop diagrams for explaining creative processes. These artists seamlessly integrated visual languages into their work.

We next examined historical trends that led to the current state of digital performance design. Long before the introduction of the digital computer, artists built intercommunicating, performing machines. They conceived reconfigurable theaters and interactive spaces which blurred the lines between performers and audience, between the inside and outside of the area of performance. The first wave of artistic creation with computers was marked by collaborations between skilled engineers and well-known artists. With the introduction of the personal computer and a large number of commercially-available tools for design, artists appeared to be free from the burden of engineering the computer. Creation on computers became fractured into two distinct groups: the engineers, those who make the tools for design, and the designers, those who use the tools for design. As a result of this division, few avenues exist for expression within the computational medium outside the bounds dictated by commercial applications. Commercial applications for creation remain trapped in a narrow realm of interface actions guided mainly by metaphors from preexisting media. The only means to remedy this situation is for the artist to create custom applications.

We then called for a new form of creation in the computational medium, the “digital performance,” a live, interactive event emphasizing the manipulation of time. In support of this call, we introduce several novel systems for digital performance created by the author.

To help explain this new direction, we presented a language and research initiative called *Modulator* for creating system model diagrams for digital performance. In the *Modulator* language we establish a unified visual language of interface design based around the concept of interaction as sensory performance. The language consists of seven building blocks: visual input, visual output, audio input, audio output, touch input, touch output, and the human.

In the *Modulator* research system, one creates and examines three-dimensional, living system diagrams with our new language. We constructed prototypical interface system models and demonstrated methods for analyzing feedback and communication in complex system models. Finally, we used the *Modulator* system to categorize digital performance spaces, both by the author and by others, and as a framework to appraise and explain the creation of computational performance spaces.

New Hope for Digital Design

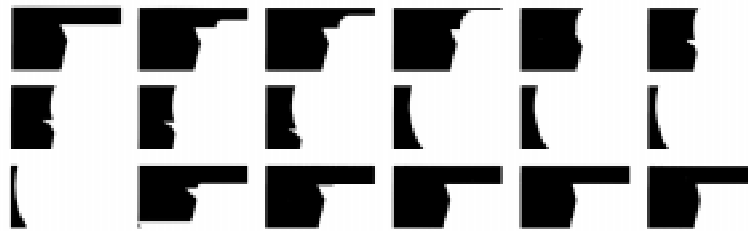


Figure 6.1: Graphic abstraction created by the author exploiting computational “bug”

The study of design with computation is not a science. To paraphrase Daniel Hillis, progress in science comes by taking things apart, progress in computation comes by putting things together. Scientific systems exist, we develop postulates as to why they exist, how they function and we test those postulates. Digital design and the digital aesthetic do not yet exist. [Hillis, 1985]

Without the aesthetic, design is either the humdrum repetition of familiar cliches or a wild scramble for novelty. Without the aesthetic, the computer is but a mindless speed machine, producing effects without substance, form without relevant content, or content without meaningful form.

Expressions such as *computer art* and *creative tool*, either because they are misunderstood or because they are misapplied, are expressions of deception. [Rand, 1996]

There can be little doubt that a vast gulf exists between the current state of expression with computer interfaces or “computer art” and the fluid, beautiful expressions seen in the artistic examples by Bauhaus masters. Yet one must believe it possible to make art in the computational medium.

...the computer may be a very helpful instrument in the hands of an artist. In the words of Nam June Paik ‘art is what artists do’. If we accept his statement, we must also accept that all computer output produced by artists is to be considered as works of art. In that case the problem remains, however, who is considered to be an artist? [Johannes van der Wolk from van Berkum, 1986]

At this point in the development of the computational medium and its applications, one finds it difficult to define computational art because it is difficult to determine who is actually creating the art.

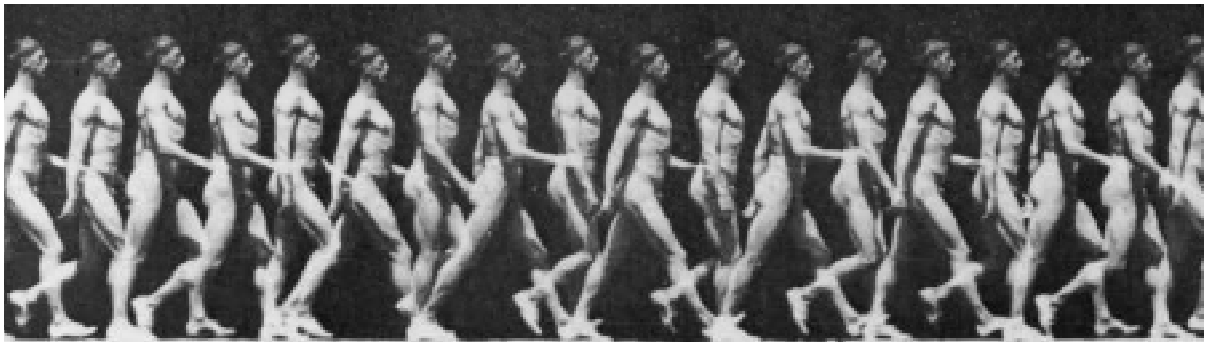


Figure 6.2: Etienne-Jules Marey; Photographic sequence of a walking man, 1885

The understanding of computational work by critics and by the public requires a degree of artistic accountability. Acceptance of film as a legitimate artform occurred in parallel to the growth of the *Auteur* theory. This theory poses that a piece of cinema, though sculpted by tens if not hundreds of hands, is the result of the purposeful vision of a single director.

Auteurism shares certain assumptions: notably, that a film, though produced collectively, is most likely to be valuable when it is essentially the product of its director ('meaningful coherence is more likely when the director dominates the proceedings'); that in the presence of a director who is genuinely an artist (an *auteur*) a film is more likely to be the expression of his individual personality; and that this personality can be traced in a thematic and/or stylistic consistency over all (or almost all) the director's films... [Critics] seemed to assume that if films were to be considered an art, as it quite generally was, then what they were urging inevitably followed: film is an art, and art is the expression of the emotions, experience, and 'world-view' of an individual artist. [Caughie, 1981]

In the same way, computational work will only be accepted if the notion of authorship can be established. Due to the large number of people in collaboration on the majority of computational works, it is often difficult to convey a single vision or design process at work.

Even the single designer toiling alone with her computer may not be sure who makes her work. Is it she herself or the team of engineers who created the software filters she employs?

Truly great works of art and design will be created with computation. We simply have not seen them yet. This is one of the reasons "computer art" seems to be a deception. We have only just begun. We can merely highlight promising avenues for exploration and point the way towards the performing machine.

Chapter 7: Bibliography

- Aesthetics & Computation Group, Kinetic Design Group Digital Communication Design Forum. Tokyo, Japan: International Media Research Foundation, 1997
- Braitenburg, Valentino Vehicles: Experiments in Synthetic Psychology. Cambridge: MIT Press, 1984
- Brave, Scott/ Dahley, Andrew InTouch: A Medium for Haptic Interpersonal Communication. Human Factors in Computing Systems, CHI 1997 Extended Abstracts, p 363
- Buxton, William Less is More (More or Less): Some Thoughts on Information Appliances, Colloquium speech at MIT Media Laboratory, Cambridge, MA, February, 25, 1998
- Cage, John Interview with John Cage. TDR (T-32), p 50, 1965
- Campbell, Kim Take Your Library Along - Even to the Beach. The Christian Science Monitor, April 16, p 14, 1996
- Caughie, Jonn (Ed.) Theories of Authorship. London: Routledge, 1981
- Dahley, Andrew/ Wisneski, Craig Water Lamp and Pinwheels: Ambient Projection of Digital Information into Architectural Space. Abstracts of CHI'98, ACM: 269-270, 1998
- Davis, Douglas Art and the Future. New York: Praeger, 1973
- Eames, Charles and Ray A Computer Perspective, Background to the Computer Age. Cambridge, MA: Harvard University Press, 1973
- Fitz, P.M. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. Journal of Experimental Psychology, 47, p 381-391, 1954
- Foley, J./ van Dam, A. Computer Graphics. Reading, MA: Addison-Wesley, 1990
- Gerstner, Karl Designing Programmes. Teufen AR, Switzerland: Arthur Niggli Ltd., 1964
- Gropius, Walter The Theater of the Bauhaus. Middletown, Connecticut: Wesleyan University Press, 1961
- Grenby, Matthew Gradus: Revealing the Shape of the English Language. Visual Proceedings, Computer Graphics, 1997, p 137
- Haeberli, Paul E. ConMan: A Visual Programming Language for Interactive Graphics. Computer Graphics, Volume 22, Number 4, August 1988
- Hillis, Daniel W. The Connection Machine. Cambridge, MA: MIT Press, 1985
- Barfield, W./ Furness, T. (Ed.) Virtual Environments and Advanced Interface Design. Oxford, 1995
- Jones, Steven G. (Ed.) Cybersociety: Computer Mediated Communication and Community. Mon-sand Oaks, CA: Sage, 1995
- Kepes, Gyorgy (Ed.) Sign, Symbol, Image. New York: George Braziller, 1966
- Klee, Paul Notebooks, Vol 2: The nature of nature. London: Lund Humphries, 1973
- Klee, Paul Notebooks, Vol 1: The thinking eye. London: Lund Humphries, 1961
- Kostelanetz, Richard Moholy-Nagy: An Anthology. New York: Da Capo Press, 1970
- Kram, Reed/ Maeda, John Dynamic3: Interactive Physics and Physicality in Three Dimensions. Visual Proceedings, Computer Graphics, 1997, p 140

- Landsdale, Mark/ Ormerod, Thomas Understanding Interfaces: A Handbook of Human-Computer Dialogue. London: Academic Press, 1994
- Langolf, G.D./ Chaffin, D.B./ Foulke, J.A. An Investigation of Fitts' Law Using a Wide Range of Movement Amplitudes. *Journal of Motor Behavior*, 8, p. 113-128, 1976
- Laurel, Brenda (Ed.) *The Art of Human-Computer Interface Design*. Reading, MA: Addison-Wesley, 1990
- Laurel, Brenda *Computers as Theatre*. Reading, MA.: Addison-Wesley, 1993
- Le Grice, Malcolm *Abstract film and beyond*. Cambridge, MA: The MIT Press, 1977
- Maeda, John/ McGee, K. *Dynamic Form*. Tokyo, Japan: International Media Research Foundation, 1993
- Maeda, John *Flying Letters*. Tokyo: Digitalogue, 1996
- Maeda, John *Reactive Books (Catalog for Digitalogue)*. Tokyo: Digitalogue, 1997
- Maeda, John *The Reactive Square*. Tokyo: Digitalogue, 1996
- Martin, Keith *Automatic Transcription of Simple Polyphonic Music: Robust Front End Processing*. M.I.T. Media Lab Perceptual Computing Technical Report #399, 1996
- McNamara, Brooks/ Rojo, Jerry/ Schechner, Richard *Theatres, Spaces, Environments: Eighteen Projects*. New York: Drama Book Specialists, 1975
- Minsky, Marvin "Perspectives" Talk at MIT Media Laboratory. October, 28, 1997
- Moholy-Nagy, Laszlo *Vision in Motion*. Chicago: Theobald, Paul, 1947
- Moser, Mary Anne *Immersed In Technology: Art and Virtual Environments*. Cambridge, MA: MIT Press, 1996
- Myers, Brad A. (Ed.) *Languages for Developing User Interfaces*. Boston, MA: Jones and Bartlett, 1992
- Naimark, Michael *Experimental Media Space Background Study for Massachusetts Institute of Technology, Arts and Media Technology Facility*, 1981
- Rand, Paul *From Lasceaux to Brooklyn*. New Haven, CT: Yale University Press, 1996
- Rosen, Kenneth H., *Discrete Mathematics and Its Applications*. New York: McGraw-Hill, 1995
- Roth, Charles *Fundamentals of Logic Design*. St. Paul: West Publishing Company, 1975
- SensAble Devices Inc. *The PHANTOM*. Literature from SensAble Devices Inc., Vanceburg, KY, 1993
- Sims, Karl *Particle Animation and Rendering Using Data Parallel Computation*. *Computer Graphics*, Volume 24, Number 4, August 1990
- Smith, Joshua R. *Field Mice: Extracting Hand Geometry from Electric Field Measurements*. *IBM Systems Journal*, Volume 35, No. 3 & 4, 1997
- Starner, T./ Mann, S./ Rhodes, B./ Levine, J./ Healey, J./ Kirsch, D./ Picard, R./ Pentland, A. *Augmented Reality Through Wearable Computing*. *Presence* 6(4), 1997
- Sutherland, Ivan E. *Sketchpad: A Man-Machine Graphical Communication System*. *Proceedings of the Spring Joint Computer Conference*, 1963, p 329-346
- Thomas, Frank/ Johnston, Ollie *The Illusion of Life, Disney Animation*. New York: Hyperion, 1981

- Thompson, d'Arby On Growth and Form. Cambridge, England: Cambridge University Press, 1961
- Thorburn, David "The Aesthetics of Transition" speech at MIT Media Laboratory, 12 February, 1998
- Tufte, Edward The Visual Display of Quantitative Information. Cheshire, Connecticut: Graphics Press, 1983
- Tufte, Edward Envisioning Information. Cheshire, Connecticut: Graphics Press, 1990
- van Berkum, Ans/ Blekkenhorst, Tom science * art. Utrecht, Holland: Fentener van Vlissingen Fund, 1986
- Viola, Bill Reasons for Knocking at an Empty House. Cambridge, MA: MIT Press, 1995
- Wiener, Norbert Cybernetics. Cambridge: MIT Press, 1948
- Wiener, Norbert Extrapolation, Interpolation, and Smoothing. Cambridge, MA: The MIT Press, 1948
- White, Tom/ Small, David A Poetic Interactive Garden. Human Factors in Computing Systems, CHI 1998 Summary, p 335
- Witkin, A./ Fleischer, K. Energy Constraints on Parameterized Models. Computer Graphics, 21: 225-232, 1987
- Witkin, Andrew Physical Based Modelling: Particle System Dynamics. Pittsburgh, PA: Carnegie Mellon U., 1997
- Witkin, A./ Kass, M. Spacetime Constraints. Computer Graphics, 22:159-168, 1988
- Youngblood, Gene Expanded Cinema. New York: E. P. Dutton & Co., 1970
- Zeleznik, R./ Herndon, K. SKETCH: An Interface for Sketching 3D Scenes. SIGGRAPH'96 Proceedings, 163-170, 1996

Commercial Software

AutoStudio, Alias|Wavefront, Toronto, Canada

Cakewalk Pro Audio, Cakewalk, Cambridge, MA

Houdini, Side Effects Software, Toronto, Canada

MAX, Opcode Systems, Palo Alto, CA

Mixman Studio, Mixman Software, San Francisco, CA

Rebirth TB-303, Propellerhead Software, Sweden

SimCity 3000, Maxis Software, Walnut Creek, CA

SoundEdit16, Macromedia Inc., San Francisco, CA