Interactivity Sketcher: Crafting and Experiencing Interactivity Qualities

Jong-bum Woo
KAIST
335 Gwahak-ro, Yuseong-gu,
Daejeon 305-701, Korea
magicianwoo@kaist.ac.kr

Jaesung Jo
KAIST
335 Gwahak-ro, Yuseong-gu,
Daejeon 305-701, Korea
nemo11.joje@gmail.com

Da-jung Kim
KAIST
335 Gwahak-ro, Yuseong-gu,
Daejeon 305-701, Korea
anes1227@naver.com

Youn-kyung Lim
KAIST
335 Gwahak-ro, Yuseong-gu,
Daejeon 305-701, Korea
younlim@kaist.ac.kr

Suin Kim
KAIST
335 Gwahak-ro, Yuseong-gu,
Daejeon 305-701, Korea
suin.kim@kaist.ac.kr

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.
ACM 978-1-4503-0268-5/11/05.

Abstract
In this paper, we introduce the Interactivity Sketcher, which is an interactivity designing tool that can visualize and experience invisible interactivity in a tangible way by controlling Interactivity Attributes(IAs). The Interactivity Sketcher is composed of the IA application, input devices, output devices, and IA controllers. The Interactivity Sketcher can help to explore various qualities of interactivity by visualizing and manipulating the relationship between an input and an output through the IA controllers and the IA application. We expect that this tool will enable interaction designers to visualize their own thoughts of interactivity qualities so that they will be able to create their design as if they had 'sketched' it.

Keywords
Sketching, interactivity, interaction design, design tool, prototyping

ACM Classification Keywords
H5.2. [User Interfaces]: User Interfaces: Interaction style

General Terms
Design

Introduction
The evolution of human-computer interaction techniques is ever ongoing. Nowadays, these new techniques are becoming formless and even more

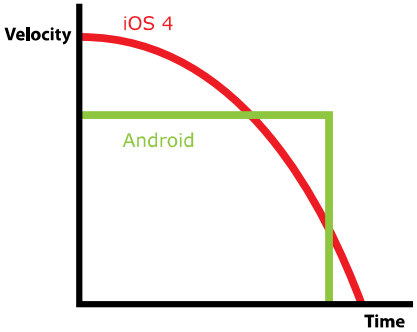

Figure 1. Velocity-time graph for bounced function at the end of list scrolling

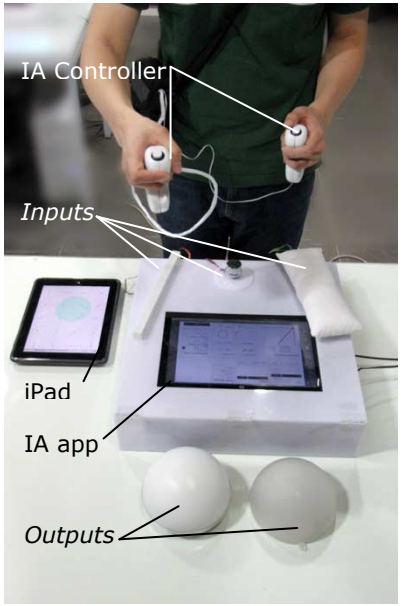

Figure 2. Interactivity Sketching Tool

dynamic. In our previous research, we developed the concept of *interactivity attribute* (IA) which provided a design language to describe the 'shape' of interactivity of an interactive artifact [5]. With *interactivity attributes* we can describe invisible qualities of interactivity in a concrete manner. For the example, the scrolling function in the OSs of the iPhone and Android have a similar interface but with very different feelings. When a user tries to scroll down from the top of a list, the list bounces at the top edge. However, the acceleration values of the bouncing are different in each OS. The list in the Android moves at a constant speed, i.e., no acceleration, but the iPhone's list moves with acceleration like a spring pendulum motion (Figure 1). This difference results in a very different user experience of the same function.

Our questions then were "How can we enable interaction designers to freely sketch this invisible but significant quality of interactivity?" and "Can we feel and experience interactivity easily and adjust its detailed qualities?" These questions led us to design and propose our Interactivity Sketcher.

There are some related works to our research. Landay, et al. introduced SILK [3], the interface sketching tool using an electronic pad and a stylus. This research is one of the original studies that showed the benefits and importance of sketch-like interaction prototyping. Our Interactivity Sketcher also promotes this aspect of prototyping, but we focused on sketching and prototyping the dynamic qualities of interactivity unlike SILK that focused on the manipulation of interface elements. Hartmann, et al. also developed a prototyping tool that has a similar motivation to ours, which is d.tools [2], a system enabling non-programmers to prototype physical user interfaces for the early stage of design. It focused on readily prototyping key interactive events assigned to each physical user interface component, which is different from what our Interactivity Sketcher targets to. Interactivity Sketcher is to prototype and to sketch invisible and dynamic qualities of interaction by manipulating visual representations using tangible user interfaces.

The Interactivity Sketcher enables the user to manipulate three Interactivity Attributes (IAs), namely, *Change Speed, Response Time, and Response Threshold*. These three IAs are used to describe any design of interactivity that depicts the relationship between an input and an output. We devised the original set of the IAs published in [4] and extracted only the ones that can be objectively measured and manipulated. *Change Speed* (which is extended and refined from *movement speed* in the original set) is the change rate of the output value change triggered by the input value change. For example, when a mouse moves the cursor on the display only a little when the input mouse is moved a lot, we feel that the cursor moves very slowly in reaction to our mouse movement, and vice versa. *Response Time* is the time delay from an input action to the output reaction caused by the input action. *Response Threshold* is the threshold of an input value that triggers the output to happen. By controlling these three IAs, the Interactivity Sketcher enables visualization and manipulation of the invisible qualities of interactivity.

The Interactivity Sketcher will enable interaction designers and developers to 'craft' and create new forms of interactivity, which is analogous to carpenters using their tools for wood crafting to create beautiful, high-quality, expensive tables in an unlimited number

Figure 4. IA Controller

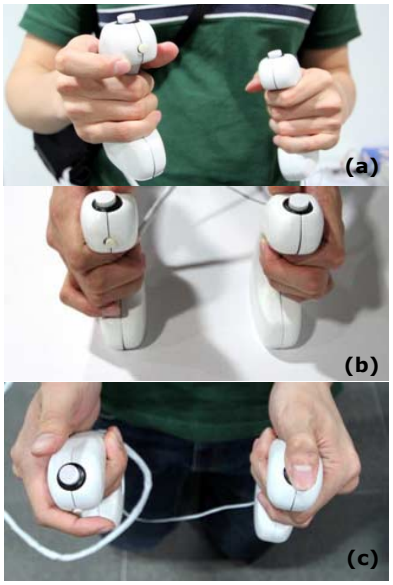

Figure 5. (a) Changing Speed (b) Response Range (c) Response Delay

of different forms. The Interactivity Sketcher provides a platform and a mechanism to embody invisible interactivity shapes into concrete and manipulative forms so that designers and developers can craft detailed qualities of interactivity that help produce the most desirable experiential qualities of the designed interactivity.

The Structure of Interactivity Sketcher

The Interactivity Sketcher works through its four parts, namely, *Inputs*, *Outputs*, *IA controllers*, and *IA application* (Figure 2). Designers can select an input and an output among the sets of input- and output-components through the IA application. Designers can then define the relationship between the input and the output components by manipulating the IA controllers. When they manipulate this relationship, they control the values of the interactivity attributes—*i.e.*, *Change Speed, Response Time, and Response Threshold*. The IA app visualizes each interactivity attribute in real time.

Implementation

Input devices and Output devices

We provide three inputs and two outputs and a touch screen (Figure 3). We select the fundamental types of input components and output components to find applicable usages. The selected inputs are a flex sensor, a potentiometer, and an accelerometer. For the input value, a designer can change the degree of bending for the flex sensor, the angle of the potentiometer, and the speed of shaking the accelerometer. For the output parts, a motor and LEDs are used. To cover various ways of expressing outputs, we made a list of output behaviors for each of them—e.g. moving forward and moving zigzag for the motor; blinking, alternating, and dimming for the LEDs.

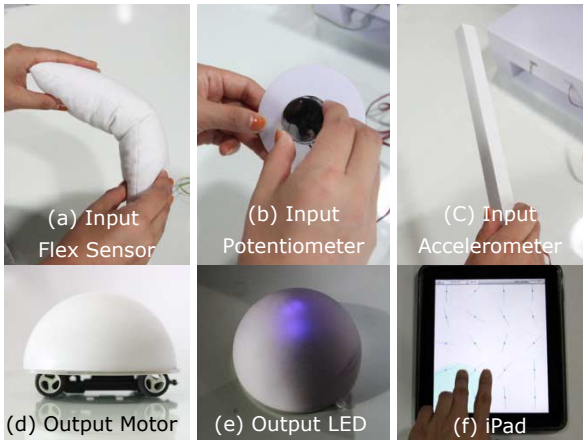

Figure 3. (a) Flex sensor, which is covered with a soft cloth; (b) Potentiometer with a knob; (c) accelerator with a stick-shape case; (d) LED; (e) motor; (f)iPad.

For the touch screen, we used an iPad (Figure 3(f)). The iPad is a special device that has both input and output features—i.e., touch interface as input and display as output. We devised a standalone application for iPad through which it can communicate with the Interactivity Sketcher's features. The bars in this application rotate as finger moves on the touched surface. The rotation speed changes according to the finger's movement speed and the distance from a finger to a bar. The affected area is altered by controlling the *Response Threshold*, and the rotation speed is manipulated by controlling the *Change Speed* with the IA controllers.

Input devices are connected to a PC by a wired USB hub. Output devices are connected to the PC by wireless Xbee technology[5] to make the outputs free to move. Data from the manipulation of the IA controllers are transferred to the PC also by Xbee so

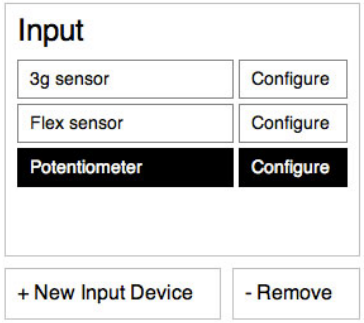

Figure 7(a). Select Input section of IA app

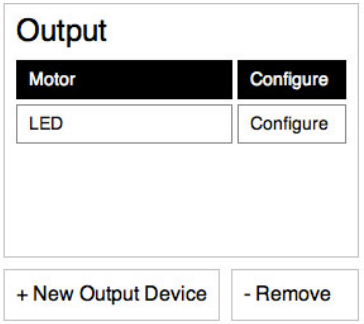

Figure 7(b). Select Output section of IA apps

that the 'interactivity sketching' activity with these controllers can be freely done (Figure 4). Different data channels were set to prevent any signal mixing problems.

IA controllers

IA controllers are one of the main parts of the Interactivity Sketcher and are for manipulating interactivity attributes. IA controllers control the three attributes, namely, *Change Speed, Response Time*, and *Response Threshold*, by an accelerometer, an ultrasonic sensor, and four buttons (Figure 6).

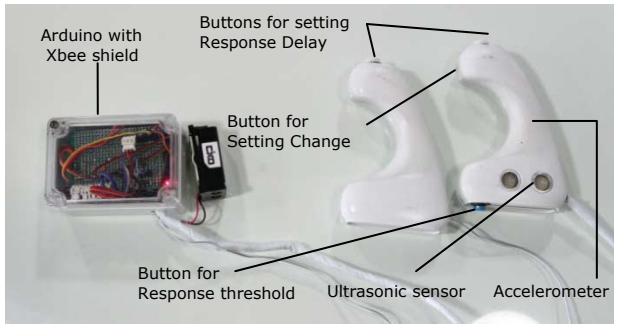

Figure 6. Structure of IA controllers

To enable users to naturally understand and manipulate the IA controller, we adapted metaphorical behaviors of each attribute when designing the IA controllers. To manipulate *Change Speed*, a user changes the *speed* of shaking the IA controllers while pushing the right-hand button (Figure 5(a)). To manipulate the *Response Threshold*, a user adjusts the distance between the left and the right sides of the controllers. Since there is a button underneath one controller, distance sensing is naturally activated when a user moves both IA controllers on a hard surface like a table where the button underneath can be pressed (Figure 5(b)). The distance metaphorically matches the concept of threshold, so we applied this method for controlling the *Response Threshold*. A user can change the *Response Time* by controlling the interval time between pushing the left button and the right one, like a stop watch (Figure 5(c)). The detailed components that make up the IA controllers are shown in Figure 6.

Arduino[1] with Xbee shield[7] is used for the IA controllers' microprocessor. Arduino transmits the attribute value data of the controlled interactivity to the IA app by wireless Xbee technology.

IA App

The IA app is another main part of the Interactivity Sketcher, which the software that serves as the central processor. This software connects four components: inputs, outputs, IA controllers, and a PC. The IA app runs on the PC. The IA app transfers and processes data across those components. The IA app enables users to control the interactivity attribute values with visual feedback (Figure 7).

The graph (Figure 7(c)) visualizes the relationship between the input and output values that a user creates by manipulating the IA controllers. Each of three IA values is also visualized as a real-time diagram: the rotation speed of a small circle on a circular track for Change Speed (Figure 7(d)); the size change of a blue circle for Response Threshold (Figure 7(e)); the line length for the Response Time (Figure 7(f)). Both input- and output-values are displayed on the graph. An output value is calculated when the PC receives an input value from a selected input device (Figure 7(a)). The PC sends the output value to the

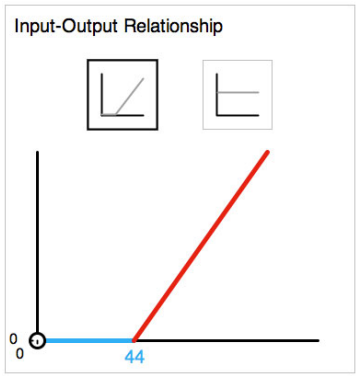

Figure 7(c). Input-Output Relationship section in IA app

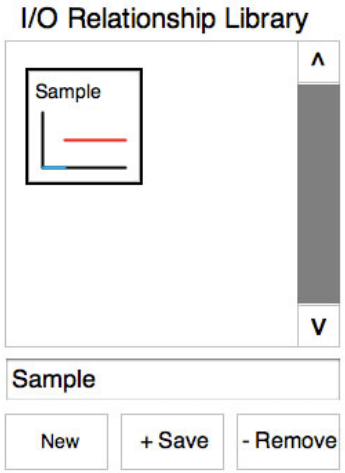

Figure 7(d). I/O Relationship section in IA app

selected output device (Figure 7(b)). The user can change the shape of the graph using the IA controllers. Elements in the graph are updated in real-time, which allow a user to react on the changes instantly. There are two types of graphs: one is a linear type graph and the other is a constant type graph. The IA app can also save a created graph to the I/O Relationship Library (Figure 7(g)). The I/O Relationship Library represents a collection of various qualities of interactivity. It visualizes each quality of interactivity designed by a user as a graph that contains the numerical values of the interactivity attributes.

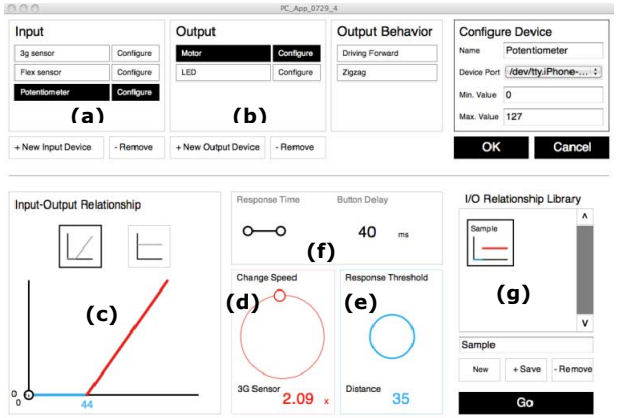

Figure 7. Layout of IA App (a) Select Input (b) Select output (c) Input-output relation (d) Change speed (e) Response Threshold (f) Response Time (g) I/O Relation Library

The IA app is developed with Processing [6]. The IA app is installed in a touch-screen notebook computer. The touch interface is accessible using a notebook computer's display panel, so there will be no need for additional interfaces like a mouse or a keyboard. It is also packaged in a custom-made acrylic kiosk as shown in Figure 2.

User Experience

In order to examine the effect of the Interactivity Sketcher in interaction design practice, we tested it with five designers who are in the master's program of the Dept. of Industrial Design at KAIST. We recruited designers who have experience in interaction design. Three types of tasks were given to the designers: 'designing a friendly on/off lamp', 'designing a vital robot cleaner', and 'designing one more on/off lamp or a robot cleaner with personality'. The reason we chose a lamp and a simple robot as design targets was that they effectively represent characteristics of our output devices—i.e., LEDs and a motor.

User Reaction

Most of the participants commented that the Interactivity Sketcher is a very unique tool for actually 'sketching interactivity'. Through the Interactivity Sketcher, the participants concretely craft their interactivity qualities, and they could test their ideas immediately. They could also adjust the detailed part of the interactivity. "I tested and experienced my ideas with the Interactivity Sketcher. If it felt little different from what I first expected, I changed the attribute values to adjust it to meet my idea. I am usually not concerned about these kinds of deep details of interactivity, but the Interactivity Sketcher made me consciously think of this detailed quality of interactivity."

The Interactivity Sketcher has a huge effect on designing. When the participants were asked to express an interactivity that fits a given emotional keyword like 'friendly' or 'vital', they first tried to visualize their interactivity idea using the Interactivity Sketcher. At first, their idea was vague, but they made several adjustments to match their ideas like sketching. "When

I first thought the emotional keyword 'vital', the idea of speedy movement came to mind. I wanted an immediate reaction to my action, so I set a no-delay for the Response Time. The slope of the graph would have to be steep to change instantly with my action. I tried various input and output behaviors to fit to my idea."

When participants heard about the task, they usually first imagined a behavior that was related to the emotional keyword. "When I decided to design an indecisive robot cleaner, I imagined that the robot cleaner was really careful and hesitated to clean my room. Usually a robot cleaner is too tough to sweep. I wanted the robot cleaner to act carefully with consideration, so I set a little delay and chose the bar type graph to make it move with a constant velocity.

Further Development & Conclusion

The Interactivity Sketcher is an exploration tool for designers' interactivity ideas that exist only in their minds. Through the Interactivity Sketcher, interactivity will not be invisible. Designers can visualize, archive, communicate, and experience various qualities of interactivity in their design. We expect new possibilities of designing interactivity through the Interactivity Sketcher; this will enable designers to create more precise and rich interactivity and new interaction experiences.

For the future work, we see the potential of Interactivity Sketcher as an educational tool for interaction design. Students may be able to freely configure inputs, outputs, and change interactivity attribute values to define the relationships between those inputs and outputs to experience various qualities of interactivity. Regarding the improvement of Interactivity Sketcher, it can be extended to represent various dimensions and variables related to interactivity sketching other than the linear format of representation.

Acknowledgements

This research was supported by the following programs: 1) HRHR(High Risk, High Return) program funded by KAIST, 2) Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0023206), 3) KMCC(KAIST-Microsoft Research Collaboration Center) funded by KAIST and Microsoft Research Asia, 4) Korea Research Foundation Grant funded by the Korean Government(MOEHRD) (KRF-2008-332-G00030). We would like to thank all the user participants. We also like to acknowledge Donald A. Norman's valuable inputs to this project.

References

[1] Arduino. http://arduino.cc/

[2] Hartmann, B., Klemmer, S.R., and Bernstein, M. 2005. d.tools: Integrated prototyping for physical interaction design. IEEE Pervasive Computing, Oct-Dec 2005.

[3] Landay, J.A. and Myers, B.A. Interactive sketching for the early stages of user interface design. In Proceedings of CHI '95: Human Factors in Computing Systems, Denver, CO, May 1995, pp. 43-50.

[4] Lim, Y. Lee, S., Lee, K., Interactivity attributes: a new way of thinking and describing interactivity. CHI2009, Boston, MA, USA. ACM press.

[5] Lim, Y. Stolterman, E., Jung, H., Donaldson, J. Interaction gestalt and the design of aesthetic interactions. DPPI 2007, Helsinki, Finland. ACM press.

[6] Processing. http://processing.org/

[7] Xbee shield. http://arduino.cc/en/Main/ArduinoXbeeShield