Computationally-Enhanced Craft Items: Toward "Programmable Parts" for Educational Robotics

Tom Wrensch, Mike Eisenberg, and Glenn Blauvelt
Department of Computer Science
University of Colorado, Boulder

The notion of educational robotics—much like the broader notion of robotics in general—permits a variety of interpretations. As robots differ in their design, so do the educational issues accentuated by these devices. The issues involved in (e.g.) creating a model of a stationary industrial robot might be quite different from those involved in creating a model Mars rover; creations made of Lego bricks raise engineering concerns different from those made of Fischertechnik pieces; the notion of "machine vision" for a soccer-playing robot might be distinct from that for a Braitenberg [1984] vehicle that simply avoids obstacles.

We begin with this observation only because the style of educational robotics described in this paper may, in some quarters, be regarded as at least slightly unorthodox. Over the past several years, we have worked on the design of several *computationally-enhanced craft items*. [Wrensch *et al.* 2000; Blauvelt *et al.* 1999; Wrensch and Eisenberg 1998] These devices are similar in their basic design and geometry to the sorts of broadly useful objects that appear in handicraft projects; but they are also programmable devices, and therefore (we believe) capable of vastly extending the expressive range of students, hobbyists, and amateur scientists and engineers.

Computationally-enhanced craft items are not designed solely for purposes of educational robotics. Nonetheless, this paper will argue that they are well suited to that end, and that the design of such craft objects holds substantial promise for raising fascinating new issues in educational robotics, particularly in areas such as distributed processing and end-user programming.

In the remainder of this paper, we provide a brief description of a recent sample craft item; we discuss our experience in creating and running a course in "embedded educational computing" at the University of Colorado, and how that course influenced the design of our craft items; and we outline the ways in which computationally enhanced craft items could provide intriguing perspectives on future efforts in educational robotics.

Computationally-Enhanced Craft Items: An Illustrative Example

Perhaps the most fruitful way to approach the definition of a "computationally-enhanced craft item" is to begin by listing the sorts of small, ubiquitous, multipurpose objects typical of craft projects—hinges, tacks, tiles, and so forth—and then to imagine the possible novel behaviors of those items when augmented by embedded computation. As an example, this section describes our recent work in creating a *Rototack*—a programmable thumbtack capable of realizing complex patterns of rotation.

The current version of the Rototack is shown at left in Figure 1, and a schematic at right. Basically, the tack is a cylindrical object containing a PIC microcontroller, stepper motor, and batteries; when objects are connected to the motor shaft at the top of the tack, they can be rotated about the axis defined by the tack's spike.

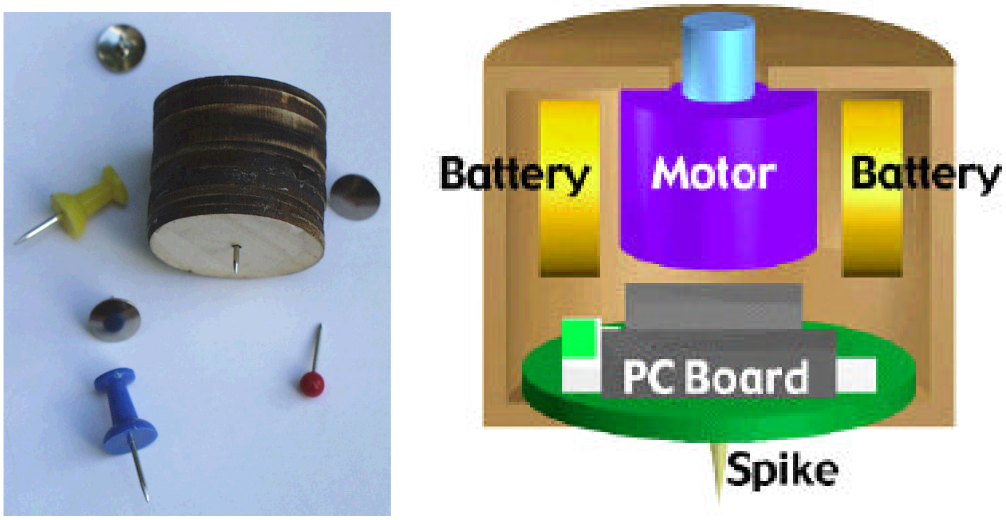

Figure 1. At left: a Rototack and a variety of noncomputational tacks for comparison. At right: a schematic of the Rototack.

The overall scenario by which the tack is employed is as follows. The user first brings up a tack-programming application on her desktop computer, and either writes a new program for the tack or retrieves and edits a pre-existing program. Once the program is to her liking, she then connects a physical Rototack to the computer via an interface port (shown in Figure 2) and downloads the program to the tack. The tack's local computer now contains a runnable version of the desired procedure. (Just to take a specific example: the program might stipulate that the tack execute a series of 60 clockwise turns of 90 degrees, interspersed with pauses of 10 seconds; then follow that by 120 counterclockwise turns of 45 degrees, interspersed with pauses, each of a random amount of time; then repeat the entire process.) When the tack is inserted into a corkboard, the program begins to run; alternatively, the user can choose to run the tack's program only when an external switch on the side of the device is set.

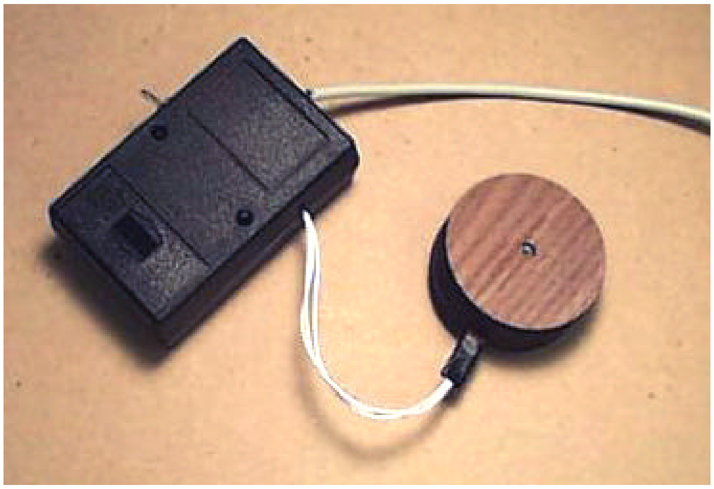

Figure 2. The Rototack connected to its interface port for downloading a newly-written program. At right, out of the photograph, is a desktop computer

Much more detail about the tack, its programming language and environment, and its prospective uses may be found in [Wrensch et al. 2000]. It should also be mentioned that, along similar lines, we have designed prototypes of a programmable hinge (in which opening/closing behavior is computationally driven) and a programmable ceramic tile (in which the programmed behavior primarily involves inter-tile communication). More discussion of these efforts may be found in [Blauvelt et al. 1999] and [Wrensch and Eisenberg 1998], but for our purposes in this paper the essential point is that all share a basic design strategy in which relatively small, simple, inexpensive, and geometrically constrained devices are endowed with a potentially large set of programmed behaviors. Crucially, the ultimate interest of these devices lies not in their stand-alone behavior, but rather in their behavior as parts of larger projects. A Rototack could (for instance) turn a set of gears on a corkboard to power a piece of kinetic artwork; it could be used to generate dynamic optical illusions; it could power a joint of a marionette or simple robot.

In our view, the overall design strategy for computationally-enhanced craft items shows a close affinity with the concerns of educational roboticists. First, we assume that the prospective users of such items are not experienced programmers; like crafters or hobbyists in general, their presumed area of expertise (or study) might as easily be mechanical engineering, architecture, instrumentation design, or sculpture as computer science. Second, programs for computational craft items are likely to be short—no more than a page or so in length (indeed, this is true of all the early sample programs that we have written thus far). This in turn suggests that computational craft items might be appropriate media through which to introduce beginning concepts of programming and engineering. Finally, as in the practice of robotics, the programming environment for computationally-enhanced craft items is distinct from the platform in which programs are run; thus, while the craft items themselves are assumed to be small objects with limited memory and processor speed, the desktop programming environment may be quite elaborate and featureful. As an example of this idea, our current Rototack programming environment includes not only tools for editing and animating sample Rototack programs, but an animation-based program browser shown in Figure 3. Here, running animations of Rototack programs are used as icons through which the user can select a pre-existing program to edit. The purpose of this example is merely to show that (as in robotics), the creation of relatively small or simple programs may be accomplished through a rich programming environment. [Cf. Eisenberg et al. 1999]

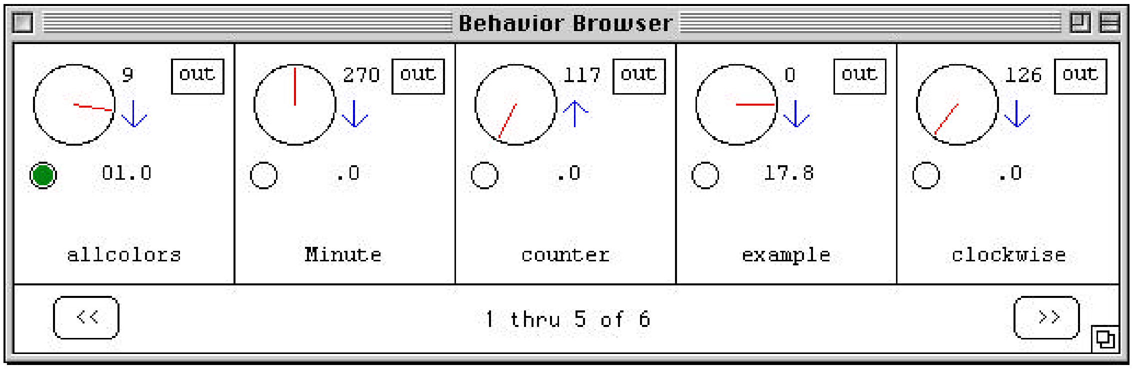

Figure 3. An animation-based behavior browser for the Rototack. Here, a variety of tack behaviors may be viewed in animation. The user selects the desired animation and this selection loads the appropriate program into the system for editing. Note that, in general, such animation-based selection is only a possibility for objects like the Rototack, with highly constrained behaviors; attempting to create a similar system for a general-purpose programming system would be vastly more challenging and, most likely, vastly less usable.

Origins of Computationally-Enhanced Craft Items: a Course in Embedded Educational Computing

It is worth pausing in our discussion at this point to sketch the similarities and contrasts between items such as the Rototack and other educationally-inspired items such as the Lego programmable brick (and its most recent instantiation in the MIT Media Laboratory "cricket") [Resnick et al. 1996; Resnick 1993]. Doing this requires a bit of historical digression. We have on two occasions taught a course entitled "Things that Think" at the University of Colorado; the course, open to both graduates and undergraduates (chiefly in architecture and computer science), focuses on integrating computation into the design of scientific toys, kits, and exhibits. The primary computational tool for the course on both occasions was the cricket, programmed through an extension of Microworlds Logo.[1]

Our experience in the course was too complex and provocative to convey in a paragraph or two; but very briefly, the students' work with the programmable brick was both exhilarating and frustrating. On the one hand, the programmable brick, by its very existence, inspired the students to try a variety of creative projects; but with the interesting exception of those projects targeted toward the relatively narrow domain of Braitenberg-type Lego vehicles, most of the students seemed to struggle with limitations of the brick. Sometimes, it seemed, the brick was just a trifle too bulky for the desired purpose; or too heavy; or its style of inter-object communication (via infrared light) too inflexible; or the types of sensors or actuators associated with it were too limited; or it was difficult to integrate the brick with other electronic devices. In our view, many of these problems stem from an implicit design decision in the Lego brick—that it should serve as a full-purpose computer in miniature. This is an excellent style of design when the brick is intended to serve as a "robotic brain", controlling the behaviors of model artificial-life creatures; in such a case, what the student wants is in fact something like a tiny, but powerful, computer. For many other sorts of projects, however, what is wanted is something at once both more and less exacting: less exacting in terms of memory, processing speed, and computational sophistication, but more exacting in its physical demands of weight, materials, or geometric specificity.

This discussion is emphatically not meant as a negative critique of the cricket; indeed, without this artifact a course such as ours could not have been imagined, let alone implemented. At the same time, however, it was surprising to us how beautifully the cricket fitted into the students' artificial life projects, and how trickily it generally fitted into everything else. Over time, we came to believe that an appropriate computational environment for a course such as ours would include a wide variety of device architectures and geometries; some of these would be close in spirit to the "tiny general-purpose computer" style of the cricket, while others would be more in the spirit of programmable tacks and hinges. In future iterations of the course, we plan to employ prototype craft items as well as the cricket (and perhaps other programmable devices, such as the Mindstorms brick, as well).

Computationally-Enhanced Craft Items and Educational Robotics

In this final section, we return to the question of how computationally-enhanced craft items might play a role in educational robotics. Our belief is that items of this sort can potentially highlight a variety of novel and interesting approaches to robotic construction. One such approach is suggested by the discussion of the previous paragraphs: rather than design a robot to be controlled by a general-purpose computational "brain", we might imagine designing a robot whose intelligence is distributed through a plethora of separately programmable hinges, tacks, fasteners, strings, rods, and so forth. While the individual behaviors of such devices may be simple, their collective behavior could be interestingly complex. (This is an instance, arguably, of de-emphasizing the centralized mindset in robotic design, following the eloquent discussion of Resnick[1994.) A related point is to rethink robotic design as distributed, not only among various computational devices, but between classically computational entities and elements that are closer in spirit to "intelligent materials", such as the shape memory alloy used in the popular Muscle Wire. [Gilbertson 1993]

More generally, we see computationally-enhanced craft items fitting into a portrait of educational robotics that is itself something of a craft discipline. To take one specific dimension of this notion: rather than designing robots from commercial kits of (e.g.) Lego or Meccano pieces (rich as such collections of pieces may be), we prefer a style of design that takes advantage of a wide variety of traditional and high-tech materials (such as fiber optics, fluorescent dyes, temperature-sensitive films, and so forth). While the commercial kits have certain advantages (e.g., in interoperability of parts), these advantages may be offset to some degree by the advent of computer output devices such as inexpensive laser cutters and three-dimensional printers [cf. the discussion in Gershenfeld 1999]; such devices should make it increasingly feasible and affordable for students to design novel or customized pieces on a computer screen, and to (in

[1] To be precise: the second author co-taught the course with Prof. Mark Gross on both occasions. The first and third authors participated as students on the first occasion, and participated both in designing and teaching the course on the second.

effect) print these pieces out for use in construction. Simple but multipurpose items such as the Rototack and programmable hinge should, we believe, work well within such free-style construction schemes. In effect, then, we see computationally-enhanced craft items as one region of the much larger design space in which handicrafts and traditional educational robotics may eventually blend together.

Acknowledgments

We are indebted to the ideas and conversation of Robbie Berg, Ann Eisenberg, Gerhard Fischer, Mark Gross, Fred Martin, Mitchel Resnick, Andee Rubin, and Carol Strohecker, among many others. The work described in this paper has been supported in part by National Science Foundation grants CDA-9616444 and REC-961396. The third author is supported by a gift from the Mitsubishi Electric Research Laboratories (MERL) in Cambridge, Massachusetts. Finally, we would like to thank Apple Computer, Inc. for donating the machines with which this research was conducted.

References

[Blauvelt, Wrensch, and Eisenberg 1999] Blauvelt, G.; Wrensch, T.; and Eisenberg, M. [1999]. Integrating Craft Materials and Computation. In *Proceedings of Creativity and Cognition 3*.

[Braitenberg 1984] Braitenberg, Valentino. *Vehicles*. The MIT Press. Cambridge, Massachusetts.

[Eisenberg, Wrensch, and Blauvelt 1999] Eisenberg, M.; Wrensch, T.; and Blauvelt, G. Geometry-Specific Languages. Colorado Department of Computer Science Technical Report CU-CS-886-99, October.

[Gershenfeld 1999] Gershenfeld, N. *When Things Begin to Think*. New York: Henry Holt.

[Gilbertson 1993] Gilbertson, R. *Muscle Wires Project Book*. Mondo-Tronics.

[Resnick 1994] Resnick, M. *Turtles, Termites, and Traffic Jams*. Cambridge, MA: MIT Press.

[Resnick 1993] Resnick, M. Behavior construction kits. *Communications of the ACM*, 36:7, pp. 64-71.

[Resnick *et al.* 1996] Resnick, M.; Martin, F.; Sargent, R.; and Silverman, B. Programmable Bricks: Toys to Think With. *IBM Systems Journal*, 35:3, pp. 443-452.

[Wrensch, Blauvelt, and Eisenberg 2000] Wrensch, T.; Blauvelt, G.; and Eisenberg, M. The Rototack: Designing a Computationally-Enriched Craft Item. In *Proceedings of DARE (Designing Augmented Reality Environments) 2000*, Elsinore, Denmark, April.

[Wrensch and Eisenberg 1998] Wrensch, T. and Eisenberg, M. The Programmable Hinge: Toward Computationally Enhanced Crafts. *Proceedings of UIST 98,* San Francisco, November, pp. 89-96.