Reality Editor: Programming Smarter Objects

Valentin Heun
Fluid Interfaces Group
MIT Media Laboratory
20 Ames Street
Cambridge, MA 02139 U.S.A.
heun@media.mit.edu

James Hobin
Fluid Interfaces Group
MIT Media Laboratory
20 Ames Street
Cambridge, MA 02139 U.S.A.
hobinjk@MIT.EDU

Pattie Maes
Fluid Interfaces Group
MIT Media Laboratory
20 Ames Street
Cambridge, MA 02139 U.S.A.
pattie@media.mit.edu

UbiComp'13 Adjunct, September 8–12, 2013, Zurich, Switzerland.
ACM 978-1-4503-2215-7/13/09.
http://dx.doi.org/10.1145/2494091.2494185.

Abstract
The Reality Editor is a system that supports editing the behavior and interfaces of socalled "smarter objects", i.e. objects or devices that have an embedded processor and communication capability. Using augmented reality techniques, the Reality Editor maps graphical elements directly on top of the tangible interfaces found on physical objects, such as push buttons or knobs. The Reality Editor allows flexible reprogramming of the interfaces and behavior of the objects as well as defining relationships between smarter objects in order to easily create new functionalities. This paper describes the different functionalities of the Reality Editor and presents several examples.

Author Keywords
Augmented Reality; Ubiquitous Computing; Internet of Things

ACM Classification Keywords
H.5.2 User Interfaces; H.5.1 Multimedia Information Systems

Introduction
In the last couple of decades, a lot of HCI research has focused on the problem of providing a closer integration of the physical world of atoms and the digital world of

bits and information. While the physical world of objects appeals to more of our senses, it is less adaptable and customizable than the digital world, which can be modified infinitely. Hence researchers have sought ways to combine the best of both worlds.

The vision of Ubiquitous Computing is that computers will be interwoven into everyday objects to support everyday interaction with these objects. Recent writing by Gregory D. Abowd [1] suggests shifting the focus of Ubiquitous Computing research towards design problems. Abowd argues that nowadays the technologies for building Ubiquitous Computing projects exist, but the field lacks good methods and visions for how people can interact with Ubiquitous Computing environments.

The work presented in this paper combines the idea of Ubiquitous Computing with the possibilities of Augmented Reality. The Reality Editor is an interface that allows one to program and operate smarter objects with the help of Augmented Reality technology. Our approach takes full advantage of the unique benefits of the physical and virtual interface components though optimal distribution of operation tasks: setup and customization of the object happens through the augmented reality graphical interface while everyday operation can be performed using the physical interface elements. The Reality Editor associates a virtual object with each physical object and enables the reprogramming of the physical object's behavior using an intuitive, visual interface. As such, it introduces a new model for ubiquitous computer interaction.

Related Work

Augmented reality editors or browsers such as Junaio [2], Layar [3] or Wikitude [4] provide the ability to generate or view content on top of newspapers, images or city landscapes. These commercial applications are enabled by progress in the field of image recognition and also by the improved processing power of mobile devices [5] so that anyone with a standard smart phone can now use these applications. The applications typically augment the physical world with rich content. There has been early research on augmenting electronic devices and generating hybrid objects [6].

AR researchers have started to explore the intersection with Ubiquitous Computing [5] [7] and [8], for example to read sensor data and support the maintenance of complex machines. Some of the latest work has shown novel concepts for direct mapping of virtual interfaces. By separating the interaction in to an understanding, setup and programming part a simplification for interacting with the physical object could be archived [9]. Other novel concepts show interfaces for robotic control that reach beyond the pure observation of systems [10][11] and enable the remote control of robots using AR technology. These works show the potential of augmenting interfaces and the benefits that the direct mapping of interaction can provide. Some other work has shown that interaction augmented through a screen onto a real image [12] can provide a better intuitive interface than an abstracted representation of such a system in the form of a two-dimensional interface. Examples of interfaces for ubiquitous computing can be seen with products such as NinjaBlocks [13] or SmartThings [14].

These related projects show that it is beneficial to map augmented interfaces on top of physical devices to generate a direct mapping. However none have presented a more generic solution for programming the behavior and interactions of a wide variety of smarter objects.

The Reality Editor

The Reality Editor is a system that enables one to program and operate objects that are represented both digitally and a physically. It has three main functionalities:

The first functionality of the Reality Editor is a directly mapped interface that allows one to operate interactions with real objects right on the spot.

The benefits of such direct mapping can be explored with a simple example scenario: Imagine a house with a kitchen, living room, bedroom and bathroom, where every room contains at least one light. A typical graphical user interface for controlling the lights on a computer screen or smartphone would represent the lights by numbers, lists or with a categorization of symbols, but never mapped to the actual position in one's home.

The light in the kitchen could be named simply "light in the kitchen". But what if there are 3 lights in the kitchen? Of course they could have the name "light one, two, and three". But this become complicated, as the user must memorize a mental image of the actual position mapped to the number. If the system gets even more complex with multiple lights, doors, radios and kitchen devices, it becomes impossible to keep all positions and relations in one's mind. This example shows the disadvantages in terms of scalability of such approaches for remote control of smarter objects.

With the Reality Editor one can just hold a device over the light that needs to be controlled and a virtual object is displayed which can be manipulated to change the light's settings. No mental relation between object and virtual interface needs to be remembered. A minimum amount of abstraction and mental demand is achieved when a user has a direct view of the object of interest and manipulates it on the spot.

The Reality Editor provides the described direct mapping of operation and therefore changes complex operations and programming into more intuitive tasks. Objects can be operated on the spot.

Aside from the operation of an object, another functionality of the Editor is a visual editing functionality. This editing functionality lets one reprogram and manipulate the behavior of the devices objects to combine their functionalities. of different physical objects. In this mode every knob, button, speaker or screen of the smart object has a virtual tag. These tags can be used to connect the functions of an object with other tags of the same or other objects. For example, the virtual object for a desk light can consist of a switch tag and brightness, hue and saturation tags. The switch provides the on and off functionality. Connecting the switch tag to the brightness tag represents the basic functionality of the desk light.

By connecting tags of different objects the user can program multi-object functionality. For example a radio has the functionality of a tuning knob, a volume knob and a speaker. By disconnecting the tuning knob and

volume knob and connecting the output of the tuning knob with the input of the light switch and connecting the output of the light switch with the input of the volume knob, the radio turns on and off whenever the light is turned on and off. This simple example shows how editing everyday objects and their functionalities can be flexible, creative, and personalized.

As a third function, the Reality Editor provides the ability to freeze the image of an object so that all interactions can be performed on a still image. This makes it possible for a user to use the reality editor, to investigate and program smarter objects from anywhere, even when the user is not in the vicinity of the object Since the interface is still shown on top of a real image of the device being controlled, a strong connection between the real object and the Mixed Reality Editor remains. For example one can freeze an image of the radio and take it into the living room and still be able to operate the radio from the distance.

These "frozen" images can be placed in a memory bar that is located on the side of the Editor. In order to save the image the user has to tap the screen for 3 seconds in an area that is not occupied by the graphical interface of the shown object and then moves it to a free spot in the memory bar. A user can click on the spot to edit the interface that has been stored.

Conclusions

The Reality Editor demonstrates how a direct mapped interface can be used to enable interfacing with a varied range of physical objects, thereby providing a very simple way of programming the behavior and interactions of physical objects. It simplifies control by removing the need for complex abstractions. The user can now focus on a playful and creative interaction with smarter objects.

REFERENCES

[1] Abowd, G. What next, Ubicomp? Celebrating an intellectual disappearing act. UbiComp' 12.

[2] Layar. http://www.layar.com/.

[3] Junaio. http://www.junaio.com/.

[4] Wikitude. http://www.wikitude.com/.

[5] Wanger, D. 2009. History of Mobile Augmented Reality, Communications. Retrieved from https://www.icg.tugraz.at/~daniel/HistoryOfMobileAR

[6] Fitzmaurice, G. Situated information spaces and spatially aware palmtop computers, Commun. ACM 36, 7 (July 1993), 39-49.

[7] Feiner, S., et al., Knowledge-Base Augmented Reality, C. o. ACM 1993/Vol.35, No. 7

[8] Raskar, R., et al., iLamps: geometrically aware and self-configuring projectors, SIGGRAPH '06 ACM SIGGRAPH 2006 Courses Article No. 7

[9] Heun, V., Kasahara, S., Maes, P., Smarter objects: using AR technology to program physical objects and their interactions. CHI '13, WIP.

[10] Leckie, W., Greenspan, M., ARPool, RCV Lab, 2006 http://rcvlab.ece.queensu.ca/~qridb/ARPOOL.html

[11] Sunao Hashimoto, et al., TouchMe: An Augmented Reality Based Remote Robot Manipulation, ICAT, 2011

[12] Kasahara, S., Niiyama, R., Heun, V., exTouch: Spatially-Aware Embodied Manipulation of Actuated Objects Mediated by Augmented Reality, TEI 2013.

[13] Nina Blocks, http://ninjablocks.com

[14] Smart Things, http://www.smartthings.com