Mechanical Hijacking: How Robots Can Accelerate Ubicomp Deployments

Scott Davidoff
Carnegie Mellon University
research@scottdavidoff.com

Nicolas Villar, Alex S. Taylor, Shahram Izadi
Microsoft Research Cambridge
{ nvillar | ast | shahrami } @microsoft.com

ABSTRACT
The complexities and costs of deploying Ubicomp applications seriously compromise our ability to evaluate such systems in the real world. To simplify Ubicomp deployment we introduce the robotic pseudopod (P.Pod), an actuator that acts on mechanical switches originally designed for human control only. P.Pods enable computational control of devices by hijacking their mechanical switches – a term we refer to as *mechanical hijacking*. P.Pods offer simple, low-cost, non-destructive computational access to installed hardware, enabling functional, real world Ubicomp deployments. In this paper, we illustrate how three P.Pod primitives, built with the Lego MindStorm NXT toolkit, can implement mechanical hijacking, facilitating real world Ubicomp deployments which otherwise require extensive changes to existing hardware or infrastructure. Lastly, we demonstrate the simplicity of P.Pods by observing two middle school classes build working smart home applications in 4 hours.

Author Keywords
Mechanical hijacking, prototyping, evaluation, robotics.

ACM Classification Keywords
H.m Information Systems: Miscellaneous.

General Terms
Design, Economics, Experimentation, Human Factors.

INTRODUCTION

Researchers have posited that domains as varied as automotive [4], the home [10], healthcare [8] and sustainability [1] will benefit from applications of ubiquitous computing (Ubicomp). The complexities and costs of deploying Ubicomp applications in the wild, however, still seriously compromise our ability to explore these domains and ultimately to evaluate these claims [2].

UbiComp '11, September 17– 21, 2011, Beijing, China.

Because deployed Ubicomp systems would interact with social mores [3], it becomes important to examine if benefits *in theory* are achieved *in practice*. As importantly, if Ubicomp applications are to fit into the fabric of our daily lives [5] they will need to *interface* with the hardware infrastructure that already exists in such spaces. These legacy infrastructures, from lighting and heating systems, to bespoke automotive navigation and cooling hardware, to legacy systems in healthcare, and to preexisting appliances in the home, offer no well-defined computational interface.

This paper introduces the term *mechanical hijacking*, an approach to leverage and control legacy hardware systems and infrastructures through mechanical means, instead of through an explicitly defined computational interface. We explore this idea through the design of robotic pseudopods (P.Pods). A P.Pod is an actuator that acts on buttons designed for the human hand. By hijacking the capabilities of existing buttons. P.Pods can offer computational control over any device with a mechanical interface without need to replace or alter existing hardware, Figure 1 shows how P.Pods use the existing buttons to control a thermostat, granting non-invasive computational control of a home's climate control system. A real-world deployment can effectively leverage existing infrastructure.

In this paper, we describe three P.Pod primitives that demonstrate mechanical hijacking, and illustrate how they can simplify application deployment across a number of domains. To demonstrate the simplicity of the P.Pod approach, we invite two middle school classes to each create a functioning "smart home" application. Each class successfully creates one application in just 4 hours. We first turn our attention to P.Pod primitives and their application.

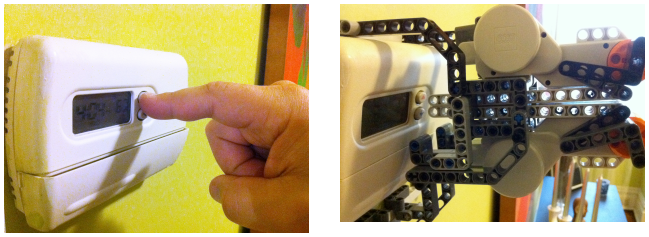

Figure 1. A robot pseudopod (right) mechanically hijacks buttons designed for the human hand, creating non-invasive computational control of a home's existing infrastructure.

APPROACH

Despite a growing belief that understanding the impact of Ubicomp systems requires longitudinal field deployment [2], few systems are actually deployed and rigorously evaluated (*e.g.* [8]). The cost of building custom electronics needed to computationally interface wth installed hardware and infrastructure can deter such deployments[1].

In this section we introduce an alternative approach to ubicomp deployment: *mechanical hijacking*. The alarm clocks, deadbolts, stovetops, light switches, and stereos can already be controlled using the mechanical buttons, knobs and switches built into them for use by human hands. Robot pseudopods (P.Pods) are tiny robots that hijack the mechanical controls on these devices. This provides a mechanical means to control existing hardware, and also a well-defined computational interface by which these robots can be controlled by Ubicomp application developers.

Though only suggestive of a broad vocabulary, we describe three simple P.Pod primitives. Each primitive simulates one capability of the human hand. Though technologically agnostic, we implement P.Pod primitives using the Lego MindStorms NXT[2] toolkit, which includes a wide variety of sensors, a rich mechanical expressivity, and native software interfaces for the Java, RobotC and C# languages.

The piston-like poker (Figure 2, left) simulates finger extension and retraction, hijacking doorbells and on/off switches, buttons that actuate away from the body. The grasping pincher (Figure 2, center) simulates finger flexion and extension, hijacking drill triggers and hole punchers, switches that actuate towards the body, The radial twister (Figure 2, right) operates on sink water flow controls and lighting dimmers, knobs that actuate around rotational axes.

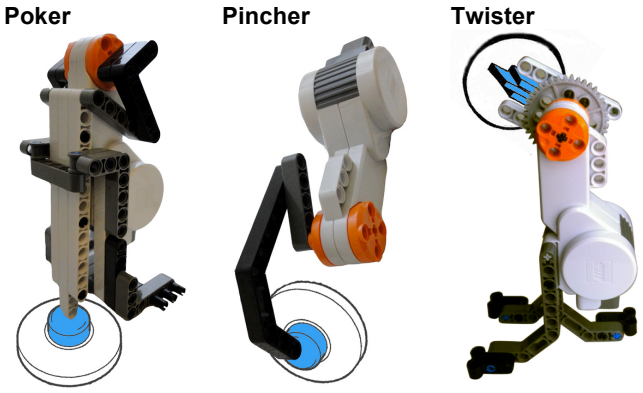

Figure 2. Three P.Pod primitives hijack distinct mechanical controls. The poker (left) controls buttons. The pincher (middle) controls triggers. The twister (right) controls knobs.

Figure 3. A P.Pod twister hijacking an in-car climate control system's mechanical dial. This P.Pod provides a way to deploy a climate control system in an automtive environment, which is normally hostile to experimentation.

IMPLEMENTATION

To illustrate how mechanical hijacking can accelerate and extend Ubicomp deployments, we illustrate the application of P.Pods in two common Ubicomp research scenarios: sustainability, and domestic robotics.

Deploying to Explore Sustainability

A field deployment of the sustainability models envisioned in [6] require a computational interface to an environment's climate control system. Typically, this is achieved in one of two ways. First, an invasive approach wires the thermostat to a microcontroller that provides network interfaces. Second, a less destructive but more costly approach replaces the thermostat with a networked device that communicates with a home control system (*e.g.* Insteon[3]), and offers built-in network interfaces.

Figure 1 shows how two P.Pod pokers can be configured to access the mechanical controls on the front of an installed thermostat without damaging the device. One P.Pod is aligned over each of the two front-facing buttons, allowing the device as a whole to both raise and lower the desired temperature setting. The two P.Pods are attached to the exterior of the thermostat using an enclosing fixture, which also provides resistance against the force of button-pushing. Figure 3 shows an automotive deployment using a single P.Pod twister. In each case, servomotors communicate with control programs over Bluetooth or USB.

The transparency of the mechanical hijacking approach lowers the technical requirements to deploy a functioning system. Even a high-level microcontroller (*e.g.* Arduino[4]) requires an understanding of thermostat controls and feedback loops. P.Pods can mechanically reproduce any interaction that a human can understand. So understanding the buttons on a thermostat is all that is needed to deploy a functioning system.

[1] Hack a day, http://hackaday.com/tag/thermostat/

[2] Lego MindStorms NXT, http://mindstorms.lego.com

[3] Insteon, http://www.insteon.net

[4] Arduino, http://www.arduino.cc/

Figure 4. Three robotic P.Pods together mechanically hijack an i-Robot Roomba. A P.Pod poker uses the on/off button to start or stop the motion of the Roomba. Side-mounted P.Pods pinch the bumpers, giving the Roomba false collision input, causing it to turn away from the side of the pinched bumper.

Deploying to Explore Machine Intelligence

The human relationship with intelligent machines is another area of active Ubicomp research [9]. Efforts are limited because deployments often require custom hardware and are often constrained to fixed locations [7]. Again, we illustrate how P.Pods can *piggyback* on existing hardware to achieve a full range of technical capabilities without their need to be engineered from scratch.

Figure 4 shows how we can configure three P.Pods to mechanically hijack the power and movement of an i-Robot Roomba[5]. A laser-cut grid mounted on the Roomba allows the P.Pods to be placed flexibly on top. A P.Pod poker uses the on/off button to control the movement of the Roomba. We use two P.Pod pinchers to mechanically activate the Roomba's two front bumpers. This signals (falsely) to the Roomba that it has hit an obstacle on that side, causing a change in direction. By pinching the left bumper, the robot goes right. Pinching the right bumper moves the robot left.

Extending the base capabilities of an i-Robot Roomba can allow for much richer interaction with intelligent machines in a prolonged field deployment. While the Roomba has an existing API, this type of low-level, self-contained, and real-time control of the Roomba would be difficult without custom electronics. It also illustrates how other legacy machines could be controlled in this way.

[5] i-Robot, http://www.irobot.com

EVALUATION

To demonstrate the simplicity, flexibility and power of P.Pods, we asked two groups of middle school students to create a deployable Ubicomp application over the course of one week. To control learning costs with the MindStorms toolkit, we recruited two groups that participate in the local First Lego Robotics League. Each group contained ten middle school girls, ages 12-15.

Protocol

Researchers explained how P.Pods work, and demonstrated their use with the hijacked Roomba (see Figure 4).

The participants were instructed to use P.Pods to "control anything automatically" and to focus on things what would "make home more fun." Each of the two groups met three times. At 1.5 hours per meeting, each group had 4 hours in total to complete the task. The groups used the last thirty minutes to present and explain their designs.

The two groups elected to each construct a voice-controlled home entertainment system. When the children cheered in unison, the system was to turn on the television, and play a DVD. To implement this concept, the groups had to develop a way to sense the background noise level in the room, and to control the TV in response to the sensed noise level.

Researchers provided no P.Pod specifications to the girls. Nor did the researchers make the Roomba available to use as a model. Participants moved about the class both collaborating and independently solving problems. The researchers observed, and moderated. They did help the participants think through technical stumbling blocks, but always required the girls to solve technical problems using their own ingenuity.

Figure 5. Group 1 focused on the remote control, using one P.Pod to press "On," another to press "Play". They mounted the devices with masking tape.

FINDINGS

Both groups 1 and 2 successfully created systems that could play a DVD in response to their cheering. After quickly solving the sensing problem, both groups spent most of their time experimenting with P.Pods. Group 1 hijacked the remote controller. Group 2 hijacked the buttons on the front of the TV.

Both groups discovered they could mount the P.Pods using masking tape (see Figure 5). Group 1 initially tried to position a P.Pod poker above the remote control. With the remote's small buttons and dense layout, the poker would often depress multiple buttons. The group adapted the rotational motion of the P.Pod pincher to solve this problem. They used this approach on both the "Power" and "Play" buttons.

With fewer and larger buttons on the front of the TV, Group 2 used the P.Pod poker to more easily gain control of their two buttons.

DISCUSSION

Both our experience, and that of our middle school classes, demonstrates that P.Pods can get prototypes out of the lab and into the field quickly. P.Pods, however, are not the silver bullet of prototyping.

First, mechanical hijacking can only operate on existing controls on off-the-shelf devices. P.Pods cannot support more exploratory applications that operate outside of legacy infrastructures. Also, "deeper" hacks could control devices in more ways than are exposed through physical interfaces.

Second, Lego Mindstorms impose physical limitations on the P.Pod approach. Their relatively large size may require elaborate constructions to actuate controls that lie in close proximity. Additionally, repeated application of servomotors may subject them to drift, requiring periodic adjustment, making long-term deployment difficult.

Even in short-term deployments, P.Pods will block users from interacting with the devices they control. Good P.Pod design can avoid these problems. In one approach, the P.Pod mounting can be designed for portability. The hanging enclosure that mounts the thermostat in Figure 1, for example, allows the P.Pods to be easily removed and then re-mounted. When the environment requires a less flexible enclosure, P.Pods can themselves expose physical or digital interfaces. For physical, semi-direct manipulation, P.Pods can be augmented with external buttons (or other controls). In this case, a user presses a button on the P.Pod, and the P.Pod presses the button on the device. Also a GUI application for mobile phones or tablets can control Lego Mindstorm P.Pods via USB or Bluetooth.

Overall, when researchers are investigating systems that manipulate existing appliances or infrastructures that have mechanical interfaces, P.Pods can accelerate real world deployment by sidestepping the creation of a custom electronic interface.

CONCLUSION

The Ubicomp community continues to move forward under the assumption that systems that sense and take action on behalf of users can provide some kind of improved quality of life. The reality of Ubicomp development is seriously compromised by the real costs and risks to deploy robust, usable systems that fit into the existing hardware and infrastructure ecology of the real world. As long as that gap exists, the proposed benefits of Ubicomp will remain largely theoretical.

The ability to mechanically hijack real world legacy systems and infrastructures using robotic P.Pods presents advances for Ubicomp applications. P.Pods provide the possibility for such applications to now computationally interface with systems such as lights, climate control, or appliances that have previously been off limits to researchers. Our approach allows such legacy systems to be utilized within Ubicomp applications through low-cost and non-destructive means, without the need for any new extensive hardware or custom electronics development.

While the prototyping and creation of research Ubicomp applications represents a considerable investment of time and resources, our hope is that this approach will allow for more rapid prototyping of systems in the wild, and enable new types of applications domains to be explored.

REFERENCES

1. Abrahamse, W. *et al.* (2005). A review of intervention studies aimed at household energy conservation. *J. Env. Psych.*, 25, 273-291.
2. Carter, S., J. *et al.* (2008). Exiting the cleanroom: On Ecological validity and ubiquitous computing. *J. HCI*, 23(1): 47-99.
3. Davidoff, S., *et al.* (2006). Principles of smart home control. *Proc. Ubicomp 2006*, 19-34.
4. Döring. T. *et al.* (2011). Gestural interaction on the steering wheel: reducing the visual demand. *Proc. CHI 2011*, 483-492.
5. Edwards, W.K. & Grinter, R.E. (2001) At home with Ubiquitous computing: Seven challenges, *Proc. Ubicomp 2001*, 256-272.
6. Harle, R. K. & Hopper, A. (2008). The potential for location-aware power management. *Proc. UbiComp 2008*, 302-311.
7. Helmes, J. *et al.* (2011). Rudiments 1, 2 & 3: Design speculations on autonomy. *Proc. TEI 2011*, 145-152.
8. Rowan, J. & Mynatt, E.D. (2005). Digital family portrait field trial: Support for aging in place. *Proc. CHI 2005*, 521-530.
9. Taylor, A. S. (2009). Machine intelligence. *Proc. CHI 2009*, 2109-2118.
10. Taylor, A.S. *et al.* (2007). Homes that make us smart. *Personal & Ubiquitous Computing*, 11(5): 383-393.