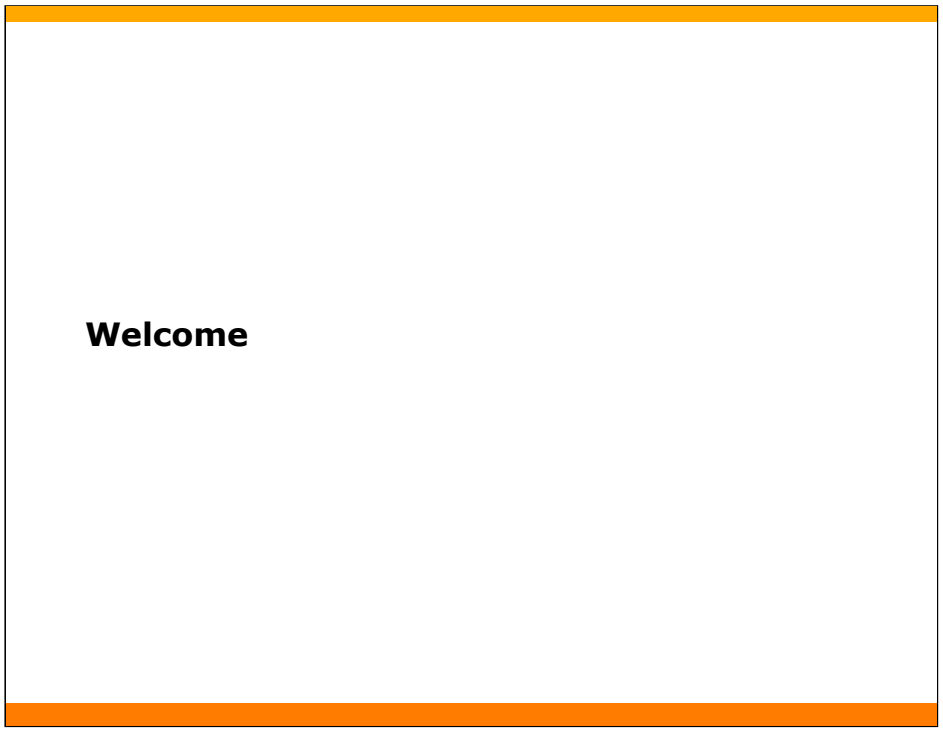
Sketching in Hardware

Matt Cottam
Mike Kuniavsky

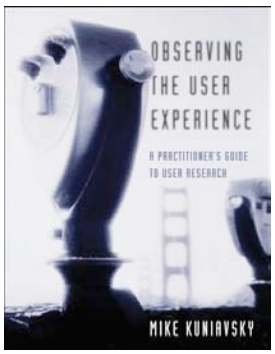


Intro: I'm Mike Kuniavsky and this is Matt Cottam. We're going to talk how to bridge where we are today to Bruce Sterling's Internet of Things, or whatever we're going to call that future of smart objects that's coming sooner than we expect.

I'm going to provide some background and pose a question. Matt will present follow up with a possible answer.

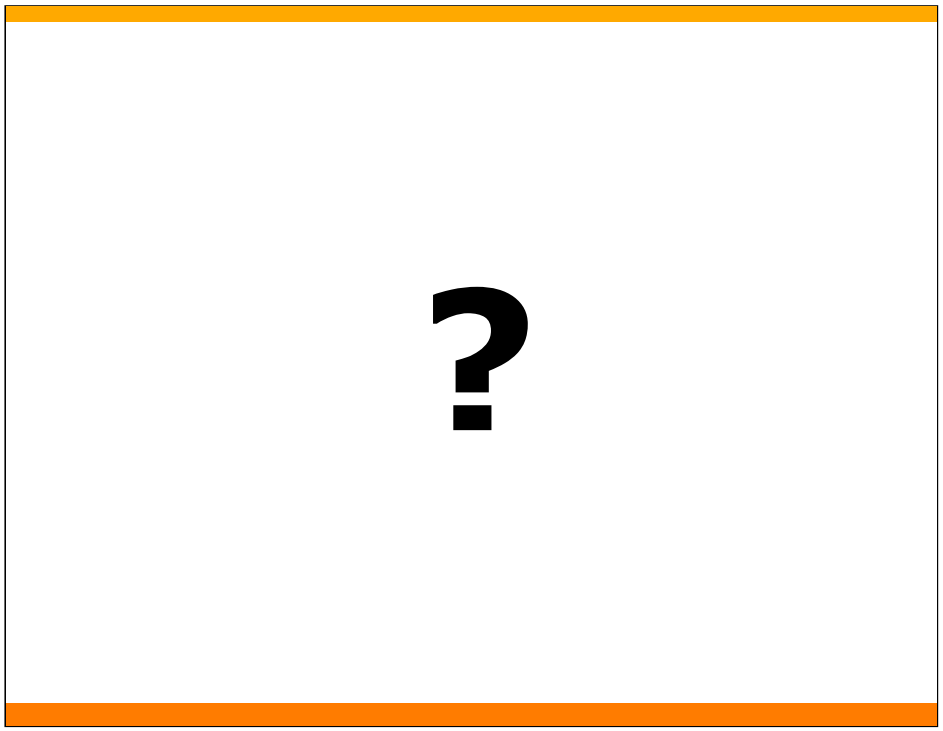
About Mike, Part 1

adaptive path



Let me give you the short version of my official bio: I'm responsible for some of the oldest graphics on the net, which I made for an early ecommerce site that's still running 12 years after HotHotHot, an early ecommerce website, launched in 1994. After that, I then went to work at HotWired, where I was the UI designer for HotBot and started Wired Digital's user experience research lab. Then I wrote a big book on techniques for understanding how people experience designed products, and finally I was a founding partner in Adaptive Path. I left AP about two years ago to focus on my other interests.

And here are those other interests. I am interested in the expressive physical aspects of technology. Here are some projects I've done. This (Stock Puppets) is a robotics project I did with Jim Mason some years ago, this (Three Dreams) is part of an installation I did with Elizabeth Goodman, this (Bass Ghost) is a generative subsonic bass project, this is a Roomba hacking project with Tod Kurt that recently got blogged by Phil Torrone of Make, and this (C4F3) is a cafe of augmented objects that I'm curating for the 2006 International Symposium on Electronic Art/ZeroOne Festival.



So where am I going with this? I'm an experience designer and the dominant theme in my thoughts about emerging technology is how it affects people. These experiments that I just showed got me involved with the intersection of material objects, information technology, and the effect the combination of the two have on people, so I've been thinking pretty hard about the design of everyday objects that have information technology embedded.

Things like this. Bruce Sterling and Adam Greenfield are of course right that these things are only the beginning, that there's a much different future that this is leading to, but in order to get from where we are to what he's talking about, we have to go through these objects. And these need to be designed.

I am convinced that NOW the time for actively innovating in terms of design for this, much more than the technology.

I feel that the action in emerging technology has shifted from being a problem of *defining and refining* the basic building blocks to being a design problem, where the action is in term of *combining the basic blocks* into interesting things that solve real problems for people.

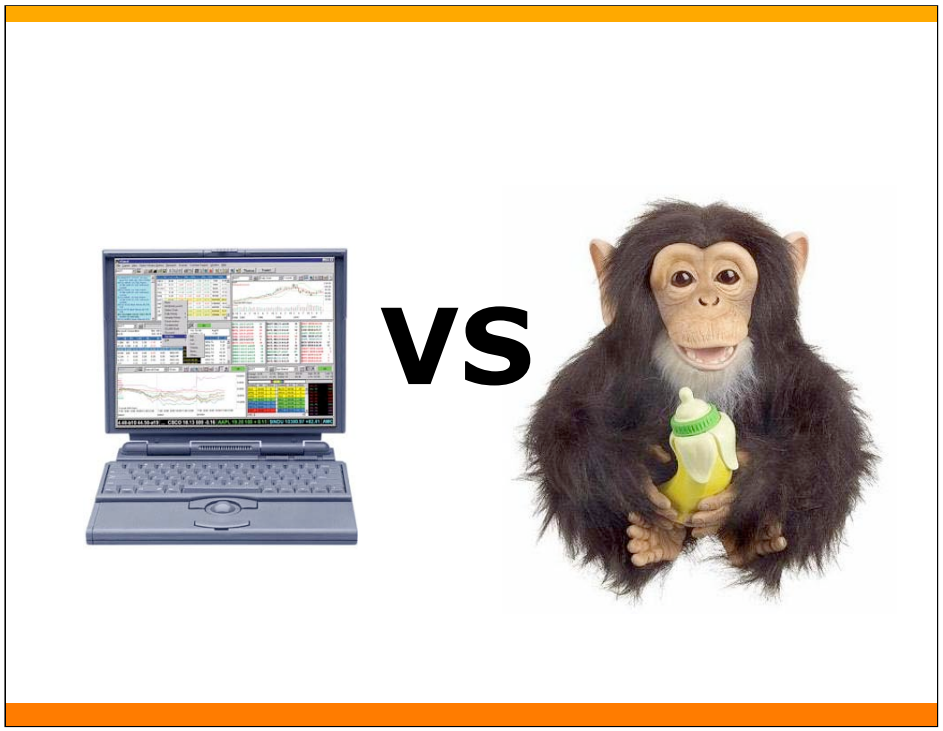
Sources: Toyota Prius w/Bluetooth, Robosapien, Adidas 1, iPod Shuffle

Casemods, not ubicomp

(Sources: Intel, Electrolux)

And we're not talking about this. This is a not a casemod exercise, it is a fundamentally different problem than making general-purpose computers. Ubicomp is more than just duct taping a tablet PC to something.

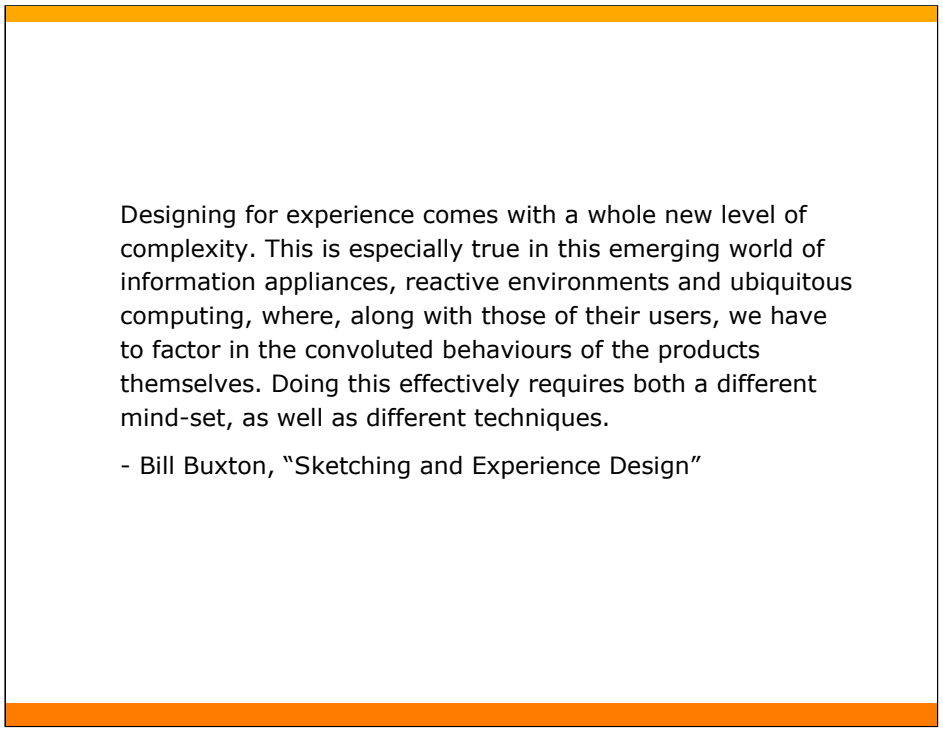
Source: Intel, Electrolux



What we're talking about is designing wholly new kinds of information processing devices. This is Cuddle Chimp, which is essentially a robot that simulates a human baby in the form of a small ape, in order to avoid the uncanny valley. Designing user experiences for ubiquitous computing is largely terra incognita. We just started figuring out how to make Web pages not totally suck, and this is a whole new game.

Why? Because atoms are much harder to push around than bits. The basic blocks are not just software, general-purpose computer hardware and screen real estate, but include the stuff that used to be the realm of robotics, mechanical engineering, material science and industrial design: sensors, actuators, pulleys, levers, plastics, metals, composites, etc. And what is becoming apparent is that the tools for developing such hybrid physical/informational devices are nowhere near as sophisticated as the tools for developing software or for working with wood. In terms of prototyping these appliances, we're where software was in the mid 1960s, debugging with printline statements and sticking oscilloscopes on pins.

Source: Cuddle Chimp by Hasbro

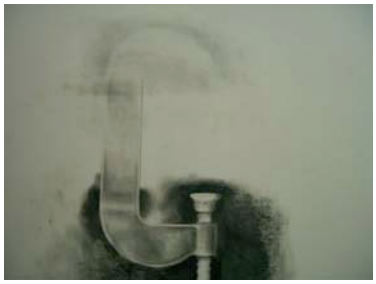


Bill Buxton, describes the problem well. And the key part of this quotation for me is the end: we need both new techniques and a new mind-set. I would add that those two ideas are embedded in the tools we use, so we need new tools.

What makes a good ubiquitous computing design tool? I don't know, but I decided to go back to the origin of all design tools to think about what qualities it could have.

Sketching

"Sketching is a process, a kind of inquiry, rather than simply a matter of externalization."

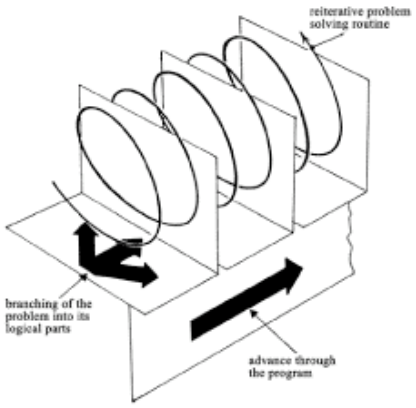


(Sources: Fallman, D., "Design-oriented Human—Computer Interaction", CHI2003, Jim Dine, *Untitled* from *Ten Winter Tools*, 1973)

I went back to sketching.

Sketching is the archetypical process that defines all iterative design methods, whether you're working with paper, or stone or software. All development consists of the three fundamental steps of Analysis, Synthesis and Evaluation, iterated. These are the heart of any design methodology, and the tighter the loop that connects them in a given medium, the faster ideas can be expressed in that medium.

Sketching is iteration



(Source: Archer, L. Bruce, 1969, "The structure of the design process")

Here is an image from 1969, describing an abstract design process that looks through the Analysis-Synthesis and Evaluation steps on its way to some arbitrarily defined "more designed" state. It looks a lot like the kinds of diagrams people are showing about agile software development, but there's a difference, and its those rectangles—those are essentially sketches.

So what are the parts of sketching as an abstract activity that makes it interesting?

Source: Archer, L. Bruce, 1969, "The structure of the design process"

I came up with three basic qualities, there are probably others, but I thought that these are particularly interesting.

1. The less time it takes to explore an idea, the more ideas can be explored. Explore possibilities as you stay focused on the broader goals and the relationships between the pieces you're working with, instead of having to recreate those ideas later interrupted by technical issues. Time spent recreating a mental state where you had an idea after dealing with grungy details is often a big hurdle to making the right thing, versus just the most expedient thing. This is the continuous partial attention problem.
2. You know a sketch is not the final product. There are a bunch of indicators that say it's not the real thing to you and to others, That way everyone stays focused on the core ideas, rather than being distracted by peripheral details. As the sketching process iterates, you can place the granularity of focus at the proper level.
3. Sketching shows you in one place the record of successful ideas, experiments and failures. You're constantly defining the envelope by being able to glance back to your dead ends and successes.

Medium	Speed	Provisional	History	Score
Drawing	5	5	5	125
Theater	5	5	3	75
Writing	5	4	3	60
Music	5	5	1	25
Software	2	3	4	24
Architecture	3	4	1	12
Interaction design	3	2	2	12
Information Architecture	4	2	1	8
Screen-level interfaces	2	3	1	6
Hardware	1	1	3	3

(Source: highly unscientific guesstimation)

So I decided to use these three qualities as the basis of a highly unscientific ranking exercise between various media and, what do you know, my gut-level expectations were confirmed. Sketching in hardware really blows. To make it clear: the problem is not that making hardware isn't as easy as sketching, it's that it's *nowhere near as easy*. The problem is not moving it to the top, but moving up in that list, at least a little.

For me this means that the focus of design for emerging technology changes from talking about capability, what something can do, to functionality, what kinds of needs it satisfies, what kinds of behaviors it enables, what kinds of human problems it solves. That's also what's interesting to me about Ajax. People complain that it's not a new technology. It's not, it's a refocusing of the existing building blocks on how to recombine them, how to use them, how to design with them.

To do this for ubicomp, I think that there need to be new design technologies whose purpose is to allow people to make something that's interesting, but who do not necessarily have the required technical knowledge to build something from scratch. How do we help people tie the world of information to the world of focused tools for everyday living?

And to talk about an answer to that, I give you Matt.