VoodooSketch – Extending Interactive Surfaces with Adaptable Interface Palettes

Florian Block[1], Michael Haller[2], Hans Gellersen[1], Carl Gutwin[3] and Mark Billinghurst[4]
[1]Lancaster University, {block, hwg}@comp.lancs.ac.uk, [2]Upper Austria University of Applied Sciences, haller@fh-hagenberg.at, [3]University of Saskatchewan, gutwin@cs.usask.ca, [4]HitLab NZ, mark.billinghurst@hitlabnz.org

ABSTRACT

Voodoosketch is a system that extends interactive surfaces with physical interface palettes on which users can dynamically deploy controls as shortcut to application functionality. The system provides physical 'plug and play' controls as well as support for sketching of controls, and allows controls to be associated with application functions via handwritten labels. The system uses a special digital pen, which writes 'real' ink on the palettes while functioning as a digital input device on the interactive surfaces. The palettes can be seamlessly integrated into existing applications, be appropriated by the user to suit different input requirements, and support new interaction styles across multiple surfaces, palettes and users.

Author Keywords

Physical Interfaces, Sketched Interfaces, Adaptable Interfaces

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

Interactive surfaces – large areas which work as direct input and output space for applications – are becoming more and more commonplace. With improving display and sensing technology, ordinary surfaces such as tabletops or walls can be augmented and extend the traditional desktop computer beyond a single set of displays and input devices.

On the one hand this creates new possibilities for applications to facilitate the new screen real-estate and its input capabilities. Applications can extend across larger areas and multiple displays. Also, interactive surfaces offer great potential for collaborative scenarios in which multiple users interact with the system at the same time. However, these possibilities also create a series of new challenges for the graphical user-interface:

- *Interface localization*: interface elements can be arranged across large areas or several displays. Moving between the actual activity and graphical interface components can hence be straining and time-consuming. The interface has to address this problem and offer controls which are localized and easily accessible to the user at all times.
- *Collaboration*: scenarios with several interactive screens offer great potential for multi-user collaboration. The interface needs to support appropriate interaction, including multiple users sharing a set of interface components or the handover of an interface from one user to another.
- *Different perspectives*: several users can have different perspectives on the augmented screen and interface elements need to adapt to different views.
- *Adaptability*: users performing different task my have different requirements. The interface should therefore be able to offer customized controls for different users / tasks.

Implementing these features in software and extending the *graphical UI* respectively introduces some complex challenges (such as sensing user location / orientation). In contrast, we want to propose an approach, which extends the graphical user interface by tangible components and supports the mentioned functionality on a *physical* layer.

VoodooSketch is a system which extends an environment of interactive surfaces with customizable interface palettes. The extension provides users with separate palettes as additional interactive surfaces (on top of the existing interactive surface, c.f. Figure 1) for deployment of custom controls (physical as well as sketched controls).

The concept, demonstrated with Adobe Photoshop as the application, is to separate interface and work surface, to provide tools that are custom-assembled for a task at hand, and to support fluid interaction across the physical and digital surfaces for one or multiple users. This is realised with Anoto pen and paper for digital labelling and sketching as well as pen interaction across physical and digital surfaces, and with VoodooIO as physical interface technology.

TEI 2008, February 18–20, 2008, Bonn, Germany.

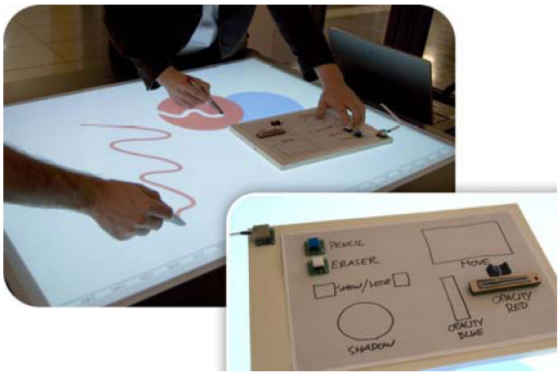

Figure 1: Voodoosketch allows users to bind application functionality to physical or sketched controls on separate interface palettes.

The physical palettes can provide handy access to application functionality and eliminate pointer interaction with the graphical user interface. They also offer free user-customization and can be appropriated to various input requirements, offering not only binary hotkeys, but several types of controls such as sliders, joysticks or 2D-touchpads. Their light-weight physical implementation makes them easy to move and afford seamless localization by the user. Furthermore, palettes can be physically shared by multiple participants in order to support collaborative tasks.

RELATED WORK

The challenges that interactive surfaces impose on the conventional graphical user interface have been systematically investigated within DiamondSpin, an around-the-table interaction toolkit [3]. The authors identified two key challenges for interfaces on interactive surfaces: (1) supply collaborative users with the ability for multiple concurrent interactions with objects and interface elements on the augmented tabletop and (2) provide easy object and interface rotation to suit different user perspectives around the interactive surfaces. The toolkit presented a software approach to extend existing graphical user interfaces. In contrast, VoodooSketch implements the mentioned functionality by extending the graphical interface with physical interface palettes that are separated from the workspace.

VoodooSketch is based on work of the Office of Tomorrow project [1], which extends interactive tabletops with additional physical interface elements such as colour pickers or toolbars. In contrast to these static interface extensions, VoodooSketch offers the dynamic extension and creation of user-customized interface areas. Also, it supports physical plug-in components which support further interaction styles. The support for user-adaptation of physical interface components has been demonstrated by Phidgets [5]. However, in contrast to the loose assembly of wired controls and their on-screen configuration, VoodooSketch combines plug-in interface elements on physical palettes and furthermore supports the live sketching of custom shaped controls and their configuration by handwritten labels on the palette surface. Sketched controls have been formerly introduced by interface design tools such as SILK [2], giving interface designer the possibility to sketch interfaces and program underlying behaviour while maintaining the sketchy look. In contrast, VoodooSketch is targeted at the end-user and allows the creation and configuration of sketched controls during runtime.

DEMO SCENARIO

To demonstrate the VoodooSketch system, we have integrated the support for our interface palettes into Adobe Photoshop, which runs on an augmented tabletop (cf. Figure 1). The user can directly interact with the application and its graphical user interface by using a special digital pen. Alternatively, the user can move several controls from the graphical user interface to a VoodooSketch palette. Having a function in mind (e.g. "Layer Opacity") the user can add an appropriate control to the palette. Users can choose between either physical plug-in components such as buttons or sliders, or to use the input pen to sketch controls onto the palette's surface (e.g. a long rectangle as a slider). For this purpose, the digital pen writes with real ink on the interface palettes In order to bind either physical or sketched components to a function, the user can annotate a new control with a handwritten label (e.g. "opacity"). The new component is then instantly associated with the appropriate function and input from that control is mapped to the appropriate action in Photoshop. Figure 2 shows a sample palette configuration.

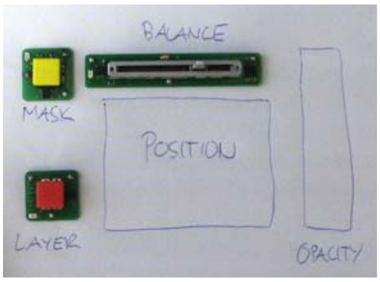

Figure 2: A sample VoodooSketch palette combining physical and sketched components

IMPLEMENTATION

The VoodooSketch system integrates two existing technologies, Anoto and VoodooIO, as input devices. Anoto comprises a unique pattern of very small dots that can be printed on paper of any size, and a digital pen that leaves 'normal' ink on the paper but in addition has a built-in camera to track pen movement in relation to the pattern. From the small view that a pen has at any point in time, the Anoto system can identify the paper on which the pen is used, as well as its position on the paper. This information is provided as input stream via Bluetooth to Voodoosketch for further analysis of pen interactions, such as stroke detection.

VoodooIO [4] is based on a flexible substrate material with embedded conductive layers, and provides a toolkit of physical control devices (buttons, sliders, dials, etc.). The physical controls have coaxial pins for their flexible attachment on the substrate, which acts as power and data bus for attached devices using the 1-wire network protocol. The VoodooIO system continuously monitors the substrate network to detect arrival of new controls (which have a built-in unique ID and type), and to track manipulation of present controls. The system accordingly provides two types of event to the Voodoosketch system: creation of a new control, and user input generated with an existing control.

Figure 3 illustrates how Anoto and VoodooIO are combined in our system at the hardware level. The schematic on the right shows how a palette can be composed of Anoto paper on top of a Voodoo substrate, allowing attachment of VoodooIO controls with pins through the paper to the substrate. Note also that both Anoto paper and VoodooIO substrate are highly customisable. They can be produced in different sizes and cut to shape by their users, even after deployment. It is for instance possible for a user to cut a palette that is already populated with controls in half, at run time, and continue to work with the resulting two palettes without having to reconfigure the controls.

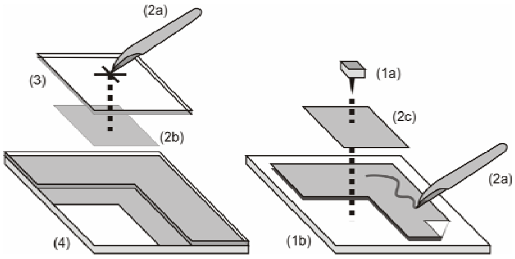

Figure 3: Combination of VoodooIO and Anoto technology: tabletop display (left) and Voodoosketch palette (right). (1) VoodooIO: a. control, b. substrate; (2) Anoto: a. pen, b. and c. paper with separate IDs(3) ink repellent glass; (4) display base.

Voodoosketch is independent of this particular display technology, but their combination facilitates particularly fluid work across display and palettes with the Anoto pen. Note that in this combination the pen is used for sketching and writing with real ink on palettes, and as pointing device on the display which, for this purpose, is protected with an ink-repellent surface.

Adding Controls

At any time, the user can add a control to the interface palette, either by plugging in one of the predefined physical controls of the VoodooIO repository (buttons, sliders, etc.) or by sketching new controls onto the Anoto surface of the palettes. To keep it simple, the user can draw any shape as possible control area, as long as it is fulfils the criteria of a closed shape (drawn in one go and start and end point are close to each other). This definition is easy to understand by both recognition-engine and user and consequently offers a solid detection while obtaining the user's flexibility to create custom-shaped controls. It is important to note that unlike the VoodooIO controls, which inherently have a physical type (button, slider, etc.), the sketched controls have no initial meaning but gain their role only after they have been configured.

Configuring Controls

After a control has been added (either VoodooIO or sketched), it can be associated with a function by annotating it with a handwritten label. The written labels can be clearly distinguished from the closed shapes of the interaction elements and be processed by a standard handwriting recognition engine. The result of the recognition is matched with a list of available application functions (e.g. "opacity" for changing a layer's opacity in Photoshop). This list is provided by an application proxy (see proxy section), which sits in between the VoodooSketch palettes and the application. If the recognized label matches the name of one of the available functions, this function is associated with the annotated control. The proxy also returns the type of input the function requires to the control. For instance, "opacity" would report that it needs a linear input between 1 and 100. For the VoodooIO controls, which have a pre-defined type, this information is used to check if the control can deliver required input and if the test succeeds the control is bound to the function. If the new control is a sketched component, its type is not determined yet. The control is then associated to the appropriate function and the information from the proxy about the input requirements is sent to the sketched control. The sketched control is then programmed to behave appropriately and delivers the required input for its associated function.

Using Controls

Once a control is labelled it becomes instantly operational. The VoodooIO components can be simply used by hand, e.g. a button can be pressed, a slider moved etc. The sketched controls need to be operated using the pen similar to a graphical control being used with the mouse. According to their associated function, sketched controls can behave differently. For instance, if a rectangle is associated to a binary action such as "Save File", the user can simply tap the control with the pen to trigger the event. If the same rectangular control is associated to a function which requires scalar input (e.g. for opacity), the user can slide the pen along the longer axis of the shape and a scalar value is sent to the opacity of the active layer in Photoshop.

Altering the Palettes

Once a particular palette setup is no longer needed, its components can be recycled. The physical components are simply unplugged and ready for re-assignment. Due to the nature of the integration of VoodooIO and the ink of the Anoto Pen the paper-surface gets permanently altered during usage. The VoodooIO components leave holes where their pin adaptors penetrated the paper and the Anoto pen writes with real ink on the paper. However, the sketched controls can be either kept for later use (e.g. controls can be cut out and be re-used) or be thrown away. The A4 Anoto paper sheets which are deployed to cover the palettes can be easily and cheaply reproduced using a standard laser printer.

APPLICATION PROXIES

For the VoodooSketch system to work with an application a developer has to provide an application proxy. The proxy is programmed with the application's set of possible functions in a one-time process. The functions are also assigned memorisable keywords which are the base for the labelling mechanisms. For each function the proxy knows one or multiple keywords, its input requirements (e.g. binary, scalar, 2d vector, angle etc.) and how the input is programmatically mapped to the appropriate function.

The underlying VoodooSketch subsystem can connect to the application proxy and prompt its function description. Once controls have been successfully created and labelled, the VoodooIO Subsystem sends appropriate input from the palettes alongside their associated function to the application proxy. The proxy can then map the appropriate input to the application function. The VoodooSketch system encapsulates input from different VoodooIO controls and sketched components into abstract input events, which the proxy can process. This leaves the user free choice over which components to use for which task. For instance, in order to set the opacity for a layer in Photoshop, the user can either choose a physical dial, a slider or a sketched dial or slider. All types would deliver the same scalar output which the proxy can map to the appropriate value.

It is important to note that the implementation and configuration of the proxy, as well as the assignment of keywords to functions has to be done by the application developer in a one-time process. The assignment and re-assignment of controls to functions at runtime is then done dynamically by the user.

DISCUSSION

We believe that large interactive displays will become more and more commonplace and that applications and scenarios will arise that take advantage of several of those displays to create rich, integrated environments for collaborative tasks. Within this context, VoodooSketch offers a physical interface extension which *physically* enables many new interactive qualities. Central to the design is the separation of the interface palettes from the actual work-surfaces and the integration of two complementary technologies (VoodooIO + Anoto). The interface palettes are light-weight and mobile and can be carried around by the user (in a minimal configuration, the palette only consists out of a sheet of paper, offering full wireless support via the Anoto digital pen). This enables the user to constantly carry the controls for the required task at hand, for instance when moving around a large screen or switching in between screens.

VoodooSketch supports multi-user collaboration on several layers. Both integrated technologies (VoodooIO + Anoto) can process multiple concurrent interactions (VoodooIO via one shared network, Anoto pens via parallel Bluetooth-streams). Also, the interface palettes inherently eliminate the problem of interface sharing and different viewpoints as the palette's orientation is naturally adjusted when being used by different users (similar to handing over documents or clip-boards).

The system supports full remote operation. Creating (introducing controls) configuration (labelling) and usage (manipulating controls) takes place on the local palette and does not require any on-screen operation. Additionally, the two different technologies (VoodooIO + Anoto) offer several distinct advantages over a graphical solution. VoodooIO components offer tactile feedback which can be beneficial to support "blind" operation, e.g. in tasks which require constant screen focus. Paper palettes enable novel practises, such as splitting and sharing a palettes (tearing). Additionally, both VoodooIO-substrate and Anoto paper are cheap and easy to extend, and the system can seamlessly be extended with additional palettes.

The physicality of our approach eliminates several complex challenges that a graphical, "non-pysical" solution would have to overcome (sensing user-location / orientation, multi-touch technology, off-screen interaction in between screens, extensibility etc.). However, we are aware that a graphical solution would be more dynamic than our physical palettes. VoodooSketch tries to compensate this by offering easy end-user configurability, allowing the system to benefit from its physical characteristics while maintaining a high degree of flexibility. Additionally, this enables the user to incorporate personal preference, creating an interface custom-tailored to the user's requirements.

REFERENCES

1. M. Haller, D. Leithinger, J. Leitner, T. Seifried, P. Brandl, J. Zauner, M. Billinghurst. The Shared Design Space, In *ACM SIGGRAPH '06*, ETech.
2. J. A. Landay. Silk: Sketching Interfaces Like Krazy. In *Proc. of CHI '96.*
3. C. Shen, F. D. Vernier, C. Forlines and M. Ringel. DiamondSpin: An Extensible Toolkit for Around-the-Table Interaction. In *Proc of CHI '04.*
4. N. Villar and H. Gellersen. A Malleable Control Structure for Softwired User Interfaces. In Proc. of TEI '07.
5. Greenberg, S. and Fitchett, C. Phidgets: Easy Development of Physical Interfaces Through Physical Widgets. *Proc. UIST 2001*, 209–218.