

Sketching Digitally Controlled Physical Products with Shared Phidgets

Saul Greenberg
Dept Computer Science
University of Calgary
Calgary, AB, CANADA T2N 1N4
saul.greenberg@ucalgary.ca

Nicolai Marquardt
Faculty of Media
Bauhaus-University Weimar
Weimar, Germany 99423
nicolai.marquardt@medien.uni-weimar.de

ABSTRACT

Within design, a sketch serves to develop and illustrate the essence of an idea. For digitally controlled physical products, we argue that designers can augment paper-based sketches with interactive working models as long they are quick to make. Using working models as sketches are beneficial, as they can help the designer reflect upon various characteristics of an idea: interaction flow over time, object animation, form factors, contextual locations, and so on. In this paper, we show by example how Phidgets hardware and the Shared Phidgets software toolkit can be used as a medium for sketching networks of digitally controlled physical products. When such systems are realized as interactive sketches, they serve as a vehicle that helps a designer make vague ideas concrete, reflect on possible problems and uses, discover alternate new ideas and refine current ones.

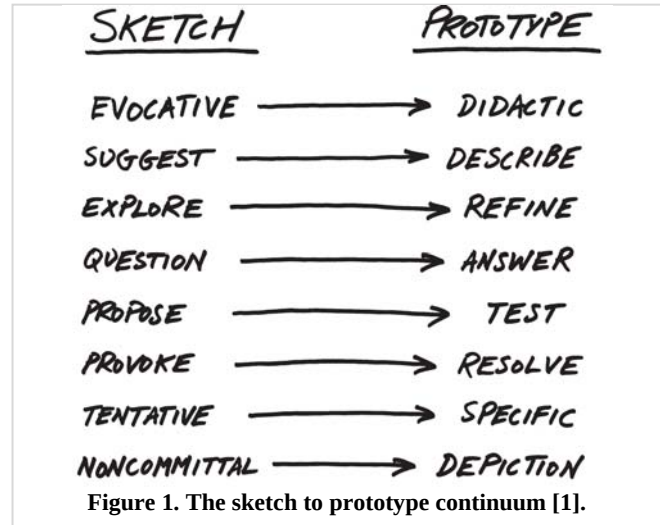
1. INTRODUCTION

Designers create sketches – typical using paper and pencil – as a means to articulate and explore the essence of an idea, to rapidly generate multiple alternatives, and to explore their implications [1][7]. Buxton explains the fundamental properties of a sketch, where he contrasts them to prototypes (Figure 1). A sketch is exploratory, evocative, tentative and non-committal, and thus has many rough and/or undeveloped aspects to it. Effective sketches are also rapid and easy to make, to the point that a designer is not only willing to make many of them (each representing different ideas), but is also willing to discard them. Designers also use sketches as conversational props to encourage discussion with others, i.e., to propose, question and provoke as part of both informal and formal design crits. When an early design is displayed as a sketch, the team recognizes it as an idea to be worked on and developed further.

While paper and pencil sketches are often used by user interface designers, other media for creating sketches are becoming increasingly accepted: storyboarding, animated slide shows, video filming and editing, and even interface builders [1]. The advantage of this new media is that they not only illustrate the look of the system, but its interaction flow over time. This is important: when ideas are portrayed as interactive vs. static sketches, the media serves as a vehicle that helps a designer make vague ideas about interaction concrete, reflect on possible problems and uses, discover alternate new ideas, and refine current ones.

Now consider physical products. They are unlike traditional user

Submitted to Physicality & Design Workshop, held at the ACM DIS Conference, February 25–27, 2008, Cape Town, South Africa.



interfaces as they are three-dimensional, and are usually surrounded by a physical context (the local environment). This is why many industrial engineers resort to physical models to capture attributes that are muted by a 2D sketch. The question is: can we can apply the idea of a 'physical model as sketch' to highly interactive digitally-controlled physical products?

We will argue that toolkits designed for rapid prototyping of physical user interfaces can also serve as a sketching platform. In particular, our experiences with our own Phidgets [4] and Shared Phidgets [5] toolkit over several years suggest that it is a reasonable 'sketching' medium for programmers to rapidly build and animate digitally controlled physical products distributed over a network. Phidgets provide the hardware building blocks for devices: sensors, actuators, RFID, and so on. Shared Phidgets provide a distributed platform that lets a programmer link hardware building blocks distributed across multiple computers and locations into a single system.

To illustrate what we mean, we first illustrate how one can quickly create a sketch of a distributed physical interface linking two homes, which in turns supports aging in place. We then take a step back, where we discuss how several properties of Shared Phidgets afford its use as a sketching medium.

2. MONITORING ELDERLY ACTIVITY

Aging in place occurs when elderly people, who are at the edge of being able to look after themselves, are still able to live independently in their home. Assistive technologies have a strong role to play in these situations. One way to support aging in place is to somehow allow the elderly's loved ones – geographically

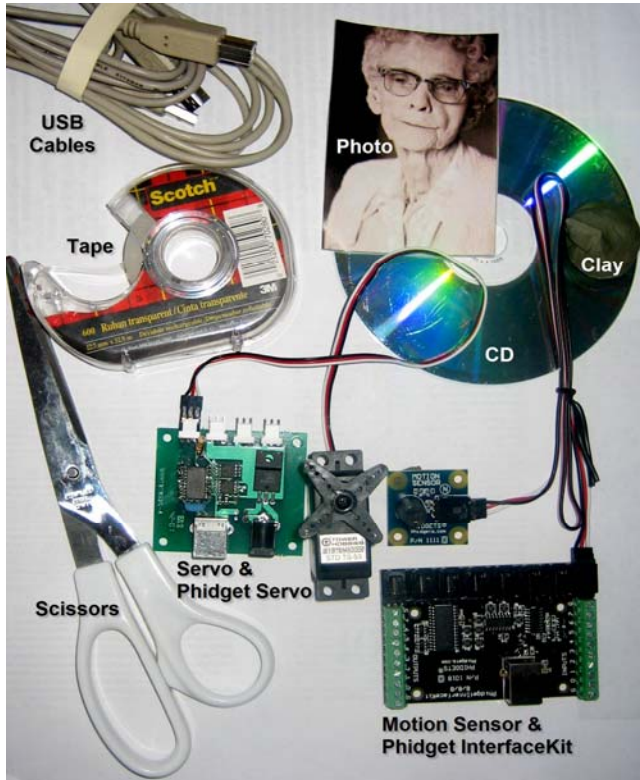


Figure 2. The raw materials

separated but nearby younger family members or members of one's extended community – to somehow monitor the well-being of that person in the home (e.g., [6]).

Now consider one simple idea for such a monitoring system that comprises two primary parts:

- i) One or more sensors are strategically located in the home of the elderly person, where the sensor detects activity;
- ii) An ambient display consisting of a photo of that person is located in the care-giver's home, where the photo is animated to portray the current activity level as captured by the sensors.

2.1 The Sketch

We will now sketch out our first version of this idea. For (i), we will use a single motion sensor. For (ii), we will mount a photo on a servo motor, and have its movement correspond to the readings of the motion sensor.

All the steps and approximate times to implement each step are detailed below. We include this detail to reveal the simplicity of this process; it is not intended to be a tutorial to the underlying Phidget / Shared Phidget system.

Materials (see Figure 2)

- servo motor, Phidget Servo,
- motion sensor, Phidget Interface Kit, USB cables.
- small photo of the elderly person,
- discarded CD, clay, tape and scissors.

Assembly (see Figure 3) - about 3 minutes

- a) **Ambient display.** As shown in Figure 3a, tape the photo on the CD, and use the scissors to cut the CD to fit the photo size. This gives the photo a rigid backing. Place a blob of clay on



Figure 3. a) the Ambient Display, b) located in the home message area, and c) on the home fireplace mantle.



Figure 4. Motion sensor, located in: a) kitchen and b) living room

the servo motor platform, and stick the photo onto it. Then place a blob of clay on the bottom of the servo motor so it can be placed atop a surface without tipping over (e.g., as in Figure 3b and 3c).

- b) **Both.** Plug the Phidget Servo and the motion sensor/Phidget Interface Kit into one or two computers (using two allows you to locate the sensor and ambient display in different locations and different homes).

Program the sensor and servo motor connection - about 3 minutes

- c) Start the Shared Phidgets Connector (a simple menu option). This server-based software initializes the server and run-time architecture. It automatically connects to any Phidgets plugged into the local computer, and lists Phidgets on other computers that are connected to this server.
- d) (optional) Start the Shared Phidgets Explorer (another menu option). Open the view representing the Interface Kit, and check to see how the sensor responds to motion in the room. Similarly, check to see how the picture is animated by the motor's range of motion. Programming is not required for this.
- e) Start a Visual studio C# project. Using the interface builder:
 - drop a Connection Manager Skin object into the window and set its URL property to the location of the server.
 - drop in a Servo and ServoSkin object, and link them together by setting the servo property; repeat this for the InterfaceKit and SensorSkin object (the sensor skin connects to a particular sensor on the InterfaceKit, shows its current value, and graphs that value over time).
- f) Create a SensorChange event for the InterfaceKit. Insert this code to animate the motor with the sensor's current value:


```
servo1.Motors[0].Position = e.Value * 180/1000;
```
- g) Run the program. The picture's movement is now linked to sensor activity, and the display shows the current value of both the sensor and the motor, and a graph of the motion sensor value changes over time (Figure 5).

While the above looks complex, all steps are quite basic operations well known to a Shared Phidgets programmer. Excepting for the 1 line of code, all operations are done by drag and drop, menu selections, and simple form filling.

2.2 Using the Sketch

The programmer can now examine, explore, critique and alter this 'sketch', perhaps in discussion with one or more others, from a variety of perspectives. Some examples are listed below.

The moving picture frame as an ambient display. The picture frame can be animated either directly through the testing interface (step d) or by moving in front of the sensor (step g). At this point, one can consider the overall effectiveness of the device. Are the sounds produced by the motors too soft or loud? Is the range of motion of the picture reasonable, or is it too distracting or imperceptible?

The effectiveness of the sensor. People can walk in front of the sensor to experiment with the field of view of the device. The actual values returned can also be monitored, either as they are being generated or over time by examining the graph (Figure 5). Questions that one can ask include whether this type of sensor is appropriate to capture activity, whether the range of values returned are adequate to drive the picture frame, and whether noise is captured, giving false readings.

Placing the sketch into context. This is a working sketch, where the sensor and photo display can be placed in separate homes (assuming internet connection) and react accordingly in real time. Figure 3 shows two possible locations for the picture frame in a home: on a mantle in the living room, or by the message area near the phone. Similarly, Figure 4 shows possible locations for the sensor: in the kitchen, or by a seating area. Contextual issues in

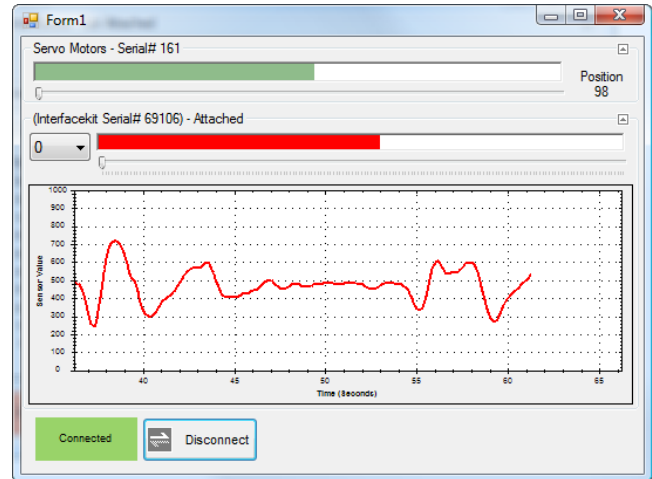


Figure 5. Screen snapshot of the above program. The top bars show the current motor position and sensor value. The graph plots the sensor value over time.

these locations can now be considered. This includes pragmatic questions (e.g., wiring), aesthetic questions (e.g., appearance), appliance visibility, usage questions (e.g., frequency of a person being in these areas), and so on.

Alternate sensors. An important property of a sketch is that it helps a designer generate alternate ideas. Because Phidgets allow sensors to be hot-swapped, the designer can experiment with other sensors in real time and in context. This can be done as the system is running and without any code changes. For example, a pressure sensor could be substituted to detect seating activity; a light sensor can detect a room's lighting levels and changes; and a switch can detect a door being opened or closed.

A multitude of sensors. The designer can also investigate how more than one sensor could work together to animate this display. This too requires little in the way of extra coding, and can be done quickly.

Alternate ambient displays. Similarly, the designer can rapidly reconsider the ambient display. This could include altering the physical properties of the display without coding (different sized pictures, different pictures, use of sculptures instead of pictures on the motor platform), extending the existing device (e.g., adding lights, which would require only a very modest amount of additional coding), or even building a completely different ambient display that works on the same sensor data. As well, computer displays (such as the graph in Figure 5) could be considered if a more foreground style of information display is desired.

Participatory sketching. Because the technology is working and can be located in context, the designer can use it to stimulate conversation not only with other designers, but with the end customer as well.

3. DISCUSSION

Phidgets were designed with rapid prototyping in mind, so it is no surprise that it also works reasonably well as a sketching medium. As Buxton argues [1], it is the designer's purpose that differentiates whether the created artifact is a sketch, a prototype, or something in-between (Figure 1). In practice, we have had

myriads of students, all with only a modest amount of training (e.g., a 1-hour tutorial) develop a variety of quite different physical devices in a short time. We originally thought of these as prototypes, but following Figure 1 we realized later that their examples were best viewed as working sketches. Examples are documented through videos available at <http://grouplab.cpsc.ucalgary.ca/> (search for Phidgets), and through various publications that argue how creativity in new domains is supported through easy-to-use toolkits [2][3].

The beneficial properties of Shared Phidgets for sketching include the following.

- A broad library of reasonably affordable actuators, sensors and controls are available (see www.phidgets.com); these become the building blocks for the sketches.
- Only basic programming skills are necessary. No knowledge of electronics, circuit board design, or networking is needed.
- Shared Phidgets transparently support distributed objects. Thus devices can be scattered and relocated across various physical contexts, where the environment then becomes part of the sketch.
- All Phidgets are accessible through a standard interface builder. This includes not only the low-level object whose properties can be programmatically manipulated and retrieved, but interface skins that let programmers interactively see and set these properties through a GUI.
- A broad set of high-level tools are included that allow people to monitor Phidgets input (e.g., sensor values) and control Phidgets output (e.g., motor position, on/off states) without programming (see step d above).
- A broad set of visualization tools are included that lets designers graph input values over time (e.g., sensor values over the course of a day) – see Figure 5. This is helpful to see what actual input is produced in various contexts.
- Phidgets are modest in size. Thus they are amenable for packaging into various form factors using traditional media such as clay, cardboard, glue, foam board, wood, tape, etc.

The above properties help realize one of Shneiderman's principles for creativity support tools: *Design with low thresholds, high ceilings, and wide walls* [7]. He explains by stating that such tools should be easy for novices to start using, while still affording advanced possibilities for experts who wish to tap into a wide range of functions. Greenberg [2] restates this as a similar principle: *simple ideas should be simple to do, while hard things should be possible*. In this manner, the toolkit becomes a language that influences creative thoughts, suggests design direction, and lets people concentrate on the design rather than on implementation details [2].

Critique. Yet Phidgets are not perfect, and it is just as important to realize its weaknesses as well as its strengths, as listed below.

- While many Phidgets are available, designers are still limited to the hardware building blocks supplied by the company. This may overly constrain the development of certain design directions, especially if the designer takes the path of least resistance.
- Phidgets are wired, not wireless. This means that they must be tethered to a computer, usually 2-8 feet away.

- For the same reason, cable placement and management is problematic or visually obtrusive in some contexts.
- Fastening Phidgets can be tedious. While glue and tape work in the short term, one would typically use nuts and bolts or other fastening system if robust fastening is needed. In contrast, toy robotic systems such as Lego Mindstorms are fashioned with rapid and reliable fastening in mind.
- While Phidgets can be programmed in a wide variety of computer platforms (Mac, Windows, Linux) and languages (C#, Java, Flash, etc.), the Shared Phidgets version described here is limited to .Net languages and to Windows.

Most of these limitations affect how robust, deployable devices are created and used in practice. Yet because sketches are caricatures of what could be rather than prototypes of what should be, Phidgets still work as a modest fidelity sketch representation in many cases.

Availability. Shared Phidgets is academic software, freely available at <http://grouplab.cpsc.ucalgary.ca/cookbook/>. Phidget hardware are commercial products, and can be purchased from <http://www.phidgets.com>.

4. ACKNOWLEDGMENTS

Thanks to Bill Buxton: our conversations help me perceive the value of Phidgets as sketches. Research is partially funded by Canada's NSERC Discovery and NECTAR Networks Grant program, by the NSERC/iCORE/Smart Industrial Chair, and by our industrial sponsors SMART Technologies Inc. and Microsoft.

5. REFERENCES

- [1] Buxton, B. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann, (2007).
- [2] Greenberg, S. (2007) Toolkits and Interface Creativity. *Journal Multimedia Tools and Applications (JMTA)*, 32(2), February. Springer. p139-159. (Special Issue on Groupware).
- [3] Greenberg, S. (2005) Collaborative Physical User Interfaces. In K. Okada, T. Hoshi and T. Inoue (Editors) *Communication and Collaboration Support Systems*. 24-42, IOS Press (Amsterdam, The Netherlands), ISBN: 1-58603-514-2, June.
- [4] Greenberg, S. and Fitchett, C. (2001) Phidgets: Easy Development of Physical Interfaces through Physical Widgets. *Proc Symposium on User Interface Software and Technology (UIST'01)*, 209-218, ACM Press. Includes video figure.
- [5] Marquardt, N. and Greenberg, S. (2007) Distributed Physical Interfaces with Shared Phidgets. *Proc. 1st Int'l Conf Tangible and Embedded Interaction. (TEI'07)*.
- [6] Mynatt, E.D., Rowan, J., Craighill, S. and Jacobs, A. (2001). Digital family portraits: Providing peace of mind for extended family members. *Proc. ACM Conference on Human Factors in Computing Systems (CHI 2001)*. Seattle, Washington: ACM Press, pp. 333-340.
- [7] Shneiderman, B. *Creativity Support Tools: Accelerating Discovery and Innovation*. *Com. ACM*, 50(12), 20-32.