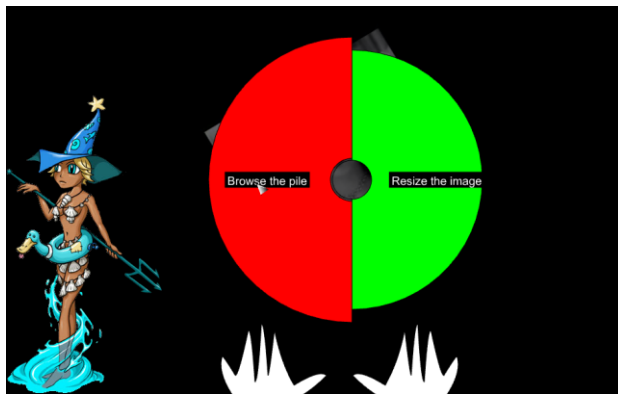


# Ghosts in the Interface: Meta-user Interface Visualizations as Guides for Multi-touch Interaction

Davy Vanacken, Alexandre Demeure, Kris Luyten, Karin Coninx  
Hasselt University – tUL – IBBT  
Expertise Centre for Digital Media  
Wetenschapspark 2, 3590 Diepenbeek, Belgium  
firstname.lastname@uhasselt.be



## Abstract

*Multi-touch large display interfaces are becoming increasingly popular in public spaces. These spaces impose specific requirements on the accessibility of the user interfaces: most users are not familiar with the interface and expectations with regard to user experience are very high. Multi-touch interaction beyond the traditional move-rotate-scale interactions is often unknown to the public and can become exceedingly complex. We introduce TouchGhosts: visual guides that are embedded in the multi-touch user interface and that demonstrate the available interactions to the user. TouchGhosts are activated while using an interface, providing guidance on the fly and within the context-of-use. Our approach allows to define reconfigurable strategies to decide how or when a TouchGhost should be activated and which particular visualization will be presented to the user.*

## 1. Introduction

As a result of the standardization of graphical user interfaces, a user can often rely on prior knowledge of other applications when confronted with a new user interface. Traditional widgets in desktop environments

provide affordances to the user, “suggesting” how an object may be interacted with, while the visibility principle states that designers should make the important aspects of a system visible to the user. Ideally, which actions a user can perform on an interface and how the actions can be executed, should be obvious just by looking at the interface.

In contrast, multi-touch interface components allow multiple concurrent points of control that are often not visually represented in the interface. Due to this characteristic design of multi-touch user interfaces, first-time users have a difficult time figuring out what can be done with the interface and how it can be done.

Especially multi-touch setups in public spaces suffer from this problem: the limited time and “direct use” of the system do not allow for much exploration of the interaction techniques. This leads to a rather complex situation, in which the multi-touch interface is assumed to support very intuitive interaction through gestures that come close to real-world interaction [6]. However, users experience difficulties finding out how to interact with such an interface due to the amount of different gestures and the lack of visibility and familiarity. In early deployments of large multi-touch displays in public spaces [2], users often try to operate the interface with just one index finger. Only after some time they try to use their second index finger.

This paper proposes an approach to learn multi-touch interaction “on the spot”: visual guides are merged with the actual user interface to inform the user about the supported interaction techniques.

## 2. Related work

Coutaz [1] introduces the idea of a meta-User Interface as a user interface dedicated to control and evaluate the state of an interactive ambient space. The meta-UI concept is an expansion at the ambient computing level of what the shell or desktop metaphor is at the personal computer level. Our objective is much more restricted, but can be related to the general goal of the meta-UI in the context of multi-touch interactive surfaces.

Vogel [15] presents UIs that self reveal help using looping video sequences. However, the paper don’t focus on when trigger the help, how to choose a help topic and how to represent it.

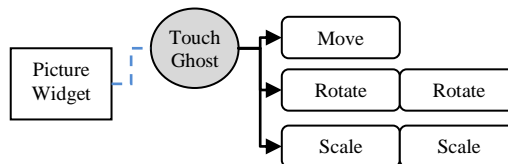
Most of the existing literature addresses help systems in environments other than multi-touch user interfaces. Myers [9] proposes an application framework that aims to answer the user’s questions on why something happens in an interactive system or why not. Unlike many help designs that focus on shortening the gulf of execution, Myers’ method aims to shorten the gulf of evaluation [10] by helping users to interpret what they exactly perceive on the screen and to determine how to alter the outcome if necessary. The philosophy of this approach is similar to ours, as we aim to assist users in understanding how they can interact with a multi-touch user interface.

Some literature discusses how we can improve the initial interaction between users and help systems [4], or how help can be presented to the user, including animated demonstrations [11], animated context-sensitive help [14], and task-oriented [12] or dialog-based help [13]. Multi-layered interfaces [7] provide integrated initial guidance, to help users get started with a new interface. Other studies provides research data on how end users actually behave when using embedded help [5] or what effect the placement of help messages has on performance [8]. Although none of this literature takes multi-touch interfaces into account, a lot of the concepts are relevant to our approach.

## 3. TouchGhost: self-explanatory interface

The foundation of our work is the concept of a self-explanatory touch-based user interface: a *TouchGhost interface*. A TouchGhost interface embeds context-sensitive help, illustrating which functionalities are offered by the system and how the user can apply

them. A TouchGhost is a meta-level interface object, providing help for a particular interface component by associating visual guides to every action the component supports. If an interface supports the moving, rotating and scaling of pictures, we call the set of visual guides demonstrating these actions the TouchGhost of the picture component (Figure 1).



**Figure 1. A TouchGhost object can associate multiple visual guides to every action a user interface component supports.**

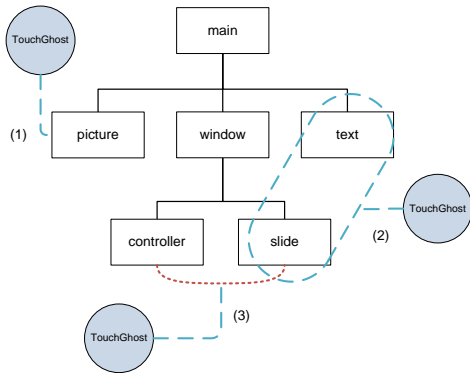
Possible visualisations of a TouchGhost guide include playing a small video, representing the user’s fingers by small dots or simulating real actions with virtual, animated hands. The latter options have the distinctive advantage of presenting the guidance directly on the interface components the user is currently working with, while a video fragment can only show a generic example.

As Figure 1 shows, a TouchGhost object can associate multiple guides to every action the interface component supports. Each guide can present the help to the user in a different way, so the designer is free to provide different visualizations for each action. TouchGhost strategies eventually decide how or when a TouchGhost is activated and which particular visual guide will be presented to the user. Strategies will be discussed in more detail in the next section.

A TouchGhost can be associated to a single user interface element (e.g. a picture, Figure 2-1), a set of user interface elements (e.g. two objects that can be merged, Figure 2-2) or a relationship between several user interface elements (e.g. a controller that pilots a slideshow, Figure 2-3). A TouchGhost manager can traverse the graph and make the necessary TouchGhost available through the user interface.

In order to accomplish this, we make use of a special purpose user interface toolkit that allows us to query the underlying semantics of a user interface: the COMETs toolkit [3]. The COMETs toolkit is based on high-level interactors (e.g. a “choice” widget instead of a radio button widget). COMET widgets can be decorated with meta-data, expressing their role in the interactive system. As a result, we can query a concrete user interface for the meaning of user interface widgets (e.g. which task they support) and relationships between different parts of a user interface (e.g. whether a ‘draggable’ object is ‘droppable’ into a zone). For further information on the Comets toolkit itself, we

refer the reader to [3]. Although the use of the COMETs toolkit is not necessary, it provides us a valuable level of abstraction.



**Figure 2. TouchGhosts attached to different parts of a user interface scene graph: (1) to a single element, (2) a set of elements, (3) a relationship.**

#### 4. TouchGhost strategies

A TouchGhost interface can be configured through *TouchGhost strategies*. These reconfigurable strategies decide how or when a TouchGhost should be activated and which particular visual guide will be presented to the user. In this section, we propose a wide variety of strategies that can be explored, ranging from strategies where the user has to call for help explicitly, to implicit strategies, where the system detects the need for guidance. Depending on the strategy, particular input from the system and/or the user may be required, as indicated in the following examples.

A straightforward example of an explicit strategy is simply pressing a ‘help’ button or avatar, which lists all the available actions on an interface component in a menu. The user can select the appropriate action in the menu, after which the corresponding visual guide is shown. Alternatively, with the right hardware support, a user can hover with a finger above a user interface component or look at it longer than a predefined threshold to invoke all the TouchGhost guides one after another. The order in which the guides are presented, can be dependent on general usage statistics (e.g. show the guide related to the most common action first). In that case, a TouchGhost observer service needs to be running alongside the user interface, recording all events that are done on the display surface.

Implicit strategies can further exploit the information collected by a TouchGhost observer to detect the need for guidance automatically. The user might be operating a user interface component very ineffectively without knowing the existence of a more

efficient technique. Based on information collected from the system’s previous users, such as the usage history (e.g. which actions occur often, rarely or never) and interaction patterns (e.g. which actions often, rarely or never occur consecutively), the TouchGhost interface can propose an improved way of interacting by launching the appropriate guides. An implicit strategy may for example detect the repeated usage of single-touch interaction techniques. A user might always scale a picture with only one finger, while it would be more efficient to use both hands. By showing a TouchGhost guide, the interface informs the user of the alternative method.

An implicit strategy might not always be able to determine which guide to show. A strategy could attempt to help the user after an erroneous action, or after a delay in the user’s actions. However, it may be impossible to predict what a user is trying to achieve. In such a case, the strategy can recommend the most likely guides to the user.

In public spaces, a dedicated strategy can invoke the available TouchGhosts one after another when the system has been inactive for some time, showing the possibilities of the interface to bystanders. When an explicit strategy is being employed, it might even be useful to show a TouchGhost guide on how to activate TouchGhosts once a new user starts to interact with the system.

#### 5. TouchGhost example



**Figure 3. Illustration of a TouchGhost interface, showing how to interact with a picture collection using multi-touch interaction techniques.**

Figure 3 illustrates an example of a TouchGhost interface. We use an explicit strategy to make the TouchGhosts available: the user can call for help through an avatar (bottom left side of Figure 3). When the user touches the avatar, a help mode is activated

and a black translucent layer covers everything except for the avatar. The user is then asked to touch the part of the user interface on which she wants assistance. A menu displays all the related topics on that specific interface component. Once the user selects a topic, the associated visual guide is shown.

The visualization method of the guide in this example consists of animating two virtual hands, which control the user interface elements that are currently displayed on the screen. As a result, the guide shows the user exactly what she needs to do to achieve the task. In this particular example, the guide explains how to browse through a picture collection using multiple fingers: while holding the top picture with two fingers, the user can extrapolate a 'strip' by dragging the picture outwards with the second hand (Figure 3). While the guide is playing, the user can interrupt it at any moment by touching the avatar.

## 6. Conclusion and future work

This work presents *TouchGhosts*: visual guides embedded in a multi-touch user interface that demonstrates interaction techniques while the user is interacting with the system. It is a first step towards exploring a help mechanism for multi-touch user interfaces in public spaces. Our initial prototype makes use of an explicit strategy to trigger the *TouchGhosts*; the user simply touches an avatar to request assistance. We are currently exploring more advanced and implicit strategies, and we will assess the effectiveness of other approaches, such as measuring hesitations.

We also need to examine how to deal with a system that targets multiple users at once. Problems may arise when several users need different guides at the same time. For example, if the visualization is based on animated virtual hands, four or six hands may be demonstrating interaction techniques simultaneously, which can lead to cluttering and confusion.

We focus on multi-touch user interfaces, since there is an important need for these types of solutions (especially when targeting public spaces). However, we believe our approach can be suitable for other systems, such as tablet PCs and PDAs, since a *TouchGhost* helps users to discover touch-based interaction techniques that are not directly perceivable by the graphical presentation of a user interface.

## Acknowledgments

## 7. References

[1] J. Coutaz, "Meta-User Interfaces for Ambient Spaces" In *Proceedings of TAMODIA '06*, Springer, 2006, 1-15.

[2] T. Cuypers, J. Schneider-Barnes, J. Taelman, K. Luyten, P. Bekaert, "Eunomia: Toward a Framework for Multi-Touch Information Displays in Public Spaces." In *Proceedings of HCI 2008, the 22<sup>nd</sup> British HCI Conference*, 2008

[3] A. Demeure, G. Calvary, K. Coninx, "COMET(s), a software architecture style and an interactors toolkit for plastic user interfaces." In *Proceedings of DSVIS 2008*, Springer-Verlag, 2008, 225-237.

[4] G. Dworman, S. Rosenbaum, "Helping users to use help: improving interaction with help systems." In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, ACM, 2004, 1717-1718.

[5] T. Grayling, "A Usability Test of Two Browser-based Embedded Help Systems." In *Journal of the Society of Technical Communication*, vol. 49, no. 2, 2002, 193-209.

[6] R. J. Jacob, A. Girouard, L. M. Hirshfield, M. S. Horn, O. Shaer, E. T. Solovey, J. Zigelbaum, "Reality-based interaction: unifying the new generation of interaction styles." In *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, ACM, 2007, 2465-2470.

[7] H. Kang, C. Plaisant, B. Shneiderman, "New approaches to help users get started with visual interfaces: multi-layered interfaces and integrated initial guidance." In *Proceedings of the 2003 Annual National Conference on Digital Government Research*, ACM, 2003, 1-6.

[8] S.G. McLellan, A.W. Roesler, A.L. Elliot, "The Effect of Advice Message Location on User Performance." In *IEEE Transactions on Professional Communication*, vol. 39, 1996, 43-48.

[9] B. A. Myers, D. A. Weitzman, A. J. Ko, D. H. Chau, "Answering why and why not questions in user interfaces." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2006, 397-406.

[10] D.A. Norman, *The Design of Everyday Things*, New York: Doubleday, 1988.

[11] S. Palminter, J. Elkerton, "An evaluation of animated demonstrations of learning computer-based tasks." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Reaching Through Technology*, ACM, 1991, 257-263.

[12] M. Priestley, "Task oriented or task disoriented: designing a usable help web." In *Proceedings of the 16th Annual international Conference on Computer Documentation*, ACM, 1998, 194-199.

[13] B. Raskutti, I. Zukerman, "Generating Queries and Replies during Information-Seeking Interactions." In *International Journal of Human-Computer Studies*, vol. 47, 1997, 689-734.

[14] P. Sukaviriya, and J. D. Foley, "Coupling a UI framework with automatic generation of context-sensitive animated help." In *Proceedings of the 3rd Annual ACM SIGGRAPH Symposium on User Interface Software and Technology*, ACM, 1990, 152-166.

[15] D. Vogel, and R. Balakrishnan, "Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users". *UIST '04*.