ICon: Input Device Selection and Interaction Configuration

Pierre Dragicevic

Ecole des Mines de Nantes
4, rue Alfred Kastler, La Chantrerie
44307 Nantes Cedex, France

Pierre.Dragicevic@emn.fr

Jean-Daniel Fekete

Human Computer Interaction Laboratory,
A.V. Willams Building
University of Maryland
College Park, MD20742, USA

fekete@cs.umd.edu

ABSTRACT

ICon is an editor designed to select a set of input devices and connect them to actions into a graphical interactive application. ICon allows physically challenged users to connect alternative input devices and/or configure their interaction techniques according to their needs. It allows skilled users – graphic designers or musicians for example – to configure any ICon aware application to use their favorite input devices and interaction techniques (bimanual, voice enabled, etc.).

ICON works with Java Swing and requires applications to describe their interaction styles in terms of ICon modules.

Keywords

Multiple inputs, input devices, interaction techniques, toolkits, assistive technologies

INTRODUCTION

Today, interactive desktop applications manage a very limited set of input devices, typically one mouse and one keyboard. However, the population of users requiring or simply possessing alternative input devices is growing, as well as the number of new devices available.

The complexity of supporting alternative input devices in interactive applications is currently very high: each application has to explicitly implement some code to manage each device and all the desired interaction techniques that use these devices. At best, specialized applications support a limited set of devices suited to their task.

ICon (Input Configurator) is a system for selecting alternative input devices and configuring their interaction techniques interactively.

Using ICon, users can connect additional input devices – such as tablets, voice recognition software, assistive devices or electronic musical instruments – to an ICon aware application and/or assign specific interactive behavior to connected devices. Figure 1 shows a typical configuration.

Adding new devices and configuring them using ICon not only opens applications to the use of alternative, better suited devices but also tremendously simplifies the integration of new interaction.

ICon is implemented using Java Swing and requires applications to externalize their interaction techniques to be effective. The effort required to do so is very modest compared to the benefits.
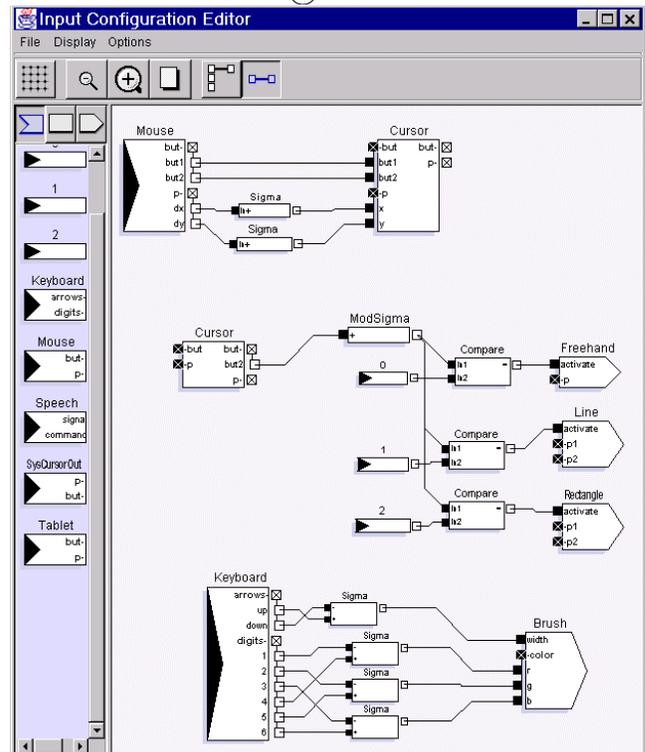

Figure 1: Screenshot of ICon showing part of the configuration of a drawing editor. The right mouse changes the selected tool and keypad keys change the color and line width attributes. Mouse moves are added and sent to a "cursor" module that abstracts a 2D locator device.

SCENARIO

In this section, we show how John, a typical ICon user, may adapt a sample application called "IconDraw" that can draw lines, rectangles and freehand curves. By invoking the "configure" menu of IconDraw, John starts ICon that displays the current configuration as in Figure 1. This dataflow diagram shows several connected blocks called modules. Each module has input or output slots where links can be connected. Only connected slots are displayed in the figures. The two input devices of the default configuration – the mouse and the keyboard – are on the left, connected to some processing modules and finally to IconDraw's interaction tools appearing as input modules.

In IconDraw's standard configuration, the mouse is connected to a cursor module which displays feedback and is used as a virtual device. The right mouse button is used to cycle through the drawing tools. Keyboard keys are used

to change the color and size of the lines. John can then change the configuration, save it to a file or load it from the configuration files available for IconDraw.

Stabilizing the Mouse Position

John remembers his friend Jane couldn't use a regular drawing program because she suffers from Parkinson's disease. With ICon, he can stabilize the pointer position by inserting a low-pass filter – averaging pointer positions to remove quick moves – between the mouse device and the cursor as shown in Figure 2.
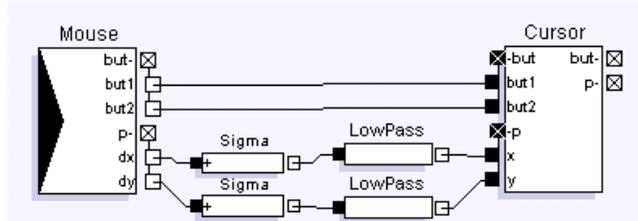

Figure 2: LowPass filters are inserted between the mouse and the cursor to stabilize the position.

Adding a Pressure Sensitive Stylus

John is a graphic designer and has a tablet with a pressure sensitive stylus. To use it inside IconDraw, he needs to disconnect the mouse and connect the tablet module to the cursor through scale modules.
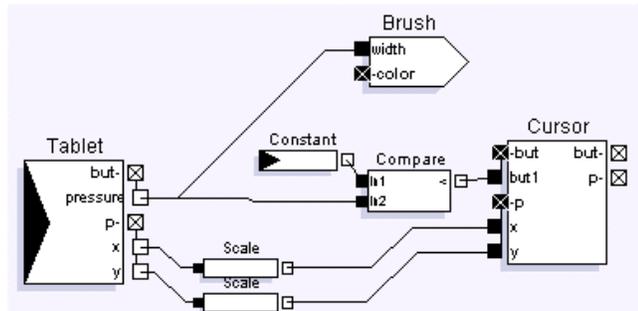

Figure 3: Adding a pressure sensitive stylus to IconDraw where the pressure changes the line width of the drawing.

To map the pressure to the line width, he connects the pressure slot of the stylus to the size slot of the brush used by the drawing tools, as shown in Figure 3. Brushes abstract graphic attributes just like cursors abstract positional devices.

Configuring for Bimanual Interaction

Figure 4 shows part of the configuration controlling IconDraw's interaction tools. John now wants to use both his mouse and the stylus. Figure 5 shows the configuration required to create a line using bimanual interaction: one

should be connected to the "p1" slot and the second to the "p2" slot. A combination of boolean modules determine when the creation mode is triggered and when it is finished.
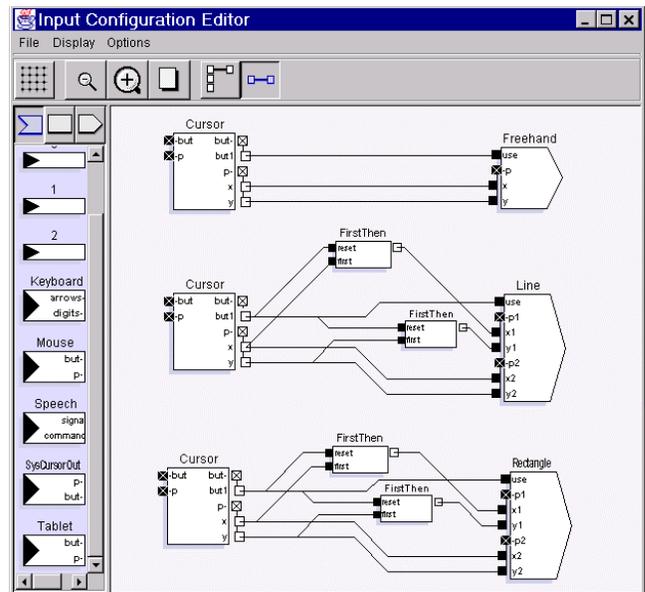

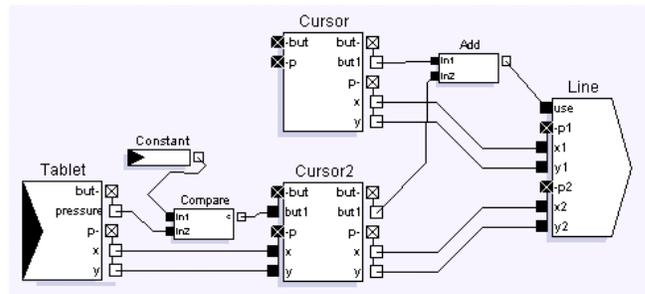*Figure 4: creation tools of IconDraw represented as input modules.*

Figure 5: Configuration for creating a line with bimanual interaction.

Current input modules also include voice recognition and toolglasses, to name a few.

CONCLUSION

ICon is easy to integrate with Java programs. It can also be used with the Piccolo toolkit available at http://www.cs.umd.edu/hcil/piccolo and it's multiple input device extension "Instrumental Jazz".

By using ICon, new multimodal applications can be created with much less complexity, the access and low-level management of input devices being already solved.

ICon can be downloaded at

http://www.emn.fr/dragicevic/ICon

REFERENCES

1. Dragicevic, P. and Fekete, J.-D., Input Device Selection and Interaction Configuration with ICon, Proceeding of IHM-HCI 2001,Blandford, A.; Vanderdonckt, J.; Gray, P., (Eds.): People and Computers XV - Interaction without Frontiers, Lille, France, Springer Verlag, pp. 543-448.