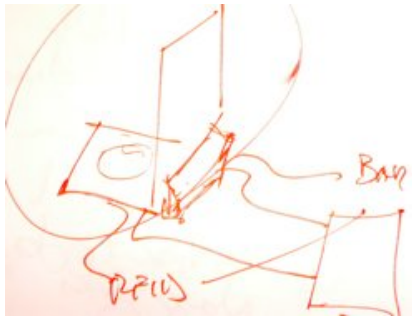
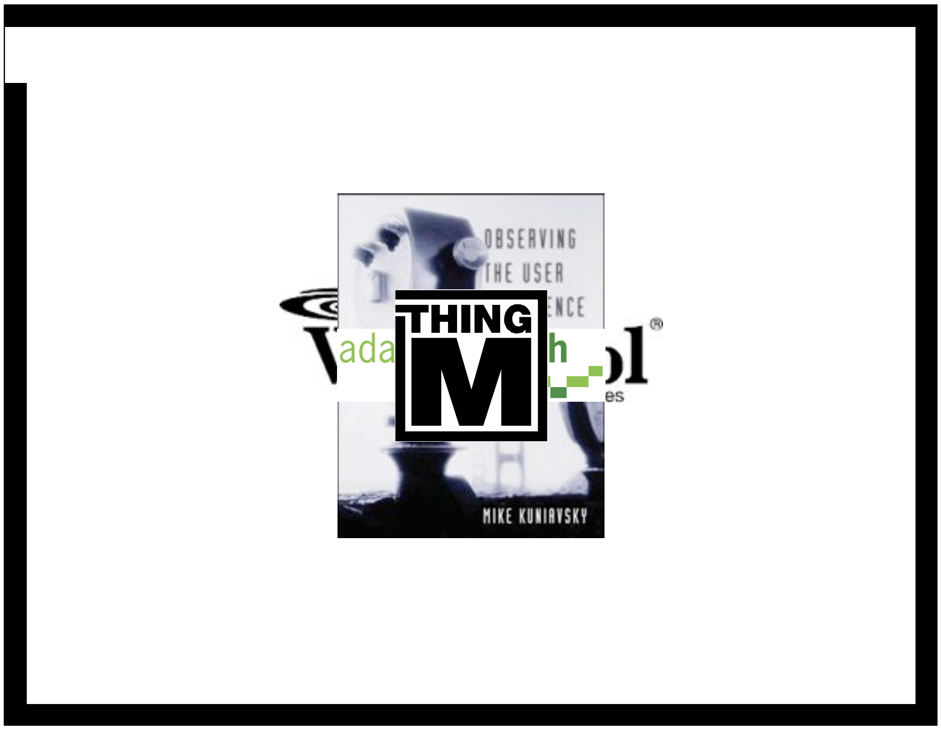
Sketching Smart Things



Mike Kuniavsky
CHIFOO
January 9, 2008

Hi. Thank you, Carrie and CHIFOO for inviting me.

I'm dividing tonight's talk roughly in half. In the first half, I'm going to talk about some of the high level ideas we're developing and working with, in the second I'm going to talk about some specific techniques we're using and give you some examples of how we've used them in the last year.



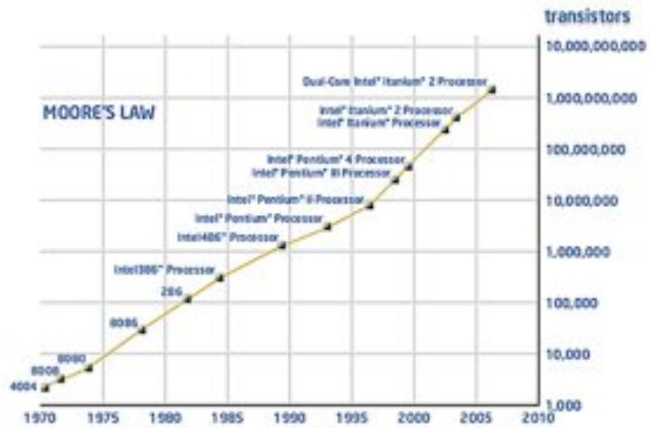
I’d like to start by telling you a bit about who I am.

I specialize in user experience design and user research. Over the years, I’ve worked with a wide variety of organizations to help them develop technology with people in mind.

[click]

As part of this practice, I wrote a book, a kind of cookbook covering a wide range of user research techniques. I co-founded a design company called Adaptive Path.

In 2006 I started a company called ThingM. We make ubiquitous computing consumer products.



Source: Intel

Let me start by talking a bit about ubiquitous computing and why I think it's really important. The term was coined by Mark Weiser in the early 1990s, but I don't think true ubiquitous computing was possible until just recently. Let's start with Moore's Law.

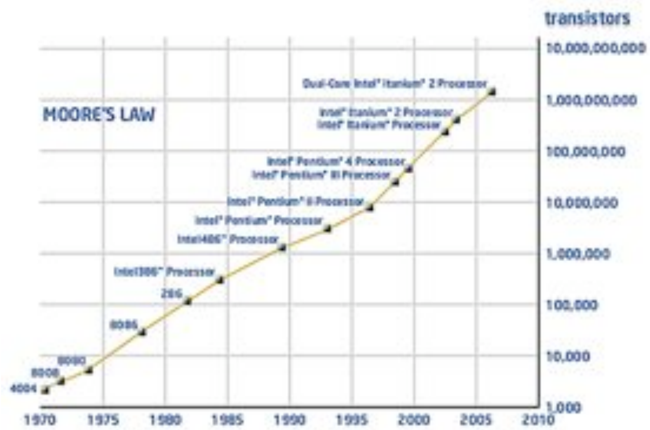
You've seen it a thousand times, but let's look at it again. People typically read this chart as a trend focusing on the number of transistors.

What's implicit in this trend, however, is that this is happening within the context of a marketplace.

This is not just the theoretically largest number of transistors that's possible to put on a on a CPU die. It's the number of transistors that can be sold at a specific price point.

Data source: http://www.islandnet.com/~kpolsson/micropro/

And if you look, the prices of new CPUs has stayed roughly the same. This is a graph of the price of major personal computer CPUs over the last 25 years, at the time of their introduction. Even with fluctuations because of market positioning and competition between Intel, Motorola and AMD, the price of a new chip has remained pretty steady, generally between $500 and $1000 at the time of introduction.



Source: Intel

Taken in light of processor prices, the other way that you can read this chart is that the PRICE of older processor technology decreases proportionally to the increase in transistor density. People tend to concentrate on the right side of the curve, but I'd like to draw your attention a little to the left, to what I call the Hidden Middle of Moore's Law. I think it starts right around the 486. These processors can do an immense amount and correspond roughly to the beginning of the modern, internet-connected computer.

1989: $900

33MHz, 20MIPS

2007: $0.53

20MHz, 20MIPS

Sure enough, you can see that the actual chip prices reflect that price drop.You can get roughly the same processing power today for 53 cents that cost $900 in 1989.

Unlike the early 90s when Weiser named and predicted the phenomenon, because of this price behavior ubicomp has only recently become a practical reality.

Source: Flickr, jurvetson

What does this mean for design? As interaction designers, we just figured out how to make Web pages not totally suck; as industrial designers we're just starting to realize that functionality and psychology are part of the job, too.

Tod Kurt and I founded ThingM because we believe that ubiquitous computing holds amazing promise for making the world a better, happier and more interesting place. We want to reinvent everyday objects in light of the capabilities of these new technologies. However, ubiquitous computing is today where the Web was in 1992, and the field is still mostly full of unknowns.

To wrap our brain around it, we've been bouncing around a number of abstract concept to explain what ubiquitous computing means to experience design.

Source: Flickr, Kate_A

1. INFORMATION PROCESSING IS A MATERIAL

First, we need to think of embedded information processing and networking as a material.

Let me explain. When a designer can include information processing in a product for very little cost, the calculation becomes not one of engineering complexity, that's relatively cheap, but one of competitive advantage. Including a CPU to produce behaviors becomes a line item in the competitive analysis of making an object, just like the calculation about what to make it out of. What you do with that CPU becomes part of the design of the product and needs to be designed with the same attention to the other parts as any of the materials being used. And just like a material, it creates some new capabilities, and imposes new constraints.

I think the toy industry is leading the way here. All kinds of toys now depend not just on their physical appearance, but on behavior created by information processing, for their competitive advantage.

Information as material is the heart of ubiquitous computing. Any tool can now use information processing to help it do its job better.

Let me make it clear, I’m not talking about casemods. That’s the practice of putting a general-purpose computers into a fancy new box. There’s no point. It’s two things that have been stuck together, not a new thing or an augmented old thing.

Let me give you a less silly example. 1998, V-Sync, a Japanese company, introduces their Internet Refrigerator. It has a computer with a touchscreen built in. 1999, Electrolux introduces their Screenfridge, it has a computer with a touchscreen built in. 2000, Whirlpool and Cisco introduce a fridge with a touchscreen. 2002, Whirlpool tries again with their Connected refrigerator. 2003, LG introduces their internet fridge. 2006, Electrolux tries again with a new Screenfridge.

Why am I showing you all of these refrigerators? Two reasons: one, to show you that although fridge computers have been for sale for nearly 10 years, the odds are that you’ve never seen one. Why? Because there’s no point. They don’t treat information processing as a key design material in the development of the fridge, they treat it like something that has to be forcefully grafted on.

> ‘Applianceness’ [is] the set of properties that guide the design process towards simple, helpful devices that exploit the potential of embedded information technology in everyday things.
>
> - Bill Sharpe,
> *Information Appliances: an introduction*,
> 2001

2. APPLIANCENESS

The second reason is that I’d like to introduce the idea of “applianceness.” This is a term coined by Bill Sharpe, a British interaction designer who has a consulting company called, appropriately, The Appliance Studio. They designed Steelcase’s RoomWizard scheduling device, a classic information appliance.

The core of the idea for me is that focus in functionality is more important than arbitrary flexibility. When computation is cheap, we no longer have to make general purpose computers.

Source: Whirlpool

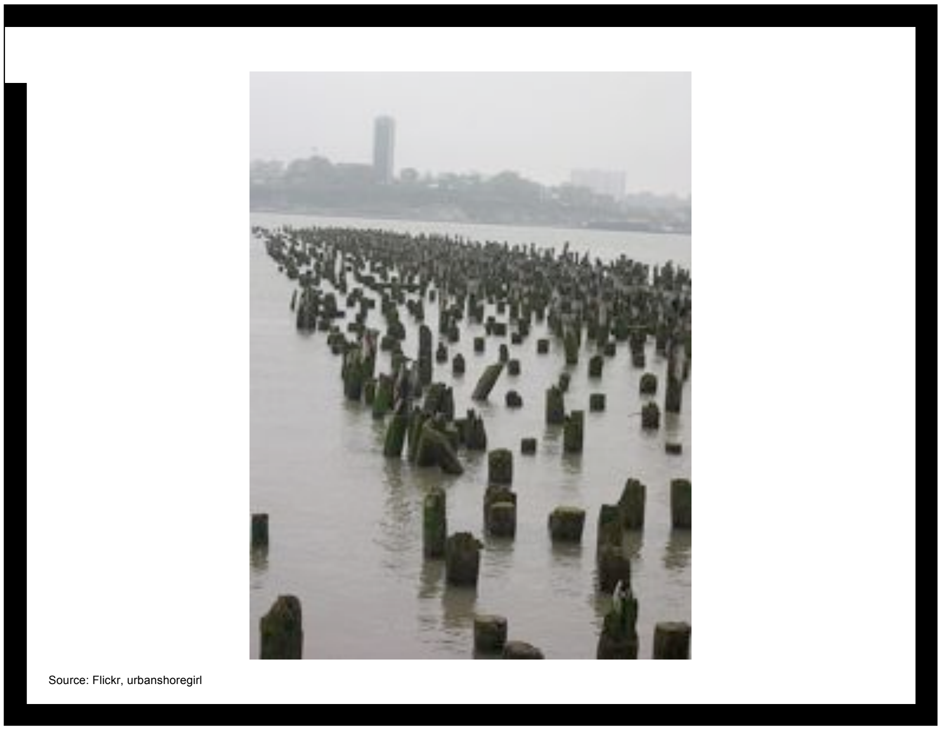
Let me show you a computer fridge that exhibits a lot of applianceness, in Sharpe’s definition. This is Whirlpool’s third attempt at mating a computer with a refrigerator. I think it’s the most successful so far.

Some background: Whirlpool owns KitchenAid, who make a famous mixer with attachments. Same idea here. You can buy widgets that plug into the front of the fridge. There’s a picture frame, an iPod dock, a DVD player, a calendar and a message board. They’re all computers, but each has its own interface tuned that task. It’s no longer about computation, it’s about the experience you can create with it.

3. PHYSICAL OBJECTS CAST INFORMATION SHADOWS

Now let me move on to another high-level concept we're working with. We started looking at what it would take to redesign everyday objects using ubiquitous computing technology. As we did, we realized that every material object casts what we call an "information shadow." In our modern world, everything exists simultaneously in the physical world and in the world of data. Nearly every object's information shadow can be examined and manipulated without having to touch the physical object. Think of the Amazon and Google book APIs. Information shadows have lives of their own. Those lives can be as rich as the physical object's life, maybe richer.

What are some objects with rich information lives? Well, there is media: books, CDs, images, etc. I suspect more people manipulate images of the Mona Lisa than have ever seen it in person. There is also anything on Amazon. Amazon's ASIN system allows people to reference and discuss virtually anything they sell because there's a unique identifier that gives you fast access to that object's information shadow. Then there is wine. I'll talk about our work with wine a little later, but wine has particularly active information shadows. Virtual Vineyard was the first ecommerce site. Bruce Sterling talks about wine extensively in Shaping Things, his book on ubiquitous computing, and wine has long been a favorite example for information architects.



5. DEVICES ARE SERVICE AVATARS

The other key hardware component of ubiquitous computing is networking, which brings me to the next idea. Networks mean that the same information can be accessed and manipulated through a variety of devices. Most value rests in information, rather in the device that’s communicating it, which means that the devices become secondary. A number of familiar information appliances--cell phones, ATMs--are basically worthless without the networks they’re attached to. They are physical manifestations, avatars, projections into physical space of services, but are not services themselves. You really start to see this in purely information entities: what’s a plane ticket? what’s money? what’s a book? They become subscriptions and agreements, for which a device becomes a nearly disposable channel.

This means that when thinking about how to design user experiences for ubiquitous computing, the design of the service becomes as important as the design of the device. This, is the genius of the iPod: it’s primarily an avatar of the iTunes Music Store. The Amazon Kindle, as questionably designed as it is, is a physical manifestation of the Amazon Kindle Store.

6. GRANULARITY DETERMINES KEY ASPECTS OF EXPERIENCE DESIGN

General purpose computers traditionally have interfaces that are person-scale. They're designed to be used in a wide variety of ways, and what typically makes sense is to make the input device about the size of your hands and the output about the size of your head.

Ubiquitous computing devices can come in all sorts of sizes and the user experience design for them must take this into account. This has been true since the earliest days at PARC when Weiser defined the tab, pad and board as names for the scales of the devices they were developing.

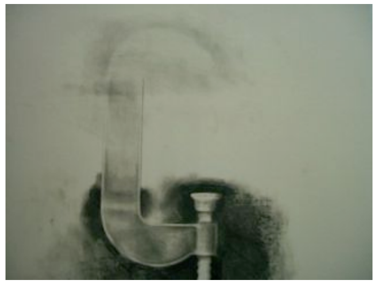
That made sense in the document-centric world of Xerox, but I think it's too limiting when thinking about everything that's possible with ubicomp.

From Flickr: watch by funadium, box by ubermichael, phone booth by rastrus, room by bigpinkcookie

Scale	Label	Examples
1 cm	covert	RFID, nail polish, cochlear implant
10 cm	mobile	phone handset, portable media player, wallet
1 m	personal	chair, car, ATM, payphone, laptop
10 m	environmental	wall, door, chandelier
100 m	architectural	church clock, billboard, bus
1000 m	urban	street intersection, landmark, crowd

This is the scale I've been using. It's a set of definitions to talk about granularity and it helps us identify that works and doesn't work at various scales. Screens don't work when you approach the covert scale, which is why wrist TVs have never taken off. Buttons don't work well on the environmental scale and above, because they're too small relative to the object. You probably can't make anything that's designed to be immediately social at anything above the environmental level.

7. MAGIC IS A POWERFUL UBICOMP INTERACTION METAPHOR
The last general idea I'd like to talk about is magic. Not as hand-waving to obscure functionality, but as a design metaphor like the desktop metaphor. Specifically, I mean using the concept of enchanted objects to generate ideas about interaction and as a way to create user experiences that are easier to explain. People have a tendency to create animist explanations for the behavior of technologies that exhibit unpredictable behaviors. They treat their Roombas like pets, they get mad at their laptops, they think their iPod is obsessed with a band, etc. We can use these natural associations to design ubiquitous computing interactions. Many existing ubicomp products already reference magic implicitly. This is Ambient Devices Ambient Orb, which is a kind of digital crystal ball. The Wiimote acts very much like a magic wand, as do all of the phones that have accelerometers in them. Wearables often take the form of amulets. There's still a lot of useful space left in the metaphor.



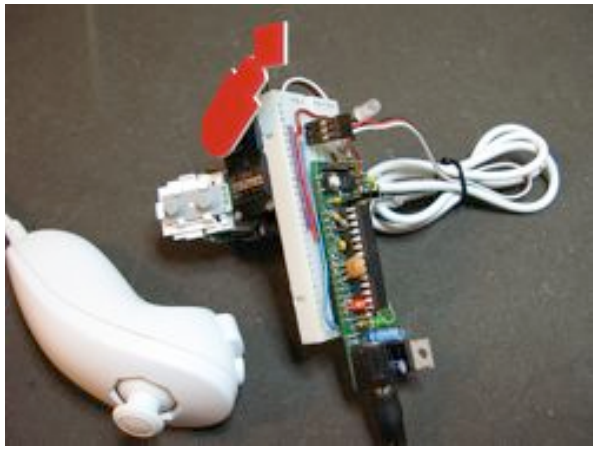
(Jim Dine, *Untitled* from *Ten Winter Tools*, 1973)

How does this all this theory work in practice? Well, since there are no best practices, you have to start somewhere. I went back to sketching. As Bill Buxton, who is scheduled to speak in June, points out in his book, sketching is not prototyping. It is not the first step in solving a problem, it is the process by which we understand the design space so we can define the problem in the first place.

We see sketching as the application of agile software development principles to design. The agile principles we embrace are

- Rapid iteration. We typically work on one-week cycles when sketching
- Barely sufficient implementation, in other words, doing just enough to move on
- Always having something that works, even if It's a paper mockup and fakes the functionality
- User research at every step
- Interdisciplinary collaboration from the start
- High adaptability, if our ideas are wrong, we move on and have other ideas

The first sketching technique we use is hacking. Which is kind of like using tracing paper, but with hardware. We take existing technology and attempt to extend its capabilities to see where things can go. What's possible? That's Tod, my business partner, on the left at Maker Faire last year. Our early hacking experiments led him to write a book about how to hack Roombas to make them draw, play music, work as giant computer mice, etc.

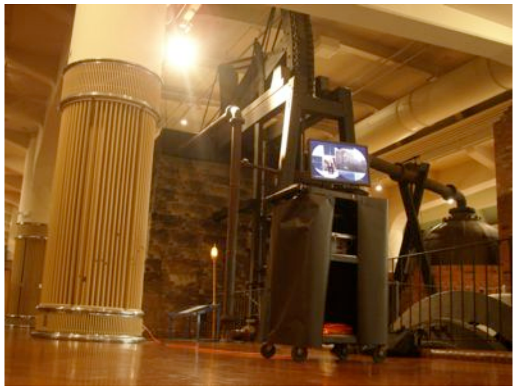


Here’s a hacked Nintendo Wiimote nunchuck. Tod used the accelerometer in the Wiimote nunchuck, connected to an Arduino open source hardware I/O board and a small servo to create a kind of mini-Segway mechanism that always points up.

Unfortunately, the tools for doing this kind of hacking, what we call sketching in hardware, are pretty primitive, but they’re coming up fast. Matt Cottam, who will be speaking here in September, has been doing very interesting work with enabling this kind of experimentation, and he and I started a conference--called appropriately Sketching in Hardware--to discuss these ideas.

Let me tell you about a hardware sketching project we did last year. We worked with the Henry Ford museum in Dearborn to help them understand how to make their enormous collection more relevant. They're in an interesting position: they have a collection of millions of artifacts related to the history of technology from the 17th century until today. The place covers nearly a square mile and to give you a sense of scale: they have a large collection of buildings and airplanes. Making all those objects relevant is hard. A retired engineer is going to have a very different perspective than a 12 year-old on a class trip and one set of wall text isn't going to satisfy both of them.

We did a series of sketches to see how ubiquitous computing technology could create different experiences for different audiences. Our goal was to understand the boundaries in the experience design space of the problem, rather than creating a finished solution. We began with an exhaustive literature search and once we knew what others had done, we started sketching. We spent about a week every month creating a completely new experience, with quasi-realistic content and semi-functional technology and then we tested it with real visitors. This is April’s. We made RFID tickets that selected one of a set of videos which told a different narrative for each user experience persona. There was one that more engineering focused, one that more about history, one that was a game, etc. Using our previous definitions, we designed the ticket and reader on the mobile scale, while the video was on the environmental.

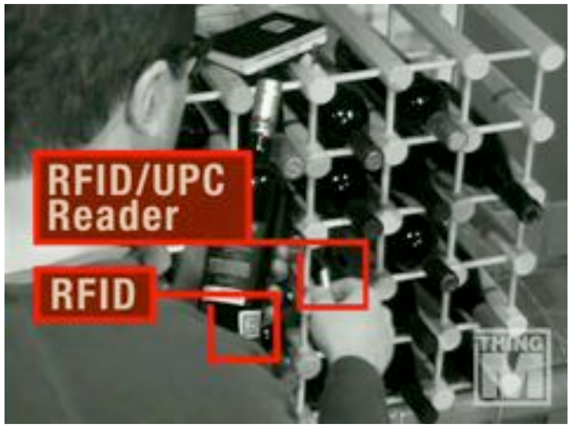


This is May's sketch. Our challenge here was to make sense of these enormous steam engines. These used to be in textile factories in England and moved water around giant locks in Birmingham. Few people even know how a steam engine works, and the relevance of the differences between different engines is difficult to explain. Since the machines were so large, we took our inspiration from coin-op binoculars and sketched out a set of magic binoculars, an enchanted spyglass, that are a kind of lightweight augmented reality. You see the scene in front of you, but there's additional information overlaid on it. With one setting, you see the machines in their original context. If it was out in a field pumping water out of a mine, you see it in a field. If it was in a mill in the middle of town, you see it in a mill. Another setting gives you a historical narrative with old photographs that float around each machine as you look at it, another turns each machine into its blueprint, with all the parts labeled. We made this by putting an optical mouse under a lazy susan and then moving images inside a vignette that looked like binoculars.

Here’s another sketch. We used a pair of Wiimotes to create a kind of magic wand that allows people to point at objects and get extra information about them. Depending on which wand they use, different information is projected around around each object. They can also pull the trigger and get additional in-depth audio description of that artifact.

This one, by the way, was really tough from a technological standpoint. Wiimotes are great, but they use infrared LEDs to orient themselves. The spotlights in the museum kept confusing the Wiimotes until we realized that we could just the lights themselves as the visual anchors.

We tested each of these prototypes with end users and used that to distill a set of design and development guidelines for how to think about systems like this in general.

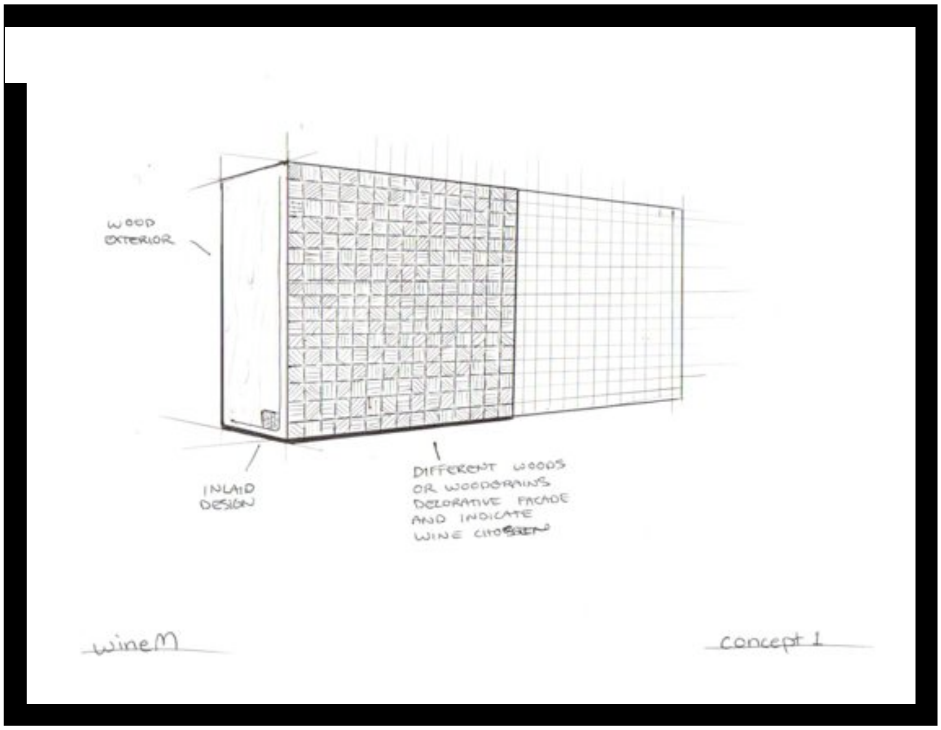


Another sketching technique we use is video prototyping, This again helps us imagine how a technology could work, without actually having to make it. We fake all of the technology using video so that we can concentrate on the interaction. Making the video is fun, but it's also incredibly valuable from a design perspective. We have to face a lot of our assumptions about the experience we're trying to create.

Last January, we made one about a smart wine rack.

[show video]

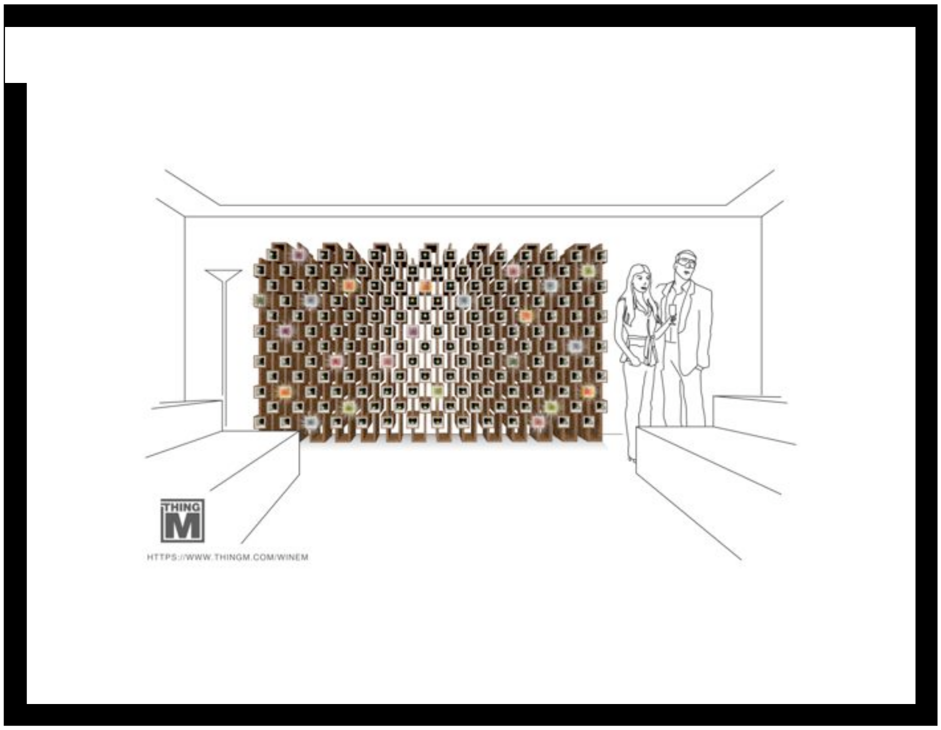
Having made this, we thought that we had learned a lot about how to interact with physical objects. We blogged about it, and moved on. But the blogosphere didn't let us get away that easily. For the next couple of months people kept asking us where they could buy it. Then, two months after we put the video out, Wired Invited us to exhibit it at NextFest and we agreed. The problem was that it didn't exist, it was a video.



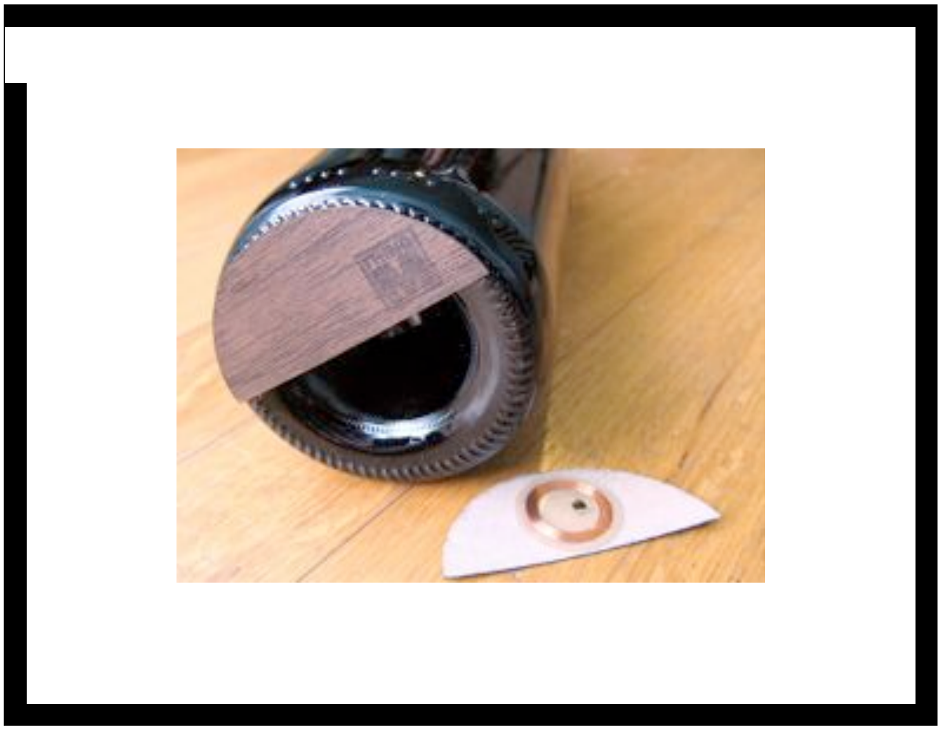
So we actually had to make it. Tod focused on solving the RFID engineering problems while I designed the user experience. I started by developing several personas with Ryan, the industrial designer we work with. Jack was the person we thought would be the primary buyer, but we didn’t designed this wine rack for Jack. We designed it for Vince, the Wired NextFest visitor, who had a very different set of needs and a very different esthetic. Our design needed to instantly communicate to Vince that this is a fundamental rethink of what a wine rack is. Here you can see how traditional industrial design and interaction design start bleeding into each other.

[Go to video, start talking]

Here are some of Ryan’s ideation sketches for a physical design of the rack that would work for Vince.



And here’s the final design. This is number 38 of 60.



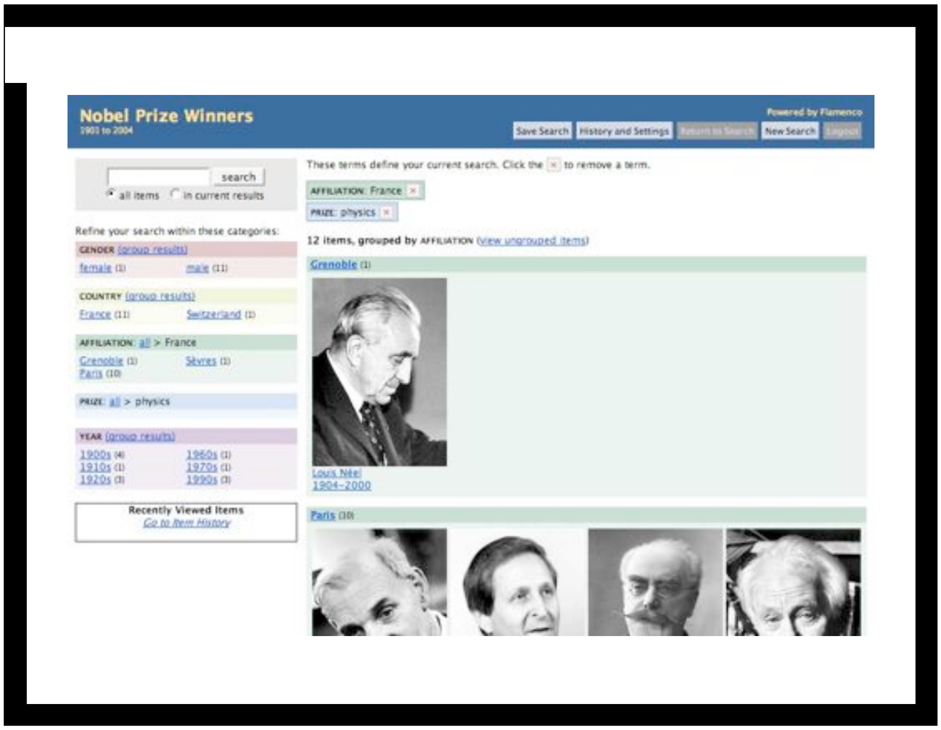
This is our RFID bottle tag. We wanted every piece of the design to communicate a set of values with the way it looked and worked. It’s made of the same walnut veneer as the rack, sits flat, isn’t visible when the bottle is on a table, and leaves enough space so that you can put your thumb in the back to pour the wine.

We also didn’t want to go the home automation technofetishism route and make “a wine rack that lets you see your wine inventory from anywhere on Earth with your phone.” Whatever. That’s easy, but not interesting to the person standing in front of it. This is a environmental-scale product, not an urban one.

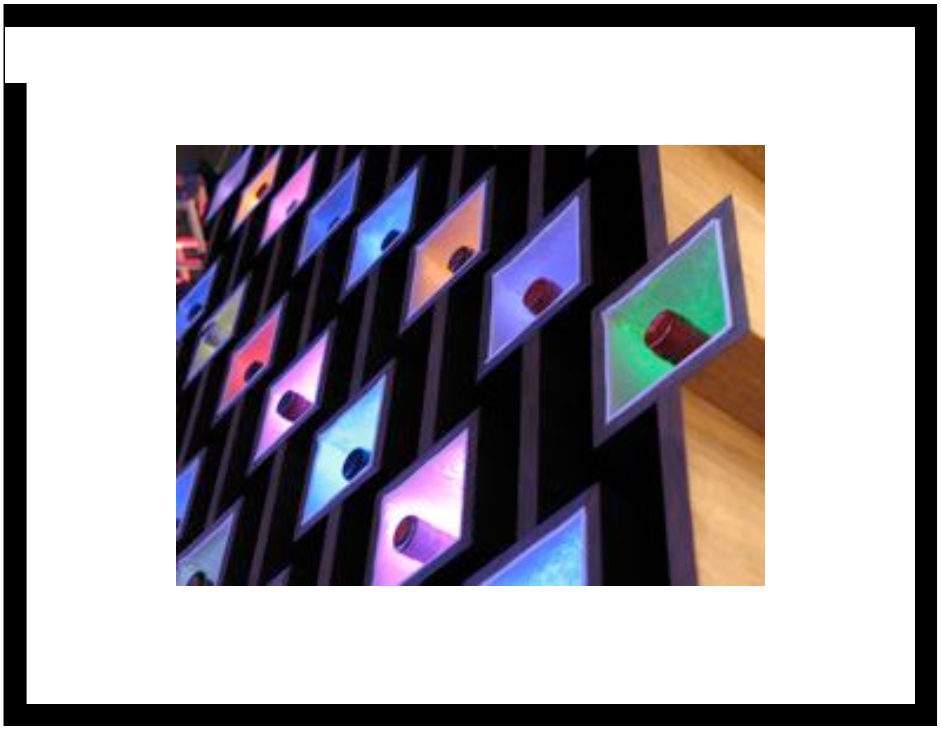
We tried to really treat the rack as an appliance and avoid using screens entirely and design only for the environmental scale, but we couldn't come up with a workable solution, so we decided to make a control panel that detached from the main unit and used as little of the visual language of software as possible. We used Nokia wifi touch tablets as control panels. We love these tablets for prototyping. They have beautiful high resolution touchscreens, connect to the Net over Wifi, they have decent Web browsers, and they run Flash. The problem is that there's not that much screen real estate…

Source: Qliktech

…and there's a lot of wine information. To deal with this I decided to use a faceted classification system. I felt that facets would allow people to explore the information they needed to choose a bottle of wine given a limited amount of screen real estate, and only using their index finger.



But faceted classification systems typically use a lot of screen real estate, with lots of little things that would be impossible to click on with your fingertip on a small screen.



This was a pretty big challenge. But this is where what we do diverges from traditional UI design, and I think to our advantage. This is not our appliance. [click] This is. [click again] This is a control panel, so it doesn’t have to show all the relevant information all the time and can focus on navigation.

So we implemented this as a kind of tabbed interface. The tabs are facet categories and they do two things: they navigate the information space on the control panel and they project corresponding information on the bottles in the rack. When you click a tab it lights up all of the bottles in different colors that match the facets. Maybe that’s all Vince needs to choose a wine. The interface does not assume that he has to narrow down to one answer until something is displayed on the rack. It’s a decision support tool at every action.

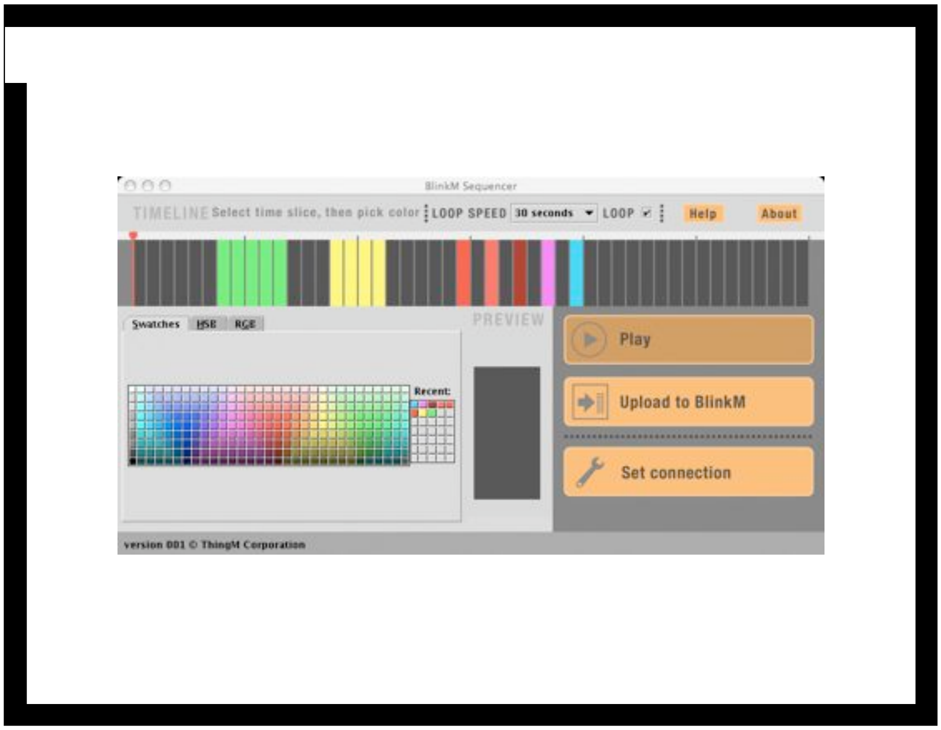
Let me show you how it works. [show demo] This is the 8th version of this UI. Our first version of this UI was on a piece of paper. The second was a bunch of images that didn’t actually have any interactivity.

When a facet is selected, it’s added to the shelf on top, that’s that empty area above. You select a facet, the facet appears on the shelf and the wines that match all the shelf constraints go white, while the ones that don’t, turn off. If you then select another category, just the ones that are currently lit stay lit, but in the colors of the selected category. Getting that sequence right took about 5 iterations and my intuition was proven wrong time and again. Which, of course, is the point of iteration.

Now let’s put all of the pieces together. Let me show you the system in action. [show video]

So that’s how we backed into our first major product. We’re actually making this a product now, and we prices by it by the end of March.

I’d like to wrap up by talking about something different. Not completely different, but different. We’re a consumer product company, but we recognize that there are other markets. Our work doing the Sketching in Hardware conferences and Maker Faire showed us that there's a need for better user experience design around basic electronic components and that ubicomp technology can satisfy that need. Making an LEDs light up is easy, but making an RGB LED a specific color or blink in a specific pattern is hard. You have to know about color theory and pulse-width modulation and power. We decided to make it easy and to see if we could push the concept of ubiquitous computing all the way down to the level of individual components. BlinkM is a smart LED. You give it an RGB triplet, it knows how to make it glow that color. You give it two, it’ll do a smooth fade between them through color space. You give it a pattern, it’ll blink in that pattern. Once programmed, it works just like an ordinary LED. It’s a slightly enchanted LED. The component as a tiny appliance.



Here's the sequencer software for it. It's a combination of a color picker interface and drum machine sequencer. The BlinkMs also network together and take input, so you can make a string of smart Christmas lights that react to sound or that generate new patterns based on the time color of the sky.

BlinkMs should be available in about two weeks from Sparkfun electronics and should cost about twelve bucks. The software is all open source and downloadable.

We're looking forward to sketching with them, to designing with our new toys. But mostly, we look forward to what this means. If two guys can hack together all of the stuff that we've done over the last year, the future of design is going to be an exciting and challenging one.

THING
M

Mike Kuniavsky
mikek@thingm.com

Thanks!