Designing Everyday Computational Things
Why Industrial Design will be the New Interaction Design
David Holman, Human Media Lab, Queen’s University
Roel Vertegaal, Human Media Lab, Queen’s University

The art of user interface design is on the cusp of a revolutionary change, one that will require a renewed consideration of the bearing of material and form on interface design. Flexible display materials\(^1\), and their three-dimensional expressiveness, expand the ways computer interfaces can be designed and—unlike before—enable designers to contextualize interaction in physical shape. No longer constrained to the flat surfaces of tablets, mobile phones, or desktop computers, these new displays will be wrapped around three-dimensional objects and, inevitably, envelop the everyday things we use [12]. The morning newspaper, credit cards\(^2\), light switches, or even kitchen plates will retain their everyday identity, yet will be augmented with a seamless and interactive high-resolution display skin.

The observation that current interface design is restrictively flat, much like the characters in Abbot’s Flatland [1], originates from our earlier research in Organic User Interfaces (OUIs). OUIs are a new category of computers that feature displays of almost any form: curved, spherical, flexible, actuated or arbitrary [9]. By design, they are non-planar computer interfaces that express both rigid and deformable shapes and are capable of sensing input and rendering dynamic content anywhere on their surface.

PaperWindows [8] is an early example of an OUI that simulates the use of digital paper displays (see Figure 1). Using a combination of motion tracking and projection, computer windows are rendered onto a piece of paper giving the illusion that the paper is, in fact, an interactive display. This metaphor is later instantiated in PaperPhone [10], a paper computer that uses flexible E Ink to bend the interface and express interaction (see Figure 2). Both these interfaces extrapolate to a scenario where we will have many paper computers surrounding us, thin enough to be tossed around, stacked, and manipulated much like real paper.

Moving from flat to organic design poses a series of design questions: how will users interact with such oddly shaped displays? What will their user interfaces look like? It is clear they will be very different from the ones we use today and it is necessary to carve out a set of organic design principles. After exploring organic design by enveloping varied objects with curved displays skins, the Dynacan [2] being another example (see Figure 3), we proposed the following three principles for OUI design:

1 - Input Equals Output

In a GUI, input devices are indirect and separable from output devices, far different from interactions with physical objects. Paper documents can be stacked, manipulated, and tossed around with a kind of physical immersion that is simply inconceivable in a desk computer. This immersion can only be attained through a synchronized fluidity between multi-finger, two-handed manipulations based on the rich visual, haptic and auditory representations on the real world. OUI displays render as a real-world object, folded into the shape that best supports data interpretation; users literally touch and deform the graphics directly on display.

2 - Function Equals Form

---

\(^1\) Specifically, Flexible Organic Light Emitting Diodes (FOLEDs) and Flexible Electrophoretic Ink (E Ink) [10]

\(^2\) Bank of America’s “SafePass®” debit card already integrates a digital display that shows a secure code for financial transactions. http://www.bankofamerica.com/privacy/index.cfm?template=learn_about_safepass
This principle echoes Gibson’s ecological approach to visual perception [6] and embodies the concept of an affordance. Specifically, the form of an object determines how we interact with it. Picking up a display activates it for input. Likewise, bending the top right corner of the display inwards may invoke a paging down action, while bending it outwards pages up. That is, the physical embodiments of interactions are grounded in useful and appropriate affordances.

3 - Form Follows Flow

OUIs adapt their form to better suit different contexts of use. Their shape transforms to the flow of user activities in a range of physical and social manifestations. A simple example is the folding of clamshell cellphones. Opening the clamshell turns on the phone, a clear and compelling affordance. Closing it ends a call and deactivates the keys, all in one go.

From Organic Design to Computational Things

Guided by these principles, our exploration of OUIs ultimately led to a critical observation: their hands-on form made them feel less like devices, and more like something encountered in everyday life: a real thing. To the user, reaching for a paper window, bending PaperPhone to answer a call, or tapping Dynacan after taking a drink is nothing like using a computer.

These experiences point to technology’s maturation to that of a computational material. When interactive hardware is commoditized and seamlessly melded with everyday things, the ‘computer’ disappears and is better seen as a basic material that happens to have interactive behavior. Computation is less like hardware and more akin to a design material, not unlike the plastics, wood, or ceramics used by a designer. This transformation, of course, presupposes a three-dimensional form in user interface design, one that entails an abundance of shapes, curves, and topologies. Treating computation as a material, then, raises a fundamental question for user interface design: how should three-dimensional interactive form be designed and explored when interaction is everywhere?

We argue that the interaction designer trained for software design may not be appropriately equipped for this new design context. Computational things need to be designed—or sculpted even—with an awareness of three-dimensional form and its interplay with material. As web design was gradually taken over by trained graphics designers, industrial designers, in a similar trend, will be the first to pragmatically encounter and reflect on the design of everyday computational things. This is not to say that interaction design will be antiquated. Instead, a new profession will arise, one that blends industrial and interactive perspectives and explores areas traditionally untouched by interaction design. Interaction will be everywhere, in the form of everyday computational things.

Interactive Sketching

We must point out the complexity this poses to the design process. Industrial artifacts operate mostly in a spatial domain. Interaction introduces a rich temporal dimension, one that complicates representing even simple interactive ideas. How will designers rapidly sketch using computational materials? In Sketching User Experiences [4], Bill Buxton argues that sketching—in itself—is not about the material form it embodies, such as an architect’s pencil drawing or an automotive designer’s clay sculpture, but more of an abstract activity that impacts design thinking and learning. To support designers, we must update our design tools to answer the challenge of seamlessly prototyping interactive systems.

With this in mind, we started a dialogue with industrial designers at Microsoft’s Hardware Input Group [7] to see how they approach the exploration of interactive systems. Of the hundreds of prototypes their designers sculpt, only a few will eventually have interactive functionality. Bread
boarding—as they refer to it—integrates hardware components into a prototype to roughly approximate its final interactive behavior. However, it is laborious, impacts ergonomics, and requires detailed knowledge of embedded hardware design. In most cases, it quickly devolves to a disjoint exploration of form and interaction that relies on synchronized designers mimicking input behavior, in place of a functioning prototype.

Clearly, this style of work is unsuitable for exploring computational things; what if a designer could explore interactive behavior without a functioning prototype? We argue that exploring interactive designs should be as seamless as working with other common design materials, such as tape, foam core, or others. To do this requires transforming passive design materials and imbuing them with innate input sensing, leading to a style of work best described as interactive sculpting. Doing so helps a designer experience a working device hands-on and is useful for triggering the serendipitous realizations arrived at with a concrete artifact.

One approach is to integrate sensing technology directly into the sketching materials already used by designers. One example we constructed is TouchTape (see Figure 4), a one-dimensional pliable touch sensor that looks and behaves like regular tape. TouchTape can be built from everyday supplies: an H2 pencil (resistive surface), tin foil (conductive surface), and a shelf liner (spacing material). We position it as a readily available material in the design studio. When a designer wishes to add touch sensitivity to an industrial prototype, they grab a roll of TouchTape, cut off a piece, and attach it to the surface. To sense touch input, the designer only needs to connect two wires: one to each of the resistive and conductive surfaces. As a hardware sensor, its electrical behavior is similar to common sensors used with Arduino [3]. Using TouchTape, the designer can explore touch input on a variety of curved and deformable surfaces: spheres, coffee cups, bracelets, paper, credit cards [12], dynacans [9], and so on.

Although this approach is in line with Corlett’s [5] discussion of physical computing and the importance of usable hardware platforms (i.e. Arduino), it still requires a certain level of technical proficiency to operate TouchTape. What if we could get rid of this hardware layer altogether?

This idea led to the design and implementation of SketchSpace [9], a lightweight environment that adds implicit input sensing to passive physical material (see Figure 5). We envision SketchSpace as a tool and set of interaction techniques that allows a designer to use passive materials as a means to roughly simulate the interactive behavior of input devices early in the design process. All input sensing in SketchSpace is achieved using Microsoft’s Kinect camera. Its depth sensing allows SketchSpace to infer a broad set of inputs, ranging from touch events on the object’s surface and surrounding workspace, orientation, position, motion, proximity, and, among others, shape deformations. A designer uses these inputs to prototype as if it had functionality, including mapping sensor values to interactions using embodied gestures and projecting dynamic interactive content on it, all without finding, attaching, and working with a collection of physical sensors. The designer works hands-on by tapping mapping icons that are projected on their workspace table and, using physical manipulations of an object in situ, to specify when and how these mappings trigger interactive behavior. Using a tool like this frees the designer from hardware constraints and helps them ideate and quickly explore interactive sketching.

Towards Hyper-contextualized Design

If interaction is truly everywhere and tools like SketchSpace and TouchTape help designers rapidly explore interactive form, what principles should guide them when designing? One thing is clear: the interfaces that run on these embedded displays should not be designed for generic computer activities. Instead, their interactive behavior should only express a few essential actions that are contingent on their form factor. This distinction moves away from the approach of the computer as a generic tool [4] and introduces cautious specializations in the interface. Computational things take this approach one step further by hyper-contextualizing the interface. A
credit card with a thin film display would show your balance and banking activity, but it would not be used to write an email or browse the web. It might have an interactive map on the back, but its interactive behavior would be limited to helping you perform mobile financial transactions or finding a nearby branch. In a similar example, a reusable water bottle might subtly glow when it is near a water fountain, only if it is almost empty.

This stringent minimalism has clear benefits: it limits the range of possibilities a user encounters when using a three-dimensional interface. It also leverages *in situ* interface design, in that physical manipulations of a computational thing can be more precisely mapped, even made more usable, when functionality is limited and excels at doing only one or two things. Keeping the back of the credit card as *just a map*, as opposed to including a web browser or any other ‘useful’ applications, means that the user simply flips their credit card to see the nearest branch. This minimizes application switching, displaces a cumbersome smartphone, and embeds this information in a predictable locale, one that is closely tied to physical starting point of their banking experience.

Thinking about hyper-contextuality applied to interface design is only a first step. How designers should go about representing it in three-dimensional interface design requires elaboration. Unlike the User Centered Design process, settling on the functions that are most critical to embody in an everyday computational thing is not necessarily linear or even formulaic. First, the context in which an interface *exists* has a much deeper impact than it did with flat interfaces. Thinking back to our reusable water bottle, a designer must deeply consider its size, material, relationship to owner, and, among other aspects, the environment in which its identity plays out. Traditionally, this is the domain of industrial design. However, designing a water bottle made from a *computational material* must also account for the ways in which the bottle can be augmented with interactive form. A starting point is a simple question grounded in McLuhan’s law of extension [11]: of the functions (or experiences) this water bottle exhibits (or should exhibit), which of them would be enhanced, intensified, or made possible by imbuing them with interactive behavior? Perhaps calmly revealing the temperature of the water after it is filled up would impact a person’s choice of water fountain. Or maybe tapping the bottle at a certain spot would reveal how much water the owner has consumed. For even simple things, the possibilities quickly exhaust themselves. It is at the discretion of the designer to decide how the interface will be hyper-contextualized (or if it should be at all).

Given a judicious hyper-contextual reduction, the last step envisions how the interactions for each critical function should be explored when designing. This is a new design approach, one that merges industrial and interaction design and binds input sensing to form. Again, building an interactive system should be as seamless as working with common design materials. We envision a designer tasked with exploring our water bottle’s interactive potential using a system like SketchSpace. They would rummage the design studio looking for a water bottle. After finding one and placing it in their workspace, they would touch, swipe, shake, squeeze, and, in general, repeatedly interact with it using the digital tabletop interface to explore different digital mockups of its interface (see Figure 5). In doing so, the designer quickly and rapidly explores multiple interactive design paths, before settling on those that feel right.

**Conclusion**

This paper argues that the best thing that could happen to user interface design is for computers to stop being technological devices and start being more like *real* everyday things. Their hyper-contextualized interfaces will feature the same kind of skins we find on products today, but extended with interactive behavior. Designing these computational things to excel at perhaps one or two functions at a time radically simplifies the design of the user interface and ensures that the ‘computer’ dissolves from the forefront. Interactive hardware will be a mere commodity to the industrial designer, to point that it looks and feels like any other design material. This, naturally, accelerates the need for new tools that allow designers to sketch using computational materials.
This is a turning point for user interface design, one that challenges us to understand how to best design in this new world of everyday computational things.
Figure 1. *PaperWindows* explores a set of interaction techniques for digital paper interfaces.

http://www.youtube.com/watch?v=gVESlp0BicE

Figure 2. *PaperPhone* is an interactive flexible paper computer that uses bending as a way of navigating.

http://www.youtube.com/watch?v=RI-qygUEE2c

Figure 3. The *Dynacan* is a pop-can computer that embodies a cylindrical interactive display and demonstrates an application in mass consumerism.

Figure 4. A typical *TouchTape* scenario. A piece is cut from the roll (foreground) and attached to an hourglass. An Arduino relays touch data to a desktop computer.
Figure 5. SketchSpace tracks and projects a texture on a roughly sculpted mouse. The icons on the top, left, and bottom correspond to virtual sensors. The icons on the right are for displaying images and adding or removing virtual buttons.
References


