A Malleable Control Structure for Softwired User Interfaces

The Authors
The Department
The Institution
Address
e-mail

ABSTRACT

Rather than existing as a computer input device with a rigid shape, a predetermined selection of controls and a fixed layout, a *malleable control structure* is made up of a set of controls that can be freely arranged on control areas. The structure is physically adaptable by users during operation*:* control areas and controls can be introduced, organized and removed to suit interaction requirements and personal preference.

We present an implementation of a malleable control structure called VoodooIO. Our design contributes a novel material – the *network substrate* - that can be used to transform everyday surfaces into control areas, and the concept of implementing basic control units (such as buttons, sliders or dials) as minimal ad hoc network nodes.

VoodooIO does not constitute an application interface in itself. Like any input device, it only becomes concrete as an interface component in the context of a particular application. We introduce the concept of *softwiring* as a collection of techniques and practices that allow users to benefit from malleable control interfaces in a number of concrete scenarios of use.

INTRODUCTION

The evolution from mechanical and electrical hard-wired machines into programmable electronic systems has resulted in products that are progressively more flexible in their functionality. As equipment is increasingly embedded with computing and becomes more ubiquitously networked, there is an immediate increase in the amount of possible uses that a single piece of technology can be used for: A PC, laptop or PDA can simultaneously act as a word processor, a music-mixing studio, a flight simulator or a teleconferencing station.

In the field of human-computer interaction the majority of research has centred on developing interfaces that rely heavily on graphical output and feedback. This is not surprising, since it is the ability of a graphical display to render any number of possible interface configurations (for any number of conceivable application contexts) that makes it such a necessary interaction component. At the same time, the controls for physically manipulating these dynamic graphical representations have evolved into standardized devices, such as keyboards, mice, joysticks or game pads. The current practice is dominated by interfaces where the GUI constantly changes, while the physical control configuration remains practically static.

This paper presents a design for a human-computer control structure that is inherently *malleable*. Rather than existing as a device with a predetermined number and type of controls (e.g. keys, scroll wheels, sliders) in a fixed layout (e.g. QWERTY keyboards), input is specified via a control structure made up of a *control set* that can be freely arranged on *control areas*, which can be shaped and adapted into a custom configuration (c.f. Figure 1). The core characteristics of the design are as follows:

Customizable form: The structure has no prescribed shape. Instead, a flexible material allows a user to specify control areas of various shapes and sizes, which can then be populated with controls.

Variable dimensions of input: Controls can be incrementally added to and removed from control areas, allowing the structure to increase or reduce the dimensions of input as required during use. A newly added control is immediately available for interaction.

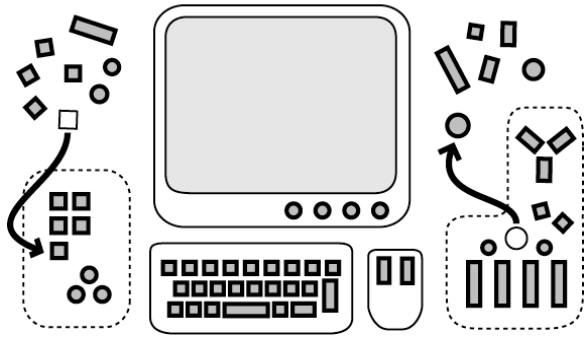

Figure 1. The input device is *malleable*: it can be physically shaped and adapted by the user into a control configuration of their choice. Individual controls can be actively added, removed, and rearranged on user-defined control areas.

Diversity of control types: The control set includes a number of control types with varied operative affordance and mechanical behaviour, such as pushbuttons, linear

sliders, rotary dials, absolute and self-centering joysticks. The control set is extensible, supporting the development and introduction of new control types.

Freeform control arrangement: Controls can be arranged in any layout and orientation on any available control area. When placed on a control area, controls become attached and are held fast in place to maintain the desired configuration. Controls can also be effortlessly (but explicitly) detached from a location for removal or rearrangement.

Physical attachment equals digital connectivity: The act of physically attaching two elements - whether a control to a control area, or one control area to another – also connects them digitally: while connected they are supplied with power to operate, and a network medium to communicate.

Control identification and presence detection: As well as being typed, controls are uniquely identifiable. The act of adding and removing any particular control instance is quickly detected and registered.

With traditional 'rigid' input devices, the user must adapt to using any one device in a particular way and depending on the application context. For example, on a graphical windows interface the effect of pressing a particular keyboard key can result in different behaviour, depending on what application is in focus at the time. It is up to the user to learn, understand and adapt to using the input device based on a changing graphical configuration.

In our design the device is malleable by the user. It acts as a single coherent, extensible system of control, composed and adaptable to the user's individual preference. The advantage is that, by being malleable, the user is able develop control strategies without being constrained to only remapping a fixed set of controls.

Experience in developing and working with a malleable control structure has yielded insights into the new possibilities and challenges that present themselves in developing interfaces that benefit from customizable physical control. We introduce the concept of *softwiring* as a collection of techniques and practices that inform the development of such interfaces. We use the term *softwired interface* to imply a scenario where there is a decoupling of physical-form from system-function, as well as user mechanisms for specifying and managing the ad hoc association between control configuration and application functionality.

This paper contributes a detailed description of a system implementation - called VoodooIO - that enables the deployment of malleable control structures. As a second contribution, it introduces the concept of *softwiring* as a set of techniques that deal with the particular challenges and opportunities in interface design that are elicited by the introduction of malleable control.

VOODOOIO, A MALLEABLE CONTROL STRUCTURE

The concept of a malleable interface device has grown from our experience in developing the VoodooIO (VIO) infrastructure for physically adaptable input [28]. VIO is a set of hardware and software components that enable users to design, deploy and adapt control structures that follow their personal ergonomic preference and control requirements.

Figure 2 shows an example VIO deployment, where a specially developed material is used to specify new control areas in an existing office environment.

Figure 2. The *substrate* material can be used to transform the surface of furniture, equipment and architecture into control areas on which controls can be freely arranged.

We call this material the control *substrate* because of its role as an underlying layer on which individual controls can be placed. In the illustration above a sector of the wall has been covered in material, and further sections of substrate have been affixed to the side of the monitor bezel, and chair armrests. The façade of the telephone unit has been extended with an additional section, as has the desk area around the keyboard - notice how the material has been shaped to fit its curved edge, making full use of the narrow space. Individual sections of deployed substrate are interconnected amongst themselves, and collectively connected to a larger networked system that allows the detection and propagation of interaction events: adding, removing and manipulating controls.

In conjunction with the substrate material, a VIO user is provided with a *control set*. The set is made up of a collection of basic mechanical control devices that can be deployed ad hoc on substrate areas. VoodooIO controls are equipped with small pins at their base to allow them be temporarily fastened to the substrate material. The pins not only hold a control fast in place - allowing it to be operated without unintentionally being moved – but also act as coaxial connectors, connecting the control to a pair of conductive layers embedded inside the substrate. These conductive layers provide connected controls with a power supply as well as a physical network medium (cf. Figure 3).

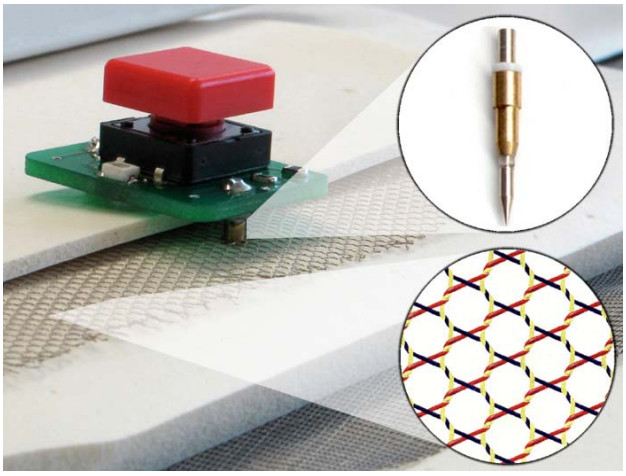

Figure 3. VIO controls are equipped with pins to allow attachment to the substrate material. Upon attachment, the pins act as connectors, connecting the control to conductive layers embedded in the substrate.

VoodooIO Control Set

The VIO control set represents a cross-section of mechanical controls that are likely to be familiar to most users of technology, and includes controls such as buttons, sliders, dials and joysticks (cf. Figure 4). These controls act as input building blocks that can be brought together in an arbitrary way to create custom control environments.

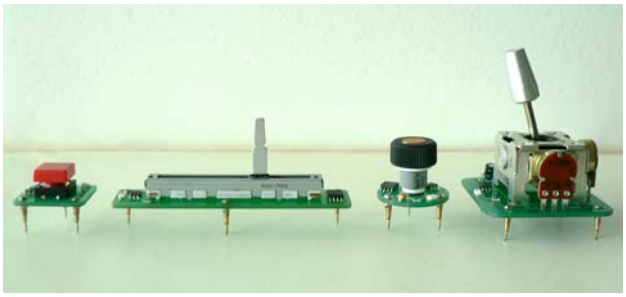

Figure 4. The VIO control set includes familiar controls that are varied in axes of input, operation and behaviour.

The control selection provides a choice of mechanical *operation* (pressing, translating, rotating) and providing binary, linear, or two-dimensional dimensions of input. Control items also vary in *behaviour*: They can be momentary (self-centering), such as pushbuttons or spring-loaded joysticks, as these types of controls are usually good for tasks requiring relative input. These are matched by a subset of persistent (put-and-stay) controls, which are often better for specifying absolute input.

The control set is extensible. A modular design allows new controls to be introduced into the system. We have documented this process as circuit board templates, schematic blueprints and practical design guidelines. The goal is to allow developers with even minimal experience in circuit design to extend the control set with additional haptic transducers, sensors and actuators that may be useful as interface components.

Substrate Material Properties

The composition of the control substrate material has been refined over a number of years to be effective in its dual role as physical support and digital power supply and network medium for controls.

The substrate is a laminate, made up of two layers of conductive fabric separated by silicone foam rubber and bonded together with a pressure-sensitive silicone adhesive. The combined layers result in a sheet of 1.5 cm thickness with a membrane-like quality. Due to manufacturing limitations we have so far been limited to producing sheets of up to 2 m^2, but calculate that sheets can be made considerably larger without degrading in quality as a network medium.

The conductive layers are made of nylon netting, coated in a silver-nickel alloy and weaved in a bobbinet-style tulle. It is highly conductive, making it well suited as an electrical medium. It is also extremely resistant to deformation of its weave-pitch, and allows a consistently reliable connection with the pin-connectors without being damaged by their insertion.

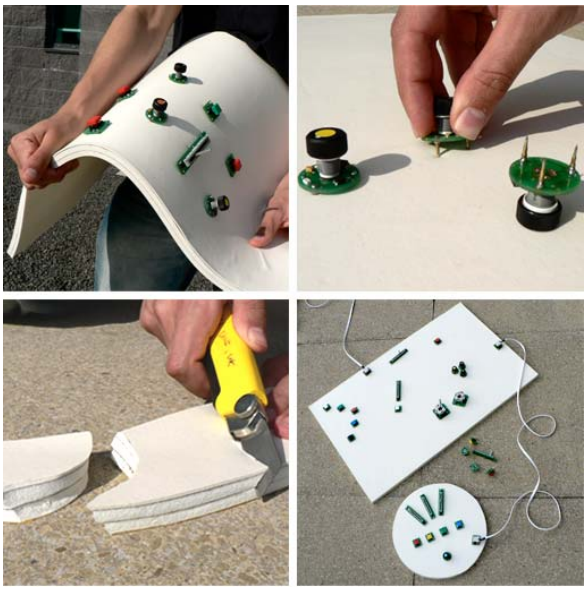

Figure 5. The substrate is flexible, reusable and can be cut and shaped. Sections of substrate can be interconnected to propagate connectivity amongst the internal conductive layers.

The silicone rubber acts as insulator between the conductive layers, and contributes a self-healing property to the surface of the substrate. Controls can be attached and detached from the same location repeatedly without damaging the substrate (cf. Figure 5, top right). The substrate can also be flexed (cf. Figure 5, top left), supporting the creation of non-planar control areas.

The conductive layers run parallel to each other in the planar dimension. This simple structure allows the substrate to be cut into sections of any shape and size (cf. Figure 5, bottom left). Individual sections can then be daisy-chained in any topology using cable interconnects with coaxial-pin

terminations (cf. Figure 5, bottom right). In this way it is possible to spatially distribute control areas with minimal constraints.

Controls as Ad Hoc Network Nodes

The VoodooIO architecture is based on the concept of implementing individual interface controls as self-contained network nodes that can be ad hoc added and removed from an interface network. Each control is implemented as a device which contains: 1) An electromechanical transducer, which can be manually manipulated and electronically sensed or triggered; 2) Digital or analogue I/O modules to measure and drive the state of the transducer; 3) Memory to store the unique ID, type information and device-description; 4) A network interface that allows the I/O and memory to be remotely accessed and controlled, and 5) Connectors that provide a physical link to a network bus (cf. Figure 6).

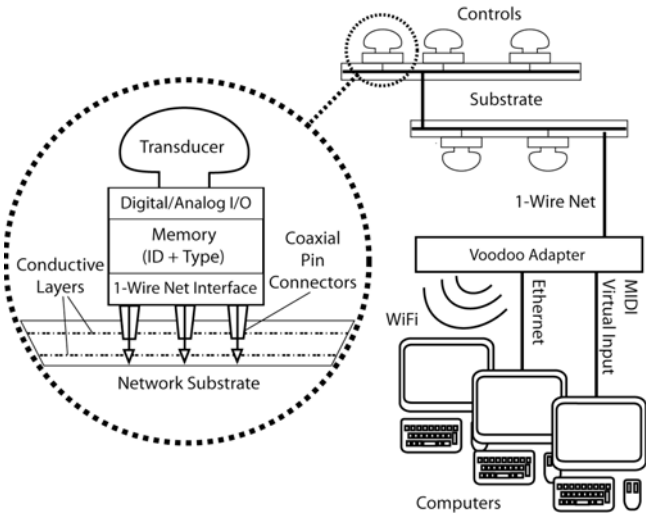

Figure 6. Multiple controls can be attached to a network substrate. Discrete substrate sections can be geographically distributed and networked in any topology. Applications interface with VIO control via Voodoo Adapter devices.

Our design uses the 1-Wire specification as underlying communication protocol with VIO control devices. A 1-Wire network follows a master-slave model supporting two-way communication with multiple slave devices on a serial bus [8]. Only two lines (data and ground) are required for transmission, with the added advantage that 1-Wire devices can be parasitically powered from the data signal. 1-Wire devices come are factory-programmed with a unique network address and are available in a number of packages and models that provide various IC functions. Depending on the requirements of a particular transducer, a VIO control can include any combination of 1-Wire devices, such as the DS2406 (1 Kb memory, 2 x digital I/O pins), DS2408 (8 x digital I/O pins), DS2450 (4 x analogue input pins) or DS2890 (single analogue output).

Voodoo Adapter

The Voodoo Adapter is a device that acts as network master to a 1-Wire bus running over the conductive layers of a substrate network. The adapter constantly scans the network for 1-Wire device presence in order to determine addition or removal of VIO controls. When a new 1-Wire device is found on the bus its internal memory is examined to determine device-description, which contains the ID and type of the VIO control that includes that particular 1-Wire device. In this way the adapter is able to aggregate subsequent device messages from disparate 1-Wire devices into coherent VIO control interaction events without predefined knowledge of the internal control composition.

Whenever a VIO control is manipulated (e.g. a button is pressed), its constituent 1-Wire devices are configured to enter an 'alarming' state to indicate a change of value that needs to be communicated to the adapter. By scanning only for alarming devices - rather than periodically examining the state of all device on the network - the system remains scalable*:* adding additional VIO controls to a substrate network does not affect the latency between a control being manipulated and the system detecting its interaction.

The adapter serves not only as a driver for substrate networks, but also provides application-level interfaces to VIO control. By default, applications are able to receive events from any number of Voodoo adapters via a TCP/IP based protocol, either locally through a loop-back interface, or remotely via Ethernet or WiFi. The adapter can also be extended with additional layers to interfaces via other protocols or application hooks. For example, we have developed a MIDI emulation layer that allows VIO control to be readily used from within a wide variety of existing music generation and production software. A virtual input layer simulates standard keyboard, mouse and joystick input to allow interfacing with existing applications via their graphical interface.

The VoodooIO architecture is designed to be inherently extensible: supporting any number of concurrent applications running on distributed computers, interfacing with multiple of Voodoo adapters, each driving a flexible substrate network topology with an ad hoc collection of controls.

Software Development Tools

Integration of VIO functionality into interactive applications is supported in a number of programming environments. Libraries and code examples have been developed for the C# .NET, Java and Processing and Adobe Flash (Action Script) platforms to provide language-specific classes, such as:

```
VIO_Adapter, VIO_Control,
VIO_DialControl, VIO_ButtonControl …
```

Language-specific methods to class instances, such as:

```
myAdapter.connect(ip, port),
myControl.getID(), myDial.getValue(),
myButton.isPressed() …
```

And some form of event events such as:

```
onControlAdded(addedControl),
onControlRemoved(removedControl),
onControlManipulated(manipulatedControl)
```

The Flash development toolkit is targeted particularly at interface designers that want to quickly sketch and evaluate ideas, and provides additional graphical development support for this purpose.

SOFTWIRING THE INTERFACE

A structure for human-computer control such as VoodooIO does not constitute an application interface in itself. Like an input device, it only becomes concrete as an interface component in the context of a particular application, which likely include additional components of graphical and audible feedback as part of its interface.

We introduce the concept of *softwiring* as the practice of designing an interface that is intentionally designed to support a user working with a malleable control structure and exploiting the benefits of malleability. The term is intended to reflect the loose coupling that can exist between user, control and application functionality. In particular, we are interesting in exploring useful interface design patterns that address the following issues:

Mechanisms for associating control to function: How are mappings between controls and application parameters specified? Are mappings fixed, or specified at run-time by the user? How is control presence (adding and removing controls) dealt with?

Making sense of the interface: How can the user understand and manage a changing control structure? How can they keep track of the meaning of each control if it is not bound to a fixed physical location? How can spatial arrangement, orientation and labelling of controls contribute to making the interface more legible?

As an initial exploration into the concept of softwired user interfaces, we present four different scenarios where VoodooIO has been used to provide malleable physical control of an application. Description of each scenario is necessarily succinct and focuses on the above issues. More details and the particular motivation and implementation behind each application can be found in [4,5, 25,26, 27].

Mixing Desk

Propellerhead Reason is a music production application with a graphical user interface that imitates the control panels of a variety of rack-mounted musical equipment. We explored the way in which a user could use VIO to generate custom mixing desks to operate the software by manipulating physical buttons, sliders and dials, which match the equivalent graphical control representations used in the Reason GUI (cf. Figure 7).

Reason supports remote control of individual interface elements via the MIDI protocol. By using the Voodoo adapter to generate MIDI control data, we were able to directly interface VIO with the software and use its internal control mapping mechanism. During a session a user is able to create a dedicated physical control for any application parameter by selecting a graphical control with the mouse, followed by inserting of a VIO control into the substrate. The effect is one where the graphical control is picked off the screen and placed as a direct physical instance on the control surface.

By arranging and orienting controls on the substrate, a user is able to effortlessly create an operational mixing desk that is specifically tailored to a session or piece of music: the complex graphical interface is reduced to no more and no less than those parameters that are necessary or desirable.

Map Navigator

Google Earth allows a user to browse a collection of satellite imagery spanning the whole surface of the globe, making it possible to point and zoom to view any place on the planet. In this scenario, we project the Google Earth visualization on a large wall-mounted section of substrate. Instead of using a keyboard or mouse to control the application, VIO controls can be attached directly on the projected image as shown in Figure 7.

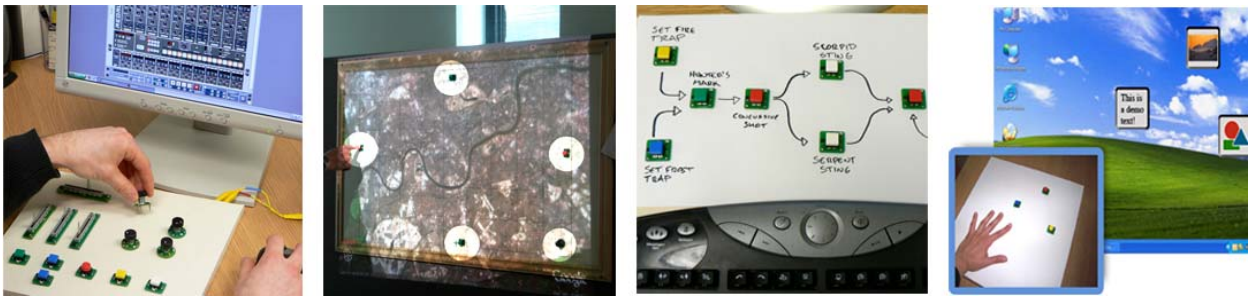

Figure 7. Example softwired user interfaces, from left to right: Mixing Desk, Map Navigator, Gaming Cockpit and Data Clipboard.

Specific controls instances (two rotary dials, two sliders and four push-buttons) are pre-associated with respective Google Earth view functions: zoom, rotate, rate of latitudinal and longitudinal pan and rotate north, south, east and west. In this situation the controls act as tools with fixed mappings, which cannot be changed by the user. Control removal triggers a reset of that tool's parameter (e.g. zoom to whole-earth view). An additional button is also pre-configured to act as a placeholder tool, allowing the user to bookmark a specific location on the map. While its role as placeholders is fixed, the exact function is only specified at the moment that each button is attached, at which time it becomes bound to the view currently being displayed. While attached, pressing the button will recall that particular view. Detaching the button clears the association.

Each control is distinctly labelled with its assigned functionality. The malleability of the interface allows users to choose between different controls for the same task that offer different behaviour, such as using a slider to specify a direction of east-west panning, rather than using two separate buttons that must be kept pressed to continuously pan. Furthermore, users can re-arrange controls to make sense of their function as the application context changes. For example, while having the directional buttons arranged north at the top, south at the bottom, east to the right and west to the left, this only make sense if the globe is oriented with the north pole at the top. If the rotation tool is used and the globe is rotated 180 degrees, the controls would appear to be reversed. The user is able to correct this situation by simply swapping the relative position of the directional buttons without having to remap their functionality.

Gaming Cockpit

World of Warcraft is an online role-playing game where the player is in control of a character in a virtual world. The character gains levels and acquires new skills as it progresses through the gameplay. We wanted to allow the player to use VIO to create a 'gaming cockpit' where they can arrange dedicated controls to their evolving character skills and abilities as they become available.

The game supports the binding of keyboard hotkeys to skills through a key-mapping menu. In this application we used a virtual input emulation layer that generates a keyboard keystroke when a VIO button control is pressed. Whenever a character gains a new ability the player can simply attach a new button to the substrate and create a binding in the same way as if it was a regular keyboard key.

The real benefit to the player comes in the way they can freely arrange the controls. Related controls can be clustered together, and critical actions can be placed close at hand. Although the buttons themselves are not labelled, a piece of paper has been overlaid on the surface of the substrate, allowing freehand labelling of controls as they become attached, as well as additional annotations such as arrows denoting an intended sequence of use for a particular set of controls.

Data Clipboard

We designed the Data Clipboard as an interface device for dedicated control for copying, organizing, transporting and retrieving digital clip data.

A substrate area is designated as a clipboard for a nearby workstation with an otherwise standard user interface. As in existing practice, a user selects an object for copying by selecting it in their GUI. Attaching a VIO button control to the surface at this time has the effect, in a single step, of generating a new clip and of introducing the button as its control. Buttons can be rearranged and clustered into meaningful groups and removed to the effect of also removing the corresponding clipboard item from the computer. Clips can also be exchanged between two systems by moving VIO buttons between surfaces. In order to retrieve the content into an application, the user simply presses the appropriate button control.

This scenario is particularly relevant to the discussion of softwiring in its use of control presence as an interaction modality: Even though each button contains only a binary input control - mapped to the action of pasting a clip - we are able to encapsulate all the other actions necessary to work with clips (copying, organizing, transporting) to the different ways in which the control is made present to the system: on a area, off a area, and on one area after another.

RELATED WORK

Experimental tangible interfaces such as [12,13, 21] have explored the use of graspable objects to directly manipulate digital information. From this body of work, we take as inspiration the importance of the interface being graspable and tactile: providing haptic feedback on manipulation, encouraging multi-handed and multi-user interactions and leveraging on user's spatial reasoning and organizational skills [20].

Interest in creating novel forms of interaction, and particularly new interface devices, has been followed by a number of tools and building-blocks that support interaction designers and interface developers. The Anduino [1] and Gainer [15] kits supply general-purpose I/O boards that abstract from the low-level complexity involved in working with electronics and microcontrollers, and are programmable from graphical and object-oriented development environments. They are widely used by researchers; designers and artists who wish to embed interactivity into their design, but have limited programming or embedded systems development experience. The Phidgets toolkit provides a developer with a collection of controls, sensors and actuators that can be easily connected via USB cables, and also provide high-level development tools in languages such as Visual Basic. [11]. Similarly, the iStuff [2] and CookieFlavours [16] systems provide wireless solutions for integrating controls

and sensors into interface systems. The Calder [17] and d.tools [29] projects focus in supporting product designers in the process of developing rapid prototypes of their ideas. Both systems allow designers to embed working controls in their models at an early stage in the design cycle. By the nature of being malleable, our design is inherently an interface-construction kit – providing control primitives that can be easily assembled into working interfaces, and complete with APIs and development tools in several programming environments. To a developer, the flexibility of the hardware can be appropriated as a way to quickly sketch interface ideas.

In our vision of softwired user interfaces we think of the malleable device as an end-user technology. There is a body of work that concerns itself with the development of input mechanisms that are re-configurable in their composition by their users. Greenberg and Boyle developed a customizable interface where graphical widgets (e.g. buttons, scrollbars) can be associated with Phidgets devices, providing each widget with a dedicated physical handle [10]. The Spinner offers a similar functionality, allowing on-screen elements to be manipulated by physical controls that can be overlaid directly on the display surface [14]. Ballagas et al presented the PatchPanel as a tool to manage the association between disparate input devices and applications – the work envisions a pervasive computing scenario, where new devices and applications are introduced into the system ad hoc [3].

Other work concerns itself not only with supporting freeform composition of the interface, but establishing mechanism by which its shape and spatial layout can be altered. Looking to increase the flexibility of the interface, Brown and Buxton used cardboard cut-outs to partition the surface of touch-sensitive tablets and create virtual controls that could be operated tactually [7]. In the SensOrg work [23], Ungary and Vertegaal demonstrated how it could be beneficial to have the user arrange and lay out control environments in ways that are both cognitively assistive and ergonomically suitable. The Ergodex DX 1 input system [9] is a commercially available product that exhibits some malleable qualities in its physical form, allowing individual keys to be arranged on the surface of a tablet.

Our design provides a selection of mechanical controls from which the user can select the most appropriate for the task at hand. We take inspiration from Buxton's work in [6], where he discusses why this choice is an important concern in the interface: a matter of having the right tool for the job. Our control set consists mainly of basic controls, as can be found on everyday appliances, machines and control panels and which should suggest clear operational values to a user of technology. This decision is reflected in the work carried out by Nakatani and Rohlich on Soft Machines [19] who try to leverage the affordance of mechanical control and machine-like operations with computing and programmable systems.

Finally, a key property of our design for a malleable control structure stems from the fact that physical attachment of controls is resultant in their becoming digitally connected. The inspiration for using coaxial pin connectors to tap into a surface-based network stems from the work carried out in the Pin&Play project [24], as well as other systems which exhibit similar characteristics [18,22].

CONCLUSION

While dynamic in function, standardized input devices are limited in operation. We propose that by providing a malleable control structures, users can become more versatile not only in what they do with their computers, but also in how they choose to carry out this interaction.

To sum up our contribution, we have introduced the concept of a malleable control structure as an alternative solution to establishing human control of computing systems. As proof of concept, we present a description of the VoodooIO system that we have developed to explore the design gap left by other solutions to human-computer interfacing.

Rather than an interface solution in itself, we consider VoodooIO as a tool for end-users to develop solutions to their own interaction challenges, and which are appropriate for their particular needs, personal preference or ergonomic requirements. With this in mind, we have introduced the idea of *softwiring* to encapsulate the different techniques and practices in interface design that allow users to benefit from a malleable control structure as part of their human-computer interface.

REFERENCES

1. Arduino physical computing platform. Project website: http://www.arduino.cc/
2. Ballagas, R., Ringer, M. Stone, M. and Borchers, J. iStuff: a physical user interface toolkit for ubiquitous computing environments. In *Proc. of Human Factors in Computer Systems (CHI'03)*, pp. 537-544, 2003.
3. Ballagas, R., Szybalksi, A., and Fox, A. PatchPanel: Enabling Control-Flow Interoperability in Ubicomp Environments. In *Proc. IEEE Int'l Conference on Pervasive Computing and Communications (PerCom'04).* pp. 241. 2004.
4. Block, F., Villar, N. and Gellersen, H. A Malleable Physical Interface for Copying, Pasting, and Organizing Digital Clips. *In Submission.*
5. Bowers, J. and Villar, N. Creating Ad Hoc Instruments with Pin&Play&Perform. In *Proc.* of *New Interfaces from Musical Expression (NIME'06)* . 2006.
6. Buxton, W. There's More to Interaction than Meets the Eye: Some Issues in Manual Input. In Norman, D. A. and Draper, S.W. (Eds.), *User Centered System Design: New Perspectives on Human-Computer Interaction.* Lawrence Erlbaum Assoc., Hillsdale NJ, pp. 319-339. 1986.

7. Brown, E. Buxton, W. and Murtagh, K. Windows on Tablets as a means of achieving virtual input devices. In *Proc. of Human-Computer Interaction (INTERACT '90)*. Pp. 675-681. 1990.
8. Dallas/Maxim. Overview of 1-Wire Technology and Its Use. http://pdfserv.maxim-ic.com/en/an/AN1796.pdf
9. Ergodex DX1 Input System. Product website: http://www.ergodex.com
10. Greenberg, S. and Boyle, M. Customizable Physical Interfaces for Interacting with Conventional Applications. In *Proc. of User Interface Software and Technology (UIST'02)*. pp. 31-40. 2002.
11. Greenberg, S and Fitchett, C. Phidgets: easy development of physical interfaces through physical widgets. Proc. UIST '01, ACM Press (2001), 209–218.
12. Fitzmaurice, G., Ishii, H. and Buxton, W. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proc. of the Conference on Human Factors in Computer Systems (CHI'95)*. pp. 442-449. 1995.
13. Ishii, H., and B. Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proc. of Human Factors in Computer Systems (CHI'97)*, pp. 234-241, 1997.
14. Kobayashi, S. and Akamatsu, M. Spinner: a simple approach to reconfigurable user interfaces. In *Proc. of New Interfaces for Musical Expression (NIME'05)*. pp. 208-211. 2005.
15. Kobayashi, S., Endo, T., Harada, K. and Oishi, S. GAINER: a reconfigurable I/O module and software libraries for education. In *Proc. New Interfaces for Musical Expression (NIME'06)*. pp. 346-351.
16. Kimura, H. Tokunaga, Tokunaga, E., Okuda, Y. Nakajima, T. CookieFlavours: Easy Buildings Blocks for Wireless Tangible Input. In *Proc. of Human Factors in Computer Systems (CHI'06)*. pp. 965-970. 2006.
17. Lee, J. C., Avrahami, D., Hudson, S. E., Forlizzi, J., Dietz, P. H., and Leigh, D. The calder toolkit: wired and wireless components for rapidly prototyping interactive devices. In *Proc. DIS '04: Designing Interactive Systems*. pp. 167–175. 2004.
18. Lifton, J., Seetharam, D., Broxton, M., and Paradiso, J. A. Pushpin computing system overview: A platform for distributed, embedded, ubiquitous sensor networks. In *Proc. Pervasive '02*. pp. 139-151. 2002.
19. Nakatani, L. H. and Rohrlich, J.A. Soft Machines: A Philosophy of User-Computer Interface Design. In *Proc. of Human Factors in Computer Systems (CHI'83)*. 1983.
20. Patten, J. and Ishii, H. A Comparison of Spatial Organization Strategies in Graphical and Tangible User Interfaces. In *Proc of Designing augmented reality environments (DARE'00)*. pp. 41-50. 2000.
21. Pedersen, E. Sokoler, T. Nelson L. PaperButtons: expanding a tangible user interface. In *Proc. of Designing interactive systems (DIS'00)*. pp. 216-223. 2000.
22. Scott, J., Hoffmann, F., Addlesee, M., Mapp, G., and Hopper, A. Networked surfaces: a new concept in mobile networking. In *Mobile Networks and Applications*. 7(5):353–364. 2002.
23. Ungvary, T. and Vertegaal, R. The SensOrg: A Musical Cyberinstrument with a Cognitive Ergonomical Touch. In *Proc. of IEEE SMC'98 Interntational Conference on Systems, Man and Cybernetics*. 1998.
24. Van Laerhoven, K. Villar, N. Schmidt, A., Gellersen, H., Håkansson, M. and Holmquist L. E. Pin&Play: The Surface as Network Medium. In *IEEE Communications Magazine, April 2003, Vol.41 No.4*. pp. 90-96. 2003.
25. Villar, N., Gilleade K., Raymundy-Ellis, D. and Gellersen, H. The VoodooIO Gaming Kit: A Real-Time Adaptable Gaming Controller. In *Proc. of ACM Advances in Computer Entertainment (ACE '06)*. 2006.
26. Villar, N., and Gellersen, H. The Friday Afternoon Project: A VoodooIO Prototyping Exercise. In *Proc. of Mobile and Embedded Interactive Systems Workshop (MEIS'06)*. 2006.
27. Villar, N., Linsday, and Gellersen, H. A rearrangeable tangible interface for musical composition and performance. In *Proc. of New Interfaces for Musical Expression (NIME'05)*. 2005.
28. VoodooIO Project Website: http://www.voodooio.com
29. Hartmann, B., Klemmer, S. R., Bernstein, M., and Mehta, N. d.tools: Visually Prototyping Physical UIs through Statecharts. In Extended Abstracts of UIST 2005, October 23-26, 2005.