The Feedforward Torch

Jo Vermeulen Kris Luyten Karin Coninx

Hasselt University – tUL – IBBT
Expertise Centre for Digital Media
Wetenschapspark 2, B-3590 Diepenbeek, Belgium
[jo.vermeulen,kris.luyten,karin.coninx]@uhasselt.be

ABSTRACT

An important challenge in deploying pervasive computing environments is the difficulty users have in understanding their behaviour. It has been suggested that these environments should be made intelligible by informing users about their understanding of the world. We propose the "Feedforward Torch", a technique that combines a mobile phone with a pico projector to help users in an pervasive computing environment understand what the result of their actions will be.

Categories and Subject Descriptors

H.5.2.m [**Information Interfaces and Presentation (e.g., HCI)**]: Miscellaneous

General Terms

Human Factors

Keywords

intelligibility, feedforward, understanding, control, pervasive computing, ubicomp

1. INTRODUCTION

An important usability challenge for pervasive computing systems – and context-aware systems in general – is the difficulty in understanding how they work. Due to the fact that these systems typically function on the periphery of human awareness and react on implicit input gathered from sensors without user involvement (such as location changes), users have difficulty predicting the behaviour or even the available features of the system [7]. Moreover, it is often not clear to users how they can override actions taken on their behalf. Unfortunately, these issues might eventually result in users abandoning or refusing to adopt the system [5].

This problem can be addressed by making pervasive computing systems *intelligible* [2]. Intelligible systems have built-in support for helping users understand how they work, what they are doing and why. Examples of techniques to provide intelligibility are textual explanations to questions [9, 3] (*e.g.,* "Why (not)?", "What if?", "How to?"), visualisations of devices, sensors, system events and actions [8], and visual, domain-specific explanations [4].

This paper focuses on *feedforward*, a specific type of intelligibility that tells users what will happen when they perform a certain action. Well-designed feedforward is an effective tool for bridging Norman's Gulf of Execution [6] – the gap between a user's goals for action and the means for executing those goals. Feedforward has been succesfully applied in gesture-based interaction to help users learn, perform and remember gestures [1]. Additionally, Lim and Dey's "What if?"-questions [3] can be seen as a type of feedforward for context-aware systems.

In this paper, we propose the *Feedforward Torch*, a combination of a mobile phone and pico projector to provide feedforward about different (smart) objects in their physical environment. Previously, we used steerable projectors to overlay an environment with a graphical representation of the different available sensors and devices and how they are connected to each other [8]. The Feedforward Torch can be seen as a continuation of that work, which is mobile, less intrusive and on-demand instead of always-on. We have currently built a Wizard of Oz prototype of the system, and are setting up a user study to investigate the suitability of this technique.

2. THE FEEDFORWARD TORCH

Feedforward tells users what the result of their action will be. In WIMP interfaces, users typically perform actions using an input device such as a mouse or keyboard (*e.g.,* clicking a button, typing a character, moving the mouse pointer). In pervasive computing environments, however, the range of possible user actions that might trigger a certain system response can vary widely depending on the available sensors and can either be implicit or explicit. Examples include moving around (location changes), different types of physical activity (running, walking, performing gestures) or making sounds (*e.g.,* talking, listening to music, speech commands).

The different nature of pervasive computing environments imposes a number of specific requirements for providing feedforward. First, feedforward should be conveyed in a uniform way in different situations, regardless of the type of user action that will trigger the command. Secondly, feedforward should be available at all times, wherever the user might be. For example, suppose Bob is in the bedroom and wonders at what time his smart home will automatically close the shutters in this room. We would not want Bob to have to go downstairs to consult a feedforward interface on a touch-screen panel in the living room. Ideally, he should be able to "select" the shutters in the bedroom in a natural way and get information about their behaviour. Finally, to make sure users can easily associate feedforward with the corresponding physical objects in their environment, it would be beneficial to integrate feedforward as closely as possible with the physical environment. It would thus be better to overlay feedforward on a photorealistic view of the environment, instead of on a more abstract floorplan.

The Feedforward Torch allows users to point at objects in their environment and reveal feedforward information about them, as if they were located under a spotlight. Users are shown under which conditions actions associated with the object will be executed by the system (*e.g.,* a displacement in time or space), so that they can anticipate and adapt their behaviour, if necessary. Based on findings with a similar technique using fixed projectors [8], we employ animations to better convey the effect an action will have.

Figure 1 shows how this would work in practice. As the user approaches a display, the Feedforward Torch shows that moving towards the display will cause a movie to be played and will turn off the lights. This is visualized using a horizontal progress bar and arrow indicating the remaining distance the user needs to walk forward before the actions will be triggered.

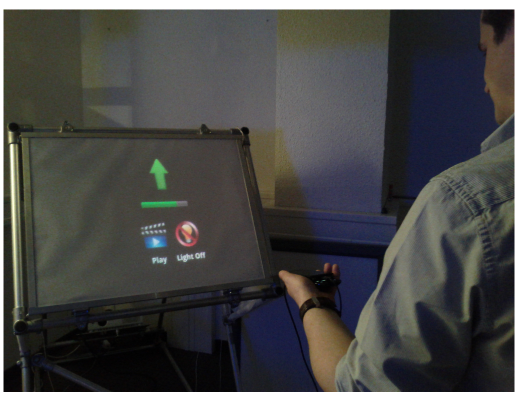

Figure 1: The Feedforward Torch shows a user that the lights will be turned off and that a movie will be played when he approaches a display.

3. DISCUSSION

We have identified a number of opportunities and challenges of our approach.

Mobile Projection It is not yet clear if a pico projector would be the right choice, due to the need for low light conditions and a suitable projection surface. Another approach could be a more traditional AR approach, where information is overlaid on a live camera view. While using a projector has the added advantage of showing information in a co-located way (*i.e.,* overlaid on the object), this advantage is lost when the result of an action affects an object outside of the user's direct vicinity (*e.g.,* closing the shutters in a different room). One approach here might be to visualize the effect using a projected floorplan, but this can also easily be done on a mobile phone display. Next to hardware issues, it is important to provide visualizations which are easy to understand.

Recognition and Modelling For this prototype to work in real-world scenarios, we need both object recognition – to know which object the user is pointing to – and a detailed model of the environment – to know what will happen given a certain action by the user and show corresponding feedforward. Especially accurate object recognition in different conditions might prove to be a difficult task.

Applicability beyond pervasive computing This technique could also be used to provide feedforward about everyday "dumb" (but possibly complex) objects in our environments, such as arrays of unlabeled light switches or complex devices (*e.g.,* copiers, coffee machines). Of course, a prerequisite for this is the availability of a model of the environment and the available objects and devices in it.

Acknowledgements

The authors would like to thank Gert Vos for developing the Wizard of Oz prototype in the context of his Master's thesis at Hasselt University.

4. REFERENCES

[1] Bau, O., and Mackay, W. E. OctoPocus: a dynamic guide for learning gesture-based command sets. In *Proc. UIST '08* (2008), ACM, pp. 37–46.

[2] Bellotti, V., and Edwards, W. K. Intelligibility and accountability: human considerations in context-aware systems. *Hum.-Comput. Interact. 16*, 2 (2001), 193–212.

[3] Lim, B. Y., and Dey, A. K. Toolkit to support intelligibility in context-aware applications. In *Proc. Ubicomp '10* (2010), ACM, pp. 13–22.

[4] Lim, B. Y., and Dey, A. K. Design of an intelligible mobile context-aware application. In *Proc. MobileHCI '11* (2011), ACM, pp. 157–166.

[5] Muir, B. M. Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics 37*, 11 (1994), 1905–1922.

[6] Norman, D. A. *The Psychology Of Everyday Things.* Basic Books, New York, USA, June 1988.

[7] Rehman, K., Stajano, F., and Coulouris, G. Interfacing with the invisible computer. In *Proc. NordiCHI '02* (2002), ACM, pp. 213–216.

[8] Vermeulen, J., Slenders, J., Luyten, K., and Coninx, K. I bet you look good on the wall: Making the invisible computer visible. In *Proc. AmI '09* (2009), Springer-Verlag, pp. 196–205.

[9] Vermeulen, J., Vanderhulst, G., Luyten, K., and Coninx, K. PervasiveCrystal: Asking and answering why and why not questions about pervasive computing applications. In *Proc. Intelligent Environments (IE) '10* (2010), pp. 271 –276.