VersaPen: An Adaptable, Modular and Multimodal I/O Pen
Marc Teyssier, Gilles Bailly, Éric Lecolinet

To cite this version:
hal-01521565

HAL Id: hal-01521565
https://hal.science/hal-01521565
Submitted on 22 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
VersaPen: An Adaptable, Modular and Multimodal I/O 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

Abstract
While software often allows user customization, most physical devices remain mainly static. We introduce VersaPen, an adaptable, multimodal, hot-pluggable pen for expanding input capabilities. Users can create their own pens by stacking different input/output modules that define both the look and feel of the customized device. VersaPen offers multiple advantages. Allowing in-place interaction, it reduces hand movements and avoids cluttering the interface with menus and palettes. It also enriches interaction by providing multimodal capabilities, as well as a mean to encapsulate virtual data into physical modules which can be shared by users to foster collaboration. We present various applications to demonstrate how VersaPen enables new interaction techniques.

Author Keywords
Pen-based interaction, Modular interaction, Adaptable device

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

In this paper, we investigate the benefits of adaptable devices, which are highly customizable in terms of input and output capabilities. We instantiate our approach in the context of pen-based interaction. While stylus and digital pens are appreciated for their precision and their form factor, which leverage the dexterity of the human hands [10], their input/output (I/O) capabilities remain limited.

We propose VersaPen (Figure 1), an adaptable device which leverages pen-based interaction. VersaPen relies on a set of versatile input/output modules including sensors and displays that can be easily combined by end users by stacking them together while preserving the form factor of a standard pen.

Our primary contributions are (1) the presentation of an open-source [18] adaptable and modular interaction device; (2) its implementation offering expressiveness while preserving the affordance of a pen (3) a set of interaction techniques, samples of opportunities raised by VersaPen.

**Related work**

**Pen-based interaction.** Pen augmentation has been proposed in several studies to alleviate the problem of mode switching, for instance for switching between drawing, annotating and issuing commands, or more generally for selecting modes [15, 26]. To achieve this, various input modalities have been considered such as tilting, rolling [26], shaking [24] the device, as well as pressure [19], grasp [21], near-surface interaction [17], speech recognition (LiveScribe) or just using an additional scroll wheel (Wacom Air Brush).

Previous studies also considered different output modalities, such as visual feedback (using LEDs [16], embedded screens or projectors [22]), audio feedback or haptic feedback [14]. These technologies propose new pen-based interactions, but each one requires a specific device and works with prototype applications. VersaPen embeds many of these I/O modalities and allows users to rapidly prototype and customize devices by combining them. It enables exploring different form factors by adding or removing modules or by assembling them differently.

Several studies recently combined pen and touch interaction [1, 7] to leverage the benefits of two-handed interaction. VersaPen also supports two-handed interaction. Not only users can interact with the non-dominant hand on a multi-touch surface, but the pen can be extended so that modules can serve as remote auxiliary controls with the other hand.

**Adaptable interaction.** Applications, video games, graphical toolkits provide various mechanisms such as themes, skins, or styles for customizing user experience (interface, inputs). Other systems have also been proposed for creating more adaptable interfaces in research [23, 25, 3], but these studies focus on software rather than device customization. The Icon toolkit [2] provides a way of adapting input interaction for multimodal interactions, but does not allow customizing each device individually.

**Modular tangible interfaces.** Some modular devices are now commercially available such as modular keyboards (e.g Azio Levetron Keyboard [11]) or the Google Ara smartphone. While users can modify their shape, the interaction techniques remain essentially the same. Tangible User Interfaces (TUIs) also aim at the customization of physical
interfaces. For instance, Phidgets [4], d.tools [6] or Calder [13] provide toolkits including a set of controls, sensors and actuators allowing quick prototyping of physical interfaces. However, the target users of these systems are developers or product designers rather than the end users who will manipulate these devices. In contrast, MagGetz [8] or WoodooIO [28] propose a set of user-customizable physical controls but their form factor is not appropriate for pen-based interaction. The tangible device proposed in [9] shares some characteristics with VersaPen such as modularity, interchangeability and the capacity to plug-in parts. However this is a tool for one specific task and doesn’t share versatility features and handleable form factor.

VersaPen
VersaPen enhances pen-based interaction by alleviating the problem of mode switching between for instance drawing tools or controls such annotations. VersaPen is a modular, adaptable, hot-pluggable pen allowing users to customize their own pen on-the-go while preserving the shape of a regular pen. The users can map each input/output module of the pen to a control or mode from their favorite applications providing multiple shortcuts at the finger tip.

VersaPen offers multiple advantages. First, users can create their own instruments adapted to their tasks and applications by physically customizing a unique device. VersaPen also reduces hand movements and cursor round trips between menus, toolbars, palettes and the objects of interest because the controls are attached to the physical device. Placing several sensors close to each other makes it easier to use them in parallel and favors multimodal interaction. Finally, VersaPen provides a mean to encapsulate data, software settings or behaviors in physical modules. These physical containers can be shared among users and foster collaboration.

Figure 2: List of available modules. It includes input and output modules and extender modules.

Figure 3: VersaPen module Design. A module is composed of a) an inner cylinder with embedded electronics which can be rotated, a$_1$) a microcontroller, a$_2$) a sensor, a$_3$) male-female connectors. This element fits in a b) outer shell.

The core components of VersaPen are its modules. A module (Figure 3) is a cylinder containing a microcontroller and a sensor or a display, that can be added and removed as needed.

Implementation
Form Factor. A module contains two 3D-printed cylinders: an inner tube (Figure 3 a) with a diameter of 14mm and an outer shell (Figure 3 b) of 18mm. The inner cylinder contains the electronic parts and fits into the outer shell to form the final module. This design allows the user to adjust the orientation of a sensor or a display by rotating the module up to ± 60°. The height of each module depends on the embedded electronics, with a minimum of 22mm. The form factor results from a compromise to achieve sufficient rigidity avoiding deformations while being small enough to enable a comfortable grasp, as with regular pens (available digital pens diameter varies between 19mm to 12mm).
Modules Capabilities. We built a set of 15 modules (Figure 2), which could easily be extended. This set includes various input modules such as a button, a mouse wheel, an accelerometer, a touch sensor and a microphone. The tip of the pen is a conductive module compatible with a capacitive screen and is sensitive to pressure. Output modules provide visual feedback, through RGB LED modules, or haptic feedback, via a module including a vibro-motor. These modules can provide information about the current modes or settings, or for issuing various notifications. VersaPen also provides modules without I/O capabilities to encapsulate and share data (software behaviors, predefined settings, etc.). These modules do not include a memory chip (size constraint) but have a static unique ID (stored in 2 bytes) to make a link with data stored on a remote server.

In order to improve ergonomics, empty modules with different heights (which only contains wires) serve to precisely position the other modules when assembling the pen. One of them is divided in two parts connected with a 40cm long flexible wire. This module allows two-handed interaction: the mobile part controls the behavior of the pen with the non-dominant hand.

Embedded Electronics. Each module encloses an Atmel Attiny85 8 bit microcontroller (Figure 3 $a_1$) which manages the embedded sensor/actuator (Figure 3 $a_2$) and handles the communication between modules. This controller is small enough (< 9.5mm) to fit in the pen-sized container. Modules are cost-effective as their total components cost about $4.

Communication and Data Processing. Messages from the entire stack are passed via a shared bus to an Arduino controller connected to the computer. Each module acquires sensor data locally. The data is transmitted to the top neighbor module and so on until it reaches the last module. Modules are connected by male-female connectors (Figure 3 $a_3$) with four pins (two serving for power and two for data transmission) that also reinforce the rigidity of the pen. Data are differentiated by the ID of the module involved and the type of sensor attached. These IDs also makes possible to send specific instructions to a given output module. The computer receives the data at a rate of 10ms, which allows fluid interaction with the main application. A Nodejs server translates incoming messages and streams them via TCP to the software managing events.

Software implementation. A flow-based interface allows end users to connect the various input and output sources to computer applications through visual programming (Figure 4). This software, built in JavaScript, simulates system mouse and keyboard events to trigger commands in existing applications. For more advanced interaction with applications we implemented an OSC bridge, a standard which is used for add-ons in many off-the-shelf programs. When the software recognizes a specific pen configuration that has already been programmed, parameters are restored and the user can resume his work immediately.

Interaction techniques and Applications
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 requires cursor round-trips between the objects of interest located on the center of the screen and graphical widgets, usually in the periphery of the screen. In contrast, VersaPen favors "in-place" tool switching [5]. 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 (Fig-
Figure 5: VersaPen in use with a 3D sculpting tool (Unity3D). Different input methods are combined to improve workflow and adapt input methods for specific tasks.

Figure 6: VersaPen fosters multimodal interaction and can be used as a game controller.

Multimodal & parallel interaction
VersaPen allows users to control multiple continuous sources of input at the same time. For instance, users can simultaneously control the size (using the pressure module) and the color (touch module) of a digital brush while drawing (moving the pen) with a single hand (Figure 5). One flexible extender module allows using both hands to control interactions. We used this capability to manipulate auxiliary controls with the non-dominant hand while the dominant hand was controlling the brush position and pressure. Adding a vibro-motor and a joystick module transforms VersaPen into a game controller (Figure 6).

Space-efficient interaction
Graphical applications (e.g. Adobe Photoshop) provide many commands gathered in palettes and toolbars. Graphical widgets mask a part of the screen and round trips between them and the objects of interest slows down interaction [12]. Users can assign a selection of functionalities to VersaPen modules to save screen real-estate. This customization benefits from existing hotkeys shortcuts. We used this capability in a drawing application which provides a novice and an expert mode. In novice mode, turning the wheel to select a command displays a palette on the screen helping the user to choose an action. In expert mode, the user turns the wheel without waiting for the palette to be displayed on the screen. The haptic module or a LED can act as feedback to display the current tool.

Shortcuts and automatic configuration
VersaPen buttons can be used to create shortcuts for activating frequent commands. For instance, we used this capability to change the stacking order of overlapping windows on the screen. When the user toggles a dedicated button, the window under the frontmost window comes to the front (and so on by clicking again if more windows are overlapped). This makes it possible to perform drag and drop operations between windows without releasing the pen. Letting the user set up his own shortcuts helps him to be more efficient when interacting with the system.

Tangible interaction
The modules have different colors (Figure 7), making it easy to distinguish them. When acting as data containers, these modules can encapsulate software settings, predefined behaviors or virtual data. A module can “contain” personalized properties for a given application such as a given font, theme, color, etc. A module without I/O can also serve to configure the pen, by changing the mapping between
its buttons, sensors and actuators. The user can then re-
store these parameters when she uses another computer or
changes the configuration of the pen.

**Collaboration**

Tangible interaction favors collaboration [20, 27]. We took
advantage of the capability of *encapsulating* data for ex-
changing images and videos between devices and users
(Figure 7). In a photo gallery application connected to Ver-
saPen, a user can share images by transferring it to the
data container he wants. Once exchanged to someone
else, the software can display the content stored in the
module currently mounted on the pen. The user retrieves
the image from the pen to his computing device. A remote
server is linking the static unique ID to the files, which are
retrieved when needed. Several persons can thus exchange
data through physical containers. In addition, a single user
can also use the same technique for transferring data from
a computing device to another.

**Device augmentation**

Adding a novel input to a device can increase its perfor-
man ce. VersaPen can serve as an auxiliary input for exist-
ing devices and provide additional feedback. For instance,
we created a 2D data visualization application taking ad-
vantage of the LED and vibrator modules to convey addi-
tional information (Figure 8). Conversely, VersaPen can
serve as an input device extending the capabilities of an-
other device. For instance, we extended a keyboard by
attaching the pen to the keyboard. The pen slider module
thus allows scrolling content without moving the hands from
the keyboard (Figure 9).

**Discussion & Future work**

We conducted a preliminary study with seven professionals
(illustrators, designers) who use digital pens in their work.

They used VersaPen for 20min sessions and were allowed
to switch modules and create custom interactions. Our par-
ticipants foresaw how this device could be used with their
own applications and appreciated the fact that "program-
ing [the device] is simple and compatible with existing
*keyboard shortcuts*". The modules for file sharing were
found particularly relevant: using a physical object while
working with co-located colleagues is "*better to track the lo-
cation of a file when working on the same file*". We seek to
further evaluate the usability of our system through a qual-
tative user study. Although our device inherits its final shape
from the modules, various ergonomic aspects can be tested
to improve the affordance of the pen while miniaturizing the
technology.

VersaPen is a step towards modular and multimodal de-
vices. While our work focuses on pen-based interaction,
our device raises the challenge of extending this modular
approach to other input/output devices. Rather than devel-
op ing specific devices to achieve a precise task, VersaPen
focuses on the interchangeability and the versatility of its
modules. It enables the exploration of different pen inter-
actions by adding or removing modules or by assembling
them differently. We expect this capability to inspire novel
usages of pen-based and multi-modal interaction.

**ACKNOWLEDGMENTS**

This research work was supported by the Agence Na-
tionale de la Recherche (ANR 13 CORD 008) and by the
EQUIPEX Digiscope ANR-10-EQPX-0026.
REFERENCES


