Scheming Textiles: End-user *Programming* for Wearables

Nwanua Elumeze, Michael Eisenberg
Dept. of Computer Science, University of Colorado at Boulder
{nwanua, duck}@cs.colorado.edu

Abstract

This paper attempts to expand the landscape of wearable computing with button-schemer, an "ambient program reader" that can be used to input program code directly from a computer screen or from specially bar-coded surfaces. The placement of programs for such a device can be made informal, creative, and practically ubiquitous, suggestive of ways to extend the traditional notions that wearables are to be worn, but not programmed away from a desktop computer.

1. Introduction

Over the past decade, human-computer interaction has increasingly focused on wearable and ambient computing[1]. Just as computers need no longer sit on a desktop, programming should take place in a variety of settings and physical spaces as well.

However, the current approac creates wearables that can only be programmed while connected (even if the link is wireless) to a full-blown computer. The whole point of wearables is to have computation interwoven with garments; yet the status quo requires the wearer to return to the desktop in order to customize/change the artifacts' behavior.

Wearable computing suggests a range of exciting things one should be able to do. Given the ability to gather data and new programs as she goes about her day-to-day activities, the wearer can:

• augment or replace running programs "in situ" with programs scattered throughout the environment (e.g. text represented in barcode transcribed on specially encoded stockings, or from a friend's bag)

• opportunistically gather input for running programs: data could be the day's humidity levels, temperature readings, sequences of motion, etc. that can then be played back, augmented, or simply brought back to the larger computational display at home.

This paper introduces a novel computational artifact called button-schemer, whose purpose is to explore and extend the possibilities for programming away from the desktop.

Briefly, button-schemer (see Figure 1) is a small device that can be sewn onto fabric, as an element of a computationally enriched garment or textile. The device is an "ambient program reader" that can read programs directly (optically) from a computer screen (i.e. it does not use a wire- or RF-based connection).

Alternatively, one can read programs into button-schemer by sweeping it physically across appropriately marked materials such as paper or cloth, in the manner of a barcode reader. One might want to think of button-schemer overall, therefore, as a kind of "program-reading button", sewn onto something like a sleeve or bracelet. Once the device has read in a program, that program can be used to activate lights, motors, or buzzers; it may also be employed to read other input besides programs (such as sensor values).

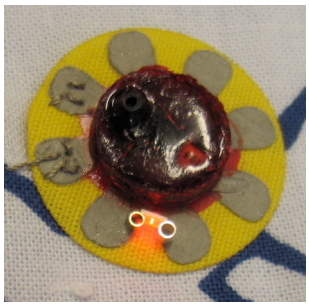

Figure 1. Example instance of a button-schemer, embedded within a protective epoxy case.

The upshot of this design is that one might read in programs from all sorts of serendipitous sources – a friend's laptop, a "program mural" on a wall, a printed-out pattern on stockings – to customize a computational wearable. The placement of programs for such a device can be made informal, creative, and practically ubiquitous.

[1] *We use the term "ambient computing" to describe computation that is worn, carried, or embedded in arbitrary surfaces and artifacts.*

2. A user scenario

Bridgette has a button-schemer-equipped bracelet that includes six multicolor lights and a pushbutton switch. Traditionally, if she wanted the lights to cycle through a list of colors, she would need to write some kind of textual program at a computer, connect, and download the code. Away at a party, she cannot significantly alter her program, or gather new inputs to change the bracelet's behavior.

Instead, because button-schemer can read programs written in barcode form, she pulls out a deck of specially designed cards with barcodes on them and lays out a "make a-named-list" card, followed by several color cards: (e.g. red, purple, green, as shown in Figure 2). She then scans the bracelet over the sequence of cards.

Figure 2. Laying down a list of color cards.

Next, she lays down a LIGHT card, followed by the same LIST card from the previous step, and swipes her bracelet again (Figure 3). The effect here is to tell the LIGHT card to use the named list as the argument to a procedure that runs through a sequence of colors.

Figure 3. The list card activates a light sequence.

Now her bracelet would continuously cycle "red, purple, green, red, purple, green..." If at some point she wanted to change the colors or their order, she could do so by creating another list of colors. Different LIST cards refer to different lists, and she may use them in any combination. She could also include SENSE cards that transition colors only when she presses a switch or the temperature rises above a pre-determined level.

3. In closing

Space precludes us from detailing related work, but the button-schemer builds upon, and has been influenced by Crickets[1]; LilypadArduino[2]; Rototack[3]; Curlybot[4]; Pendantif[5]; AlgoBlock[6]; the Programming by Demonstration community [7,8]; IO/Brush[9]; Timex DataLink[10]; and PICBit[11].

Unlike prior work, this prototype allows a mix of textual programming with physically generated input (e.g. with it, a robot can be programmed to create a spiraling motion just from having been dragged in a short arc across the floor). This blending can mix the best of both worlds in ways unprecedented. For instance, "that wiggly motion" is hard to get right textually, but a physical artifact can record a desired motion. Doing interesting things with the motion is in turn difficult without a textual way of expressing the desired effect, which is why there is a need for a system that can combine these two approaches.

One can produce delightful and non-trivial behavior in physical artifacts with very short programs. The barcode versions of programs can be printed out on paper and pasted on walls, transcribed in clothing, or embedded as notes in a song. These abilities enable us to tastefully extend the traditional landscape of wearable computing into unexpected, informal settings.

This style of programming needs to be explored further because it dovetails nicely with the very notion of wearable computation: programs can exist and be read from "anywhere", and the activity can now be situated in the wearer's physical context.

Acknowledgements

Thanks to Mark Gross and Craft Technology Lab. This research was funded in part by the National Science Foundation (awards EIA-0326054 and REC0125363).

References

[1] F. Martin. Crickets, Tiny Computers for Big Ideas. MIT Media Lab.1998.
[2] L. Buechley, "A Construction Kit for Electronic Textiles. ISWC 2006.
[3] T. Wrensch "The rototack: Designing a computationally enhanced craft item." Proc. DARE 2000, pp 93 - 102.
[4] P. Frei, V. Su, B. Mikhak, and H. Ishii. curlybot: "Designing a new class of computational toys." In CHI Letters (2000), pp 129 - 136.
[5] N. Elumeze and M. Eisenberg. "Smart tiles: Designing interactive "room-sized" artifacts for educational computing." Children, Youth and Environments 2005, 15(1): pp54-66.
[6] H. Suzuki and H. Kato. Algoblock: "A tangible programming language, a tool for collaborative learning." Fourth European Logo Conference, 1993, pp297 - 303.
[7] A. Cypher, editor. *Watch What I Do: Programming by Demonstration.* Cambridge MA: MIT Press, 1993.
[8] H. Lieberman, editor. *Your Wish is My Command: Programming by Example.* Morgan Kaufmann, San Francisco, 2001.
[9] K. Ryokai, et al. "Io brush: Drawing with everyday objects as ink." CHI 2002.
[10] Timex DataLink. http://www.timex.com.
[11] M. Feely. Picbit: A scheme system for the pic microcontroller. In Scheme Workshop 2003.