VersaPen: Exploring Multimodal Interactions with a Programmable Modular Pen

Marc Teyssier
LTCI, CNRS, Télécom ParisTech
Université Paris-Saclay
Paris, France
marc.teyssier@telecom-paristech.fr

Gilles Bailly
Sorbonne Universités, UPMC
Univ Paris 06, CNRS, ISIR
Paris, France
gilles.bailly@upmc.fr

Éric Lecolinet
LTCI, CNRS, Télécom ParisTech
Université Paris-Saclay
Paris, France
eric.lecolinet@telecom-paristech.fr

CHI'17 Extended Abstracts, May 06-11, 2017, Denver, CO, USA
ACM 978-1-4503-4656-6/17/05.
http://dx.doi.org/10.1145/3027063.3052964

Abstract
We introduce and demonstrate VersaPen, a modular pen for expanding input capabilities. Users can create their own digital pens by stacking different input/output modules that define both the look and feel of the customized device. VersaPen investigate the benefits of *adaptable devices* and enriches interaction by providing multimodal capabilities, allows in-place interaction, it reduces hand movements and avoids cluttering the interface with menus and palettes. The device integrates seamlessly thanks to a visual programming interface, allowing end users to connect input and output sources in other existing software. We present various applications to demonstrate the power of VersaPen and how it enables new interaction techniques.

Author Keywords
Pen input; Multimodal interaction; Modular input

ACM Classification Keywords
H.5.2 [Information interfaces and presentation]: User Interfaces

Introduction
User customization is broadly used in off-the-shelf software. Users can choose the position and the content of palettes or change the correspondence between hotkeys and commands. In contrast, physical devices such as keyboards,

Figure 1: Set of modules composing the pen, including sensors and displays.

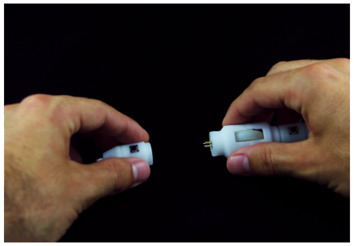

Figure 2: A user staking input modules to create a customized digital pen.

mice or stylus remain mostly static. They have few input and output capabilities that users can seldom customize for adapting these devices to their tasks and applications.

We demonstrate VersaPen (Figure 3), a highly customizable device in terms of input and output capabilities. VersaPen relies on a set of versatile modules including sensors or displays (Figure 1) that can be easily combined by end users by stacking them together while preserving the form factor of a standard pen (Figure 2).

VersaPen offers multiple advantages: Users creates instruments adapted to their tasks and applications by physically customizing a single device.By attaching controls to the physical device, VersaPen reduces hand movements and cursor round trips between menus, palettes and the objects of interest. It also favors multimodal interaction: several sensors close to each other facilitate their use in parallel.

We illustrate the adaptability of VersaPen with several interaction techniques relying on multimodal interaction, in-place interaction and devices augmentation. Our primary contributions are the demonstration of an adaptable and modular interaction device and the presentation of a set of interaction techniques, which aim to bring further discussion about the opportunities raised by VersaPen.

Related Work

Pen augmentation has been proposed to alleviate the problem of mode switching, such as switching between drawing, annotating and issuing commands, or more generally for selecting modes [1]. To achieve this, various input modalities have been considered [2]. Studies also considered different output modalities, such as visual or haptic feedback [3]. VersaPen not only embeds many of these I/O modalities, but also allows users to rapidly prototype and customize devices by combining them.

Figure 3: VersaPen has the same form factor as traditional pens.

Tangible User Interfaces (TUIs) aim at the customization of physical interfaces. Toolkits have been proposed to quickly prototype physical interfaces [4]. However, the target users are developers or product designers rather than the end users who will manipulate these devices. In contrast, MagGetz [5] proposes a set of user-customizable physical controls but their form factor (size and shape) is not appropriate for pen-based interaction.

Adaptable mechanisms such as themes, skins, or styles for customizing user experience (interface, inputs) have been provided in various applications as software features. Other systems have also been proposed for creating more adaptable interfaces in research [6]. These works focus on software and provide a way of adapting input interaction rather than device customization.

Some modular devices are now commercially available such as modular keyboards (e.g *Azio Levetron Keyboard*) or the Google Ara smartphone. While users can modify the shape of these devices, the interaction techniques remain essentially the same. The commutable device proposed in

[7] shares some characteristics with VersaPen. However it aims to be a tool for one specific task and does not share the same versatility and handleable form factor.

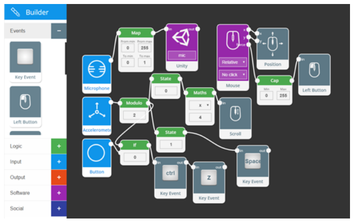

Figure 4: VersaPen flow-based programming interface.

Implementation

Our device consists in a series of modules that can be stacked to build an augmented pen. Once connected to a computer, the user can map the sensors and actuators of the pen to existing applications.

A module consists of two nested 3D-printed cylinders. The inner cylinder contains the electronic parts and fits into the outer shell to form the final module (*18mm*). We built a set of 15 modules, including various input modules such as a buttons, a mouse wheel, an accelerometer, a touch sensor and a microphone. Output modules provide visual feedback, through RGB LED modules, or haptic feedback via a vibro-motor. The tip of the pen is conductive to support compatibility with a capacitive screen and is sensitive to pressure. The form factor results from a compromise to achieve sufficient rigidity while enabling a comfortable grasp. Each module encloses electronic parts.

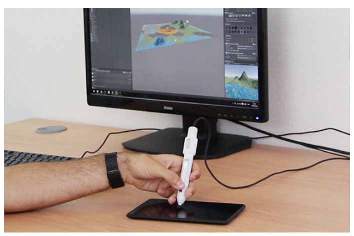

Figure 5: Different input methods are combined to improve workflow.

A microcontroller small enough (< *9.5mm*) to fit inside the pen-sized container, manages the embedded sensor/actuator and handles the communication between modules. Data is acquired locally and differentiated by the IDs of the attached modules and sensors. These IDs allow sending specific instructions to a given output module. Messages from the entire stack are passed via a shared bus to an Arduino controller connected to the computer. A server translates incoming messages and streams them via TCP to the software managing events.

Figure 6: VersaPen used as a game controller.

We implemented a simple flow-based programming interface in order to let end users to connect the various input and output sources to computer applications through visual programming (Figure 4). Users can experience and explore the demonstration as the customization process is compatible with existing applications.

Applications and demonstration

To showcase the possibilities of VersaPen, we highlight its main capabilities and illustrate how it allows creating new tools adapted for specific needs and applications.

In-place interaction

Selecting items on menus, toolbar or palettes require cursor round-trips between the objects of interests (located on the center of the screen) and graphical widgets. In contrast, VersaPen favors "in-place" tool switching. Users can rapidly change tools without moving the pen from the object of interest. We used this ability to enhance a 3D world sculpting application (*Unity3D*, Figure 5). The pressure and the slider modules are respectively used to control the topology of the 3D object and the orientation of the camera while the wheel module serves to access the texturing functions. The tangible controls can easily be reached by the user.

Multimodal & parallel interaction

Users can control multiple *continuous* inputs at the same time using VersaPen. For instance, they can simultaneously control the size (pressure module) and the color (touch module) of a digital brush while drawing (moving the pen) with a single hand. One module allows users to interact with both hands. It consists of two components, one attached to the pen and another one held in the other hand, which is connected to the pen by a flexible wire. This allows manipulating auxiliary controls with the non-dominant hand, as for instance the brush color while the dominant hand controls the brush position and pressure. Adding a vibro-motor and a joystick module transforms VersaPen into a game controller (Figure 6).

Space-efficient interaction
Editing applications typically offer many commands that are organized in palettes and toolbars that tend to hide a large part of the workspace. VersaPen solves this problem by letting users assign a subset of useful functionalities to appropriate VersaPen modules. We used this capability in a drawing application featuring a novice and expert mode to save screen real-estate. In novice mode, turning the wheel of the pen temporary displays a palette and the user can navigate between commands by turning the wheel. In expert mode, the same action is performed without waiting for the palette to be displayed on the screen. Moreover, The haptic and LED modules can serve to provide feedback about the current tool.

Figure 7: Additional information is displayed on the LED module.

Shortcuts and automatic configuration
VersaPen buttons can serve to create shortcuts for activating frequent commands. For instance, one button can change the order of the overlapping windows on the screen. When the user toggles this button, the window under the frontmost window comes to the front (and so on by clicking again). This allows to perform drag and drop operations between windows without releasing the pen. More generally, users can set up favorite shortcuts and link them to buttons or other elements of the pen, with different settings for each application, to improve interaction efficiency.

Figure 8: Modules can expand input capabilities of existing devices (here a keyboard).

Device augmentation
VersaPen can also serve as an auxiliary device of an existing device or application to provide additional feedback. For instance, a 2D data visualization application can take advantage of the LED module to convey additional information (Figure 7). VersaPen can also extend the capabilities of another device as shon in Figure 8 where VersaPen is used to extend a keyboard. The slider module of the pen allows to scroll content without moving the hands from the keyboard.

Discussion
VersaPen is a step towards modular and multimodal devices. It enables exploring various interactions by allowing to easily add/remove modules and assemble them in different ways. We expect this capability to inspire novel usages of pen interaction and leverage multi-modal interaction.

REFERENCES
1. Yang Li, Ken Hinckley, Zhiwei Guan, and James A Landay. Experimental analysis of mode switching techniques in pen-based user interfaces. In *CHI*, 2005.

2. Feng Tian, Xiang Ao, Hongan Wang, Vidya Setlur, and Guozhong Dai. The tilt cursor: Enhancing stimulus-response compatibility by providing 3d orientation cue of pen. CHI, 2007.

3. Johnny C Lee, Paul H Dietz, Darren Leigh, William S Yerazunis, and Scott E Hudson. Haptic pen: a tactile feedback stylus for touch screens. In *UIST*. ACM, 2004.

4. Saul Greenberg and Chester Fitchett. Phidgets: easy development of physical interfaces through physical widgets. In *UIST '01*. ACM, 2001.

5. Sungjae Hwang, Myungwook Ahn, and Kwang-yun Wohn. Maggetz: Customizable passive tangible controllers on and around conventional mobile devices. UIST '13, New York, NY, USA, 2013. ACM.

6. Wolfgang Stuerzlinger, Olivier Chapuis, Dusty Phillips, and Nicolas Roussel. User interface façades: towards fully adaptable user interfaces. ACM, 2006.

7. Alec Jacobson, Daniele Panozzo, Oliver Glauser, Cédric Pradalier, Otmar Hilliges, and Olga Sorkine-Hornung. Tangible and modular input device for character articulation. *ACM TOG '14*, 2014.