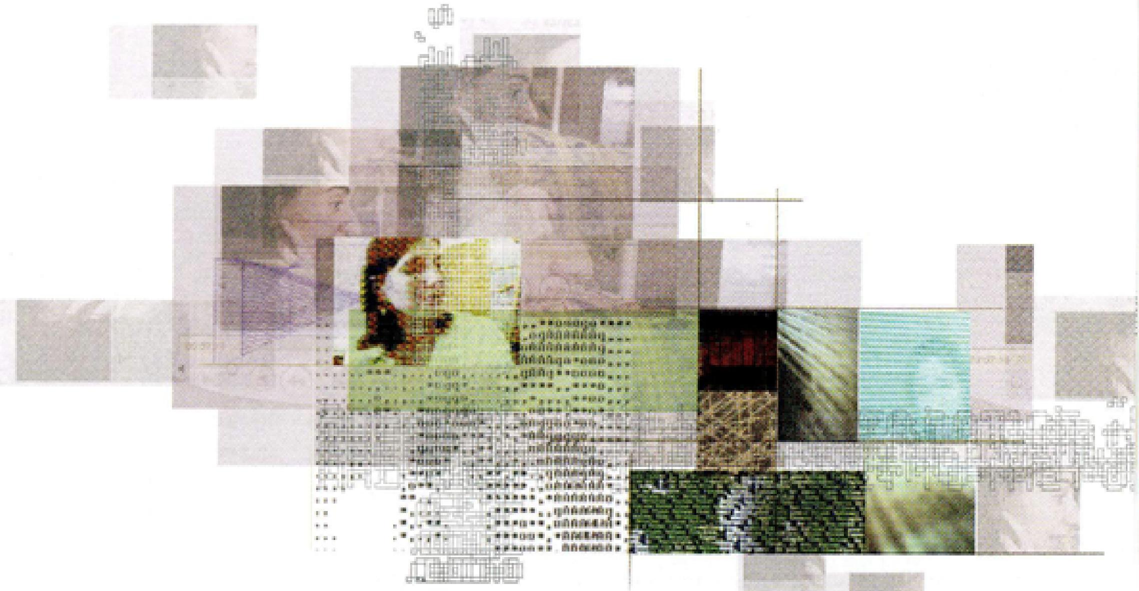


TRUE NAMES

THE ART OF ENGINEERING | THE ENGINEERING OF ART

Was the first cave painter an artist or engineer?" — Steven Johnson / Interface Culture

Until a few years ago I was naïve. I thought of computer code as a tool exclusive to engineering. My own experience with computers was limited to writing school papers and playing ***Oregon Trail.*** As far as I knew, engineers merely used code to write software and produce appliances. End of story. The only creative potential I could see for software was once-removed from the actual code: I had a dim notion that a particular combination of software packages might one day allow me to create anything my imagination could dream up. My undergraduate experience in art school reinforced these assumptions. I learned printmaking and created hand-drawn animations; both traditional media that are highly tactile.

Once I was out of school, though, my creative work slowly evolved away from traditional studio practice and toward computer technology. Software applications offered more convenience, but I frequently felt limited by the standard options available in an off-the-shelf tool. What I really wanted was to create my own tools, and this desire drew me progressively toward the intersection of engineering and art making. I began to suspect that code held more potential than I had initially glimpsed, and that it might have applications beyond the standard clichés. I started to wonder if there might not be a rhetorical quality to code, and that its manipulation was a connecting thread between engineering and art.

Then something serendipitous occurred. I attended a digital media lecture series at the ***Rhode Island School of Design***, where I came into contact with the work of two people whose perspectives on computer code would re-write my assumptions.

Camille Utterback presented work from her ***Liquid Time Series;*** witnessing her fusion of art and programming was a turning point for me. Utterback identifies herself as an artist but considers her medium to be computer code. Needless to say, her work does not always sit comfortably within the traditions of the contemporary art world. Any cues that might refer a viewer back to a specific historical precedent—the painted portrait, for instance—are noticeably absent. Instead, *Utterback's* work manufactures a new cultural space that combines the display aspects of the traditional museum with utterly new performative qualities. Her work engages the viewer in a dynamic, very personal experience, one in which the work responds in an almost sentient manner to the viewer's actions.

Liquid Time Series is an interactive video installation in which the video playback is controlled by the viewer's movement in the physical space. A large still frame of a scene from New York or Tokyo is projected on a wall, but when a viewer walks toward the image it begins to fragment into discreet vertical columns that ripple independently. Imagine what the rupture might look like

if you stuck your hand into a perfectly smooth waterfall. What initially appears as a single image is actually one moment in a timeline, with each individual vertical column of the image capable of running independently. If a viewer walks toward the center of the image, that section plays forward in time—the viewer can literally penetrate one point in the scene and walk deeper into time. If two people interact with the image, they can independently control the moment in time displayed in the section of image they each are effecting. Backing up rewinds the scene. Moving left or right, or leaving the space, causes the piece to reset back to the original moment of origin. This device, which Utterback describes as "video cubism," demonstrates both her command of programming and her ability to see the lyrical and poetic potential lurking just below the surface of algorithm and apparatus—the rise of this new art form is a result of her fresh observations and interpretations of emergent technology.

Liquid Time Series challenges basic protocols that museums have established between object and audience, particularly the taboo against "handling" the artwork. In Utterback's piece, as long as the viewer remains a passive observer the image remains static. The narrative embedded in the image can only be activated when the observer becomes an active and willing participant in the manipulation of the work. The more playful the participant, the more enlivened the experience. This demand for a new relationship between the work and viewer is a recurrent element in nearly all of Utterback's work.

The other seminal project that I learned about through the RISD lecture series was ***ChitChat Club,*** created by **Karrie Karahalios**, who was at that time a graduate student at the ***MIT Media Lab***.

ChitChat Club was a café where real-world patrons mingled with people accessing the club through the Internet. It was furnished with typical cafe tables and chairs, but also included various "avatar chairs." These chairs were designed in the form of an abstracted, seated human figure, with a monitor at head height projecting an image of the remote participant, who had the option of communicating with either audio or text. Text was displayed on a screen at chest height on the avatar chair, and a small camera on the head of the chair streamed live video back to the remote participant's computer screen.

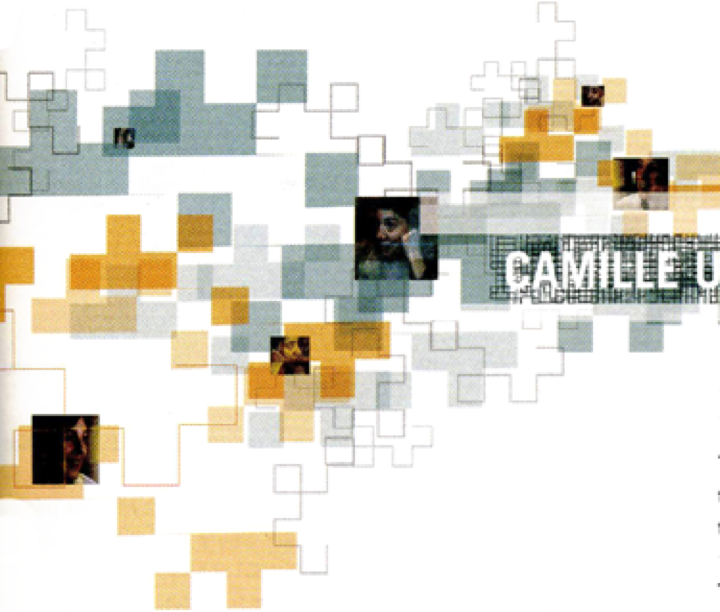
This project grew out of Karahalios's interest in computer-mediated social spaces, and it sought to use an everyday social environment as a metaphor for constructing a new technological space in which people could interact socially, even though they were dispersed geographically.

ChitChat Club focused on developing a technology that people would actually want to use. Too often, developers of new technology simply fail to consider the user's experience. But Karahalios has a distinct interest in how audiences interact with technology, and her work is characterized by refined interface design. ***ChitChat Club*** was a success not just because it was attractive and functional, but also because it allowed space for the kind of ephemeral qualities that characterize our daily social interactions.

As with Utterback's work, ***ChitChat Club*** requires an active participant rather than a passive observer. Without the patrons of the club, there would be no event. They are, in essence, the content of the work. Everything is premised on an erasure of the traditional division between author and audience.

Both Utterback and Karahalios are programmers. Both write their own software for each individual project. Yet Utterback identifies herself as a San Francisco-based artist, while Karahalios is an engineering professor at a large research university. Although they may name themselves differently, it is clear that both are situated at the intersection of art and engineering. What is the significance of this convergence of disciplines, if it is a convergence at all, and how is it properly defined? Its names are multiple and various: Digital Media, Interactive Media, Electronic Media, Narrative Media, Art &Technology, Media & Arts. Does it makes sense to invent a new discipline from two established disciplines, or are we immediately embedding unnecessary limitations?

I don't know the answer to these questions, and have yet to meet anyone who does. What I do know is that my original understanding of art and engineering was assembled through a limited exposure to works that were digital adaptations of more traditional practices, and that my exposure to the work of Utterback and Karahalios dismantled those assumptions. Their work demonstrates code as a raw material in a creative process, and this altered my appreciation for the growing influence of digital culture. — ***Matthew Yapchaian***



*In the spring of 2005, Utterback and Karahalios sat down together to talk about their work. An excerpt from that conversation appears below. When we begin, Utterback is talking about her piece **Liquid Time Series**.*

CAMILLE UTTERBACK & KARRIE KARAHALIOS

Camille Utterback: And then people start to notice, *"Wait, there's something else going on here."*

They think, "Where are these letters coming from? Oh, hey, I'm stopping them." And then people start to explore all of that: "How do I stop the letters? Can I push them around?" And then maybe they get to the point of, "Oh, they're not just random. What? Is this actually saying something?" They might start trying to decode what the actual words are in the poem, or building their own words.

So those are three really distinct elements: the recognition that the piece is responding to you; understanding how things in the physical environment create a response in the virtual space; and then the conceptual or literary question of what's being said, what's going on. I always try to talk to students about that: how do you address the full spectrum of possible interactions, even in one piece?

Karrie Karahalios: There was an interactive toy, actually not an interactive toy, it was a play-back toy called Teddy Ruxpin a while back.

I don't know if you remember it. You put a cassette in him and he tells children's stories, and the arms would move up and down. The behavior was so programmed, everyone knew what would happen, and there was nothing unexpected. It didn't learn or adapt.

I guess being in the Computer Science department, and having spent a lot of my undergraduate career dealing with machine learning, I see the next step as making interfaces that learn from previous experience.

I could just let it loose, and almost not know what's going to happen later on. I don't want the interface to attack the people using it, but what happens if you have a few simple rules and let them adapt over time? For me it's interesting to see what the piece will look like a year from now, two years from now; after people have used it, and that has affected how it grows or how it transforms. That's where I want to go.

The first pieces that I've tried to do this with have ended up being more formulaic; more like finite state machines than learning machines. Right now I'm playing with linguistics and Natural Language processing, trying systems whose parameters respond to what people say or what people do in the space.

The challenge for me is engineering the design changes that happen based on in people's language and behavior. For all I know, over time the end result might just be this big gray screen.

CU: What will people be seeing in the space, what is their impetus to talk to this interface?

KK: It will be something where you actually start off with the idea that the representations will change based on what people use. So, right now, the more you talk and the more you move, you see more graphical representations that are cartoon-like.

What if, based on the user's actions, it slowly adapted to what they said or did? But if the challenge is: How do you make that happen? Basically, you are a god in a system like that. I guess one example is playing with "L" systems—an "L" system is basically a fractal, and a lot of people use them to design virtual plants.

One of your pieces reminded me a bit of "L" systems; the one piece that has these beautiful green, sort of organic stem forms. At first, I thought, "how is she growing those?" An "L" system based on environmental change could grow in flexible ways. It could start off with the same rule base—a mathematical algorithm—but those rules could change based on surroundings.

Almost like Nature over Nurture. I want to know how you got those green stems to grow.

CU: I have very simple swirling lines that have rules about rotation and movement. What's interesting is that viewers can walk in front of the image and change it—the lines change color, although their movement is unaffected. So I have a system of all these swirling lines, and when someone walks into the frame the outline of their body is captured.

There's a really interesting intersection of the organic contour of someone's body and the movement of these lines. I'm interested in allowing that kind of variability to exist.

KK: Ahhhhhhh. I thought that was beautiful, It reminded me of some of Carl Sims's work, where there are two organisms with essentially two separate DNA strands, and when they make contact they make a new being which is a mix of their DNA.

So a new creature with new behavior and physical characteristics is let loose. Over time people can choose which ones to breed. It scared some people—the idea of cloning and selective reproduction.

I think your pieces are far more fluid than any of the pieces done by people working with the DNA metaphor and computational structure, though.

CU: I want to go back to something in your interface work—the idea about it constantly evolving. I think people are often more creative, and their projects are more interesting, if they have limitations.

I always use the example of a party: if you had a word processor on your refrigerator, it's unlikely that any of your guests would write anything interesting; maybe one friend who was a poet and wanted to perform at your party.

But the "magnetic poetry" kits people put on their refrigerators has enough constraint that the results are really interesting. I think audiences sometimes get confused about the constraints in my pieces. They might say, "Can't you let people change the color? Can you let them upload their own shapes?" My response is that I'm not making a tool, I'm not making a word processor. It's not completely open-ended, and that's a really different challenge. What I'm trying to do is a totally different project, which is more like "magnetic poetry" and hopefully a little more interesting. The constraint is a challenge, a dialogue between the audience and the work.

In one piece, I couldn't decide how to make the behaviors change. Then I thought, "Why don't I let the audience choose how they want to be represented?" And it was fascinating to me the different choices people made in how to be represented. One person would choose a photo-realistic representation, while someone else would choose something more abstract. I ended up with an abstraction lever. In a sense the abstraction was the constraint. I have this love-hate relationship with constraint.

I want to build installations that give people of full spectrum of interactions, but in the end I have to limit what I provide them. So I struggle with constraint, because if you make the interface too transparent people figure it out in five minutes, and then they walk away.

CU: That's not interesting.

KK: But if you can keep them in the flow, keep them entertained, then it becomes like...

CU: Exactly.

KK: You have to find the right balance.

CU: The structure has to be comprehensible enough that it's not like one of those awful games where the rules are way too long and you never want to play again.

KK: Or that you need to read the manual. You shouldn't need a manual.