

Semiotic analysis of multi-touch interface design: The MuTable case study

— [Source link](#) 

Jan Derboven, Dries De Roeck, Mathijs Verstraete

Institutions: Katholieke Universiteit Leuven

Published on: 01 Oct 2012 - International Journal of Human-computer Studies V International Journal of Man-machine Studies (Academic Press)

Topics: User interface design, User interface, Natural user interface, Interface metaphor and User experience design

Related papers:

- [The eleventh finger: levels of manipulation in multi-touch interaction](#)
- [The Semiotic Engineering of Human-Computer Interaction](#)
- [Conceptual Integration and User Interface Metaphor for the Multi-Touch Control of Recorded Audio](#)
- [A user experience model for tangible interfaces for children](#)
- [Editorial: Developing, evaluating and deploying multi-touch systems](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/semiotic-analysis-of-multi-touch-interface-design-the-5fm35pqd6b>

Semiotic analysis of multi-touch interface design: The MuTable case study

Jan Derboven^{a,*}, Dries De Roeck^{b,1}, Mathijs Verstraete^a

^aCentre for User Experience Research (CUO), IBBT-K.U. Leuven Future Health Department, Parkstraat 45 Bus 3605, 3000 Leuven, Belgium

^bArtis University College of Antwerp, Department of Design Sciences, Mutsaardstraat 31, 2000 Antwerpen, Belgium

Received 28 July 2011; received in revised form 8 March 2012; accepted 18 May 2012

Available online 1 June 2012

Abstract

Although multi-touch applications and user interfaces have become increasingly common in the last few years, there is no agreed-upon multi-touch user interface language yet. In order to gain a deeper understanding of the design of multi-touch user interfaces, this paper presents semiotic analysis of multi-touch applications as an interesting approach to gain deeper understanding of the way users use and understand multi-touch interfaces. In a case study example, user tests of a multi-touch tabletop application platform called MuTable are analysed with the Communicability Evaluation Method to evaluate to what extent users understand the intended messages (e.g., cues about interaction and functionality) the MuTable platform communicates. The semiotic analysis of this case study shows that although multi-touch interfaces can facilitate user exploration, the lack of well-known standards in multi-touch interface design and in the use of gestures makes the user interface difficult to use and interpret. This conclusion points to the importance of the elusive balance between letting users explore multi-touch systems on their own on one hand, and guiding users, explaining how to use and interpret the user interface, on the other.

© 2012 Elsevier Ltd. All rights reserved.

Keywords: Multi-touch; Tabletops; User studies; Semiotics; Evaluation

1. Introduction

In the last few years, much research effort from both the industry and the academic world has gone to the design and development of multi-touch user interfaces (UIs). A great deal of research has been done into specific areas of multi-touch interaction, such as gestures (Wobbrock et al., 2009), multi-user interaction (Pinelle and Gutwin, 2008; Tuddenham and Robinson, 2009), and multitouch affordances (Piper and Hollan, 2009). However, although some first steps towards theorising and guideline creation have been made (e.g. Apple Inc., 2011; Wigdor and Wixon, 2011), there is no agreed-upon, unified multi-touch UI language yet. This lack of a clear gestural UI language is also brought forward by several authors (Norman and

Nielsen, 2010; Wigdor and Wixon, 2011), Don Norman and Jakob Nielsen stating that ‘these interaction styles are still in their infancy, so it is only natural to expect that a great deal of exploration and study still needs to be done’ (2010, p. 49). This lack of standardisation makes research into the usability and likeability of multi-touch UIs, gestures, and multi-touch widgets very valuable, as it can help determine which interactions work best on multi-touch devices.

Numerous publications describe usability and user experience evaluation as a part of the development process of multi-touch technology. User studies and user evaluation are a common part of the assessment of an interactive system, whether the system introduces new input methods (Spindler et al., 2009; Weiss et al., 2009), new visualisations (Wigdor et al., 2009), or uses existing technology to create novel applications (Apted et al., 2006; Piper et al., 2006). Research methodologies used for designing and studying multi-touch interaction include user observation (Derboven et al., 2010; Peltonen et al., 2008), collaboration usability analysis (Pinelle

*Corresponding author. Tel.: +32 16 32 32 09; fax: +32 16 32 32 10.

E-mail addresses: jan.derboven@soc.kuleuven.be (J. Derboven), dries.deroeck@artesis.be (D. De Roeck), mathijs.verstraete@soc.kuleuven.be (M. Verstraete).

¹Tel.: +32 3 205 61 86.

and Gutwin, 2008), and co-design (Mazalek et al., 2009). This paper, however, approaches multi-touch interaction from an alternative point of view: that of semiotics. Using semiotic theories, in particular the one put forward by De Souza and her Semiotic Engineering Research Group (SERG), this paper argues that semiotic analysis can offer novel insights in the evaluation of new interface paradigms such as multi-touch interaction. As a case study, the multi-touch user interface developed in the MuTable project is analysed for the way in which it communicates meaning to the user. Specifically, we use De Souza's Communicability Evaluation Method (CEM), arriving at an in-depth semiotic analysis of the MuTable interface by tagging and interpreting low-level user-system communication issues. From this high-level UI profile, some implications on the communicability of multi-touch systems in general will be derived.

This paper is structured as follows. The next section offers an introduction into semiotics, and how it can be applied to human-computer interaction (HCI). It continues with an in-depth discussion of the CEM methodology, and its application to multi-touch applications. Section 3 presents the results of a case study: CEM was applied to the MuTable multi-touch application platform. Section 4 offers a discussion of the case study results, and the appropriateness of the CEM framework for the evaluation of multi-touch interaction. The final section offers some concluding remarks, and points to further work in the area of semiotic analysis.

2. Semiotics

2.1. Semiotic analysis

Semiotics is often defined as the study of sign systems (O'Neill, 2008). More specifically, it can be said that semiotics is a theoretical framework that studies signs and sign processes (such as communication, signification, metaphor) by using 'natural language as a model for many other forms of communication, spreading the strategies of linguistic description to phenomena other than human language' (Bardzell and Bardzell, 2008, p. 2469). Derived from the theories of the Swiss linguist Ferdinand de Saussure and the American philosopher and logician Charles Sanders Peirce, semiotics has been used to study most of the major media forms, including literature, painting, photography, film and architecture (O'Neill, 2008). Besides these media forms, semiotics has also been used in computer science and HCI since the 1990s. One influential approach is that of Peter Bøgh Andersen, who has developed a design method based on a semiotic analysis of the work environment, specifically of the language used in that environment (Andersen, 1990). A summary of various other approaches to semiotics and HCI can be found in the 2001 special issue of Knowledge-Based Systems compiled by the contributors to a semiotics workshop at CHI 2000 (De Souza, 2001). In the Knowledge-Based Systems special issue, topics vary from computer

navigation (Benyon, 2001) in what is called 'Navigation of Information Space' to incorporating insights from older media forms in HCI (Andersen, 2001). Furthermore, since the CHI 2000 workshop, changing technology and interaction has inspired other semiotic approaches to ubiquitous computing and embodied interaction (Kjeldskov and Paay, 2010; O'Neill, 2008).

In addition to the semiotic approaches outlined above, Semiotic Engineering can be described as an alternative theory of HCI, rooted in the Anglo-Saxon semiotic tradition of Charles Sanders Peirce. Developed by Clarisse Sieckenius De Souza and the SERG research group, The Semiotic Engineering theory has been developed since the 1990s; the first comprehensive presentation of the theory was presented in 2005 (De Souza, 2005). Starting from the study of signs and sign processes outlined above, the specific contribution of Semiotic Engineering is the way it characterises technology applications as metacommunication artifacts, in which user interfaces are seen as 'one-shot, higher-order messages sent from designers to users' (Prates et al., 2000a, p. 308). In this view, the user interface speaks for the designers, in the sense that the interface contains all the meanings that interface designers have embedded in them, and that users should interpret in order to understand and use the application. The Communicability Evaluation Method is part of this Semiotic Engineering framework: it offers an analytic framework to analyse data from user test sessions, focussing especially on the relationships between meaningful elements in the interface, and investigating how users perceive these relationships, and how they respond to them.

2.2. Semiotic analysis of multi-touch interfaces

Multi-touch interfaces are particularly interesting for a semiotic analysis, since they are often considered to be Natural User Interfaces (NUIs) (Wigdor and Wixon, 2011; Wixon, 2008). The NUI paradigm is a user interface paradigm that goes beyond the point-and-click, metaphor-based interfaces of GUIs. The objective of NUIs, as described by its advocates, is to deliver intuitive, seamless experiences that unfold through natural human input (Wixon, 2008). This means that the indirect input of graphical user interfaces using the WIMP paradigm will be replaced by speech, touch and gesture—specifically touch and gesture in the case of multi-touch interfaces. Especially in the early days of NUI, 'naturalness' seemed to imply that the content itself would serve as the interface: Microsoft's NUI framework states that users interact in a direct, 'unmediated' way with the content, instead of needing a separate user interface with metaphors and icons to access the content (Wixon, 2008). Given this different approach to user interface design in NUIs, they are particularly interesting for in-depth semiotic analysis, as this analysis can shed a new light on current challenges in multi-touch user interface design.

Given the initial idea that in NUIs, the content itself serves as the interface without needing a separate user interface with metaphors and icons (Wixon, 2008), it would be difficult to use semiotic analysis to research multi-touch interfaces, as semiotic analysis – and Semiotic Engineering in specific – focusses on the system’s interface as a communication channel in itself: messages are sent from the designers of a system to the users, *through* the system’s interface (De Souza, 2005). If NUI interfaces are unmediated, users can manipulate objects directly without mediation of a user interface, which would make a semiotic analysis of that interface impossible. However, concerning this issue of unmediatedness, opinions differ. Saffer (2008) argues that ‘metaphor will always play a role with our devices—it is impossible to use or understand them otherwise.’ In the same spirit, Microsoft detracts from their ‘unmediatedness’ claim by proposing that the NUI’s ‘icons’ are representations ‘in which a portion of the object stands for the object itself’ (Wixon, 2008). This phrase in itself describes a kind of mediation: it can be seen as a short, simple definition of metonymy.² Metonymy, in its turn, is often regarded as a special form of metaphor (Chandler, 2002). These discussions on the intuitiveness and the direct manipulation of objects in NUIs and multi-touch interfaces make investigating multi-touch interfaces from a semiotic viewpoint especially interesting, as semiotics focusses explicitly on the interface as a ‘medium’ that communicates meaning. For instance, semiotics can be useful in the controversies on the use of ‘skeuomorphic’ interfaces—see for instance Pavlus (2010). Blackwell (2006) defines skeuomorphism as ‘the preservation in new products of non-functional features from previous generations’ (p. 498)—in HCI, skeuomorphic interfaces can be understood as UIs mimicking real-world objects with high fidelity. Although outside the scope of this paper, semiotic analysis could contribute to the discussion whether or not it is a good idea to use such real-world representations (as used in numerous iPad apps, e.g. the wooden bookshelf in the iBooks app, or the Contacts app, which looks like a book) to communicate meaning and functionality to the users. In this way, semiotics can shed new light on the (un)mediatedness, intuitiveness, and the often-discussed ‘naturalness’ (Norman, 2010) of multi-touch interfaces.

Besides the ‘naturalness’ of multi-touch interfaces, there are several other issues in which semiotic analysis can be helpful. One of these issues is the need for user assistance on multi-touch applications. User guidance is, especially in multi-touch environments, often regarded as unnecessary and merely a quick fix for a poorly designed application (Vinh, 2011). While exploring a system, users should be able to find out which functionality is available with as little user assistance as possible. This view is complementary to

the idea that multi-touch interfaces should be ‘natural’, self-explaining and intuitive. However, this idealistic view often contrasts with the way users actually explore and use multi-touch interfaces (Westerman, 2008). Semiotic analysis is well positioned to contribute to this issue of whether, or to what extent designers should offer explicit user assistance in multi-touch user interfaces.

2.3. Communicability evaluation

This paper focusses on the use of the Communicability Evaluation Method as a semiotic analysis method to evaluate multi-touch interfaces. CEM is a specific qualitative method that analyses user interfaces based on a semiotic interpretation of user test sessions (De Souza and Leitão, 2009). CEM primarily focusses on interaction problems, and enables researchers to identify and classify user problems in a fine-grained analysis that adds important nuances to the analysis that would otherwise risk being overlooked. For instance, user problems due to insufficient system feedback are broken down further by CEM into several subcategories, such as users misinterpreting the design solution presented by the interface, users completely unable to make sense of interface icons, etc. In this way, CEM presents an important added value to the other evaluation methods, in that CEM’s detailed assessment of user problems provides designers with a wealth of detailed information, which can help in making informed decisions during an application redesign. It is also this level of detail that proves to be especially interesting in evaluating innovative interface paradigms, such as multi-touch interaction. While CEM does go into great detail in the specifics of interaction problems, the method is not suitable to discover other specific types of information. For instance, CEM does not address e.g. problems in task or user modelling (Prates et al., 2000a,b), and it does not highlight positive, particularly well-functioning aspects of the interface: it is an evaluation method that offers a detailed analysis of user interface problems. The method specifically targets interaction problems, or, in Semiotic Engineering terminology, communicative breakdowns between the user and the system’s interface.

De Souza and Leitão (2009) mention the necessity of triangulating scientific research results from CEM with the other complementary methods. One other Semiotic Engineering method that CEM can complement is the Semiotic Inspection Method (SIM—De Souza et al. (2006)). The semiotic inspection method is a phased method through which researchers can analyse the meaningful elements in a system’s user interface. The ultimate goal of the semiotic method is to reconstruct the entire message the designer wants to communicate through the interface (De Souza, 2005) through a thorough inspection of the UI and its components. De Souza (2005) describes how the Semiotic Inspection Method (SIM) and the Communicability Evaluation Method (CEM) can be complementary research methods, and work together in order to create an in-depth user interface analysis. However, since SIM is targeted at

²Merriam-Webster (2011) defines metonymy as ‘A figure of speech consisting of the use of the name of one thing for that of another of which it is an attribute or with which it is associated (as “crown” in “lands belonging to the crown”).’

reconstructing the designer’s intended message from the interface, and technology is often evaluated by the designers that designed the application’s interface *themselves*, a SIM analysis does not always make much sense, as the ‘designer’s intent’ is well-known to the evaluators. Apart from SIM, CEM can also be used as a complement to other methods for user experience design and evaluation outside the Semiotic Engineering paradigm. While De Souza and Leitão (2009) explicitly mention this possibility, they only mention ethnographic research methods as possible complementary methods.

As an evaluation method, CEM has already been elaborated on in several publications (De Souza, 2005; De Souza and Leitão, 2009, Prates et al., 2000b), and it has been employed numerous times to evaluate communicability, especially of desktop graphical user interfaces (GUIs)—for case studies, see for instance (Prates et al., 2000a, 2001). To date, the interfaces evaluated in communicability evaluation case studies make use of WIMP (Windows Icons Menus Pointers) interfaces. This paper applies CEM to multi-touch user interfaces, taking the Semiotic Engineering method beyond traditional GUI interfaces to multi-touch interfaces. While most of the tabletop evaluations described in the literature offer some insight into low-level design and usability issues, CEM offers a semiotic framework to process lower-level observations, and draw more generalised conclusions in order to determine an overall semiotic application profile. This process from low-level observations to high-level

semiotic profile consists of three stages: *tagging*, *interpretation* and *semiotic profiling*.

2.3.1. Tagging

The first step in the Communicability Evaluation Method is to tag all problems users encounter according to a predefined coding scheme. In Semiotic Engineering, the meaning a user interface conveys to its user, is seen as metacommunication: through the interface, it is actually the *designer* who sends a message to the user. In that perspective, the designers are ‘actually communicating with users at interaction time’ (De Souza and Leitão, 2009), making the interface message the designer’s deputy: the interface ‘contains all the meanings [...] that the designers have rationally chosen to incorporate in the application’ (De Souza, 2005, p. 24).

In order to identify all communicative breakdowns between the user and the designer (i.e., instances where users fail to receive the communication as it was intended by the designers), all test participant actions in a user test need to be recorded and tagged according to the 13 tags proposed by De Souza and Leitão (2009). These tags are expressions that users might utter during problems they encounter during the test. De Souza (2005) describes this process as “‘putting words in the user’s mouth,’ in a kind of reverse protocol analysis’ (p. 126). Table 1 lists the tags used, and what they mean in terms of human–computer communication.

Table 1

Communicability Evaluation tags. The tag on a grey background has been added by the authors for specific multiuser interaction on a multi-touch screen (see Section 2.3.4 adjustments for multi-touch applications for a detailed justification of this addition).

Categorisation	Distinctive feature	Tag
Complete failure	User is unconscious of failure, and <i>erroneously thinks the goal has been achieved</i>	Looks fine to me
	User is <i>conscious of her failure</i>	I give up
Partial failure	User <i>does not understand</i> the design solution, and <i>tries out other ways</i> to achieve the goal	I can do otherwise
	User <i>understands the design solution</i> that is offered, but <i>consciously tries to find another way</i> to achieve the goal	Thanks, but no, thanks
Temporary failures User’s sense making is temporarily halted...	...because she cannot find the appropriate expression for her intended action; i.e., she <i>cannot find the interface element</i> representing the action/content she is looking for	Where is it?
	...because she does not see or understand the designer’s deputy’s communication; i.e., she <i>does not notice that her actions did have an effect</i>	What happened?
	...because she cannot find an appropriate strategy for interaction; i.e., <i>she does not know which step to take next</i>	What now?
	...because she interprets the interface’s reaction as unexpected, while the reaction is <i>actually the result of an action initiated by another user</i>	Who did that?
User realizes her intended interaction is wrong...	...because it is uttered in the <i>wrong context</i> , e.g., in the wrong application mode.	Where am I? Oops!
	...because her <i>expression is wrong</i> , i.e., the user realises she made an error	I can’t do it this way
	...because a many-step conversation (such as a many-step wizard) has not caused the desired effects	What’s this?
User seeks to clarify the designer’s deputy’s intended signification	Through <i>implicit metacommunication</i> ; the user is left wondering what a specific interface part means	Help!
	Through explicit metacommunication; the user looks for <i>a manual or a help file</i>	Why doesn’t it?
	Through autonomous sense making; the user wonders why a specific interface part doesn’t react as expected, and, for instance, repeatedly tries again	

2.3.2. Interpretation

In the interpretative stage, the goal is to search for the main metacommunication problems between users and designers, i.e., the higher-level problems users have in understanding the designers' messages. In order to analyse this metacommunication, all tags are analysed and organised according to a number of different perspectives, such as the frequency and context of occurrence of each tag, and the existence of patterns in the sequences of tags (De Souza and Leitão, 2009). Through this analysis, several types of high-level issues can be identified, such as 'local' interaction issues vs. more general, strategic issues (e.g., related to the system's overall information architecture). In addition to this analysis, the tags can also be divided into three groups, determining the relative gravity of the problems. Tags can be divided into complete, partial and temporary communication failures (see also Table 1).

CEM prescribes that the low-level tags should be grouped in a higher-level taxonomy of interaction and usability issues. While CEM proposes its own classification, the specific taxonomy used is left open (Prates et al., 2000a) to other possibilities, such as the usability heuristics by Nielsen (see De Souza (2005)) or Shneiderman, or other, more domain-specific heuristics or guidelines. However, as there are no specific guidelines or heuristics for multi-touch interaction available, the authors chose to use the original CEM taxonomy (see Prates et al. (2000b) for a detailed discussion), with minor modifications. These modifications will be discussed in detail in Section 2.3.4—the final taxonomy is available in Table 2.

2.3.3. Semiotic profiling

In the semiotic profiling stage, as the final stage of CEM, the goal is an in-depth characterisation of the overall metacommunication of the interactive product. Based on the previous stages, semiotic profiling can answer

high-level questions on the way users interact and communicate with the computer system. For instance, the semiotic profiling will give insights into how well the designer's communication is being transmitted to the user, if the designer's solutions are consistent with the way users look at a given issue or problem. Overall, the semiotic profiling can be said to focus on the design vision behind the product, and how well this vision has been understood and accepted by the user (De Souza and Leitão, 2009).

2.3.4. CEM Adjustments for multi-touch Applications

The CEM tagging and interpretation taxonomy, as proposed by De Souza and Leitão (2009) and Prates et al. (2000a, 2000b) provides an extensive analytic framework. However, it is not completely tailored to the specific issues of multi-touch interfaces. Prates et al. (2000a) acknowledge that the 'method applies basically to single-user interfaces' (p. 37), leaving the possibility of introducing new tags and categories open for e.g. multi-user and artificial intelligence applications. In the tagging taxonomy presented in Section 2.3.1, such a multi-user tag extension has been made. When multiple, co-located users are using a multi-touch application simultaneously, specific interaction breakdowns can occur. To address these issues, the *Who did that?* tag has been added to the tagging taxonomy: the tag describes an interaction breakdown due to user confusion over an unexpected change in the user interface, while the change is actually the result of an action initiated by another user.

Apart from the specific focus on single-user interfaces of the original CEM framework, CEM also shows a 'WIMP bias' of sorts. While most of the tags can be interpreted in a more generic way, there are some tags that, starting from a WIMP point of view, focus on long, multi-step navigation paths (implying deep menu structures, and embedded

Table 2

Interpretation taxonomy. The original taxonomy based on Prates et al., 2000a and 2000b has been adapted slightly to fit the multi-touch evaluation setting (changes in grey). The *Gestures* category has been added, and tags can be attributed to more categories than in the original framework.

Tags	Taxonomy				
	Meaning assignment	Navigation	Task accomplishment	Declination or missing of affordances	Gestures
What's this?	<input checked="" type="checkbox"/>				
What happened?	<input checked="" type="checkbox"/>				
Who did that?	<input checked="" type="checkbox"/>				
Help!	<input checked="" type="checkbox"/>				
Why doesn't it?	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>
Where is it?		<input checked="" type="checkbox"/>			
What now?		<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>
Oops!	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>
Where am I?	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
I can't do it this way	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
Looks fine to me	<input checked="" type="checkbox"/>				
I give up			<input checked="" type="checkbox"/>		
Thanks, but no, thanks				<input checked="" type="checkbox"/>	
I can do otherwise				<input checked="" type="checkbox"/>	

functionalities: the *I can't do it this way* tag), or on the modality of user interfaces (applications offering separate modes in which specific functionality can be available or not: the *Where am I?* tag). These characteristics are explicitly targeted in the CEM method, but they are often not very relevant in multi-touch multi-user systems. Deep menu structures are generally avoided in multi-touch applications, and while every single screen in a multi-touch app can be seen as a separate mode (Apple Inc., 2011), these modes are generally quite distinct, and offer only a limited set of functionality, reducing the risk of confusion. While these WIMP-related tags can be expected to be less important in multi-touch applications, they were not removed from the taxonomy, to avoid overlooking these issues during analysis.

Because of CEM's taxonomy focus on specific 'WIMP-related' issues, it does not provide tags that reflect issues specifically related to multi-touch applications. For this reason, a specific multi-touch extension was added to the higher-level interpretation taxonomy in Section 2.3.2, in the form of a *Gestures* category. CEM does not include tags that focus on the input modality as such: it concentrates only on the interface that is visually presented, and not how it is manipulated (using your hands, a mouse, or another input device). In this respect, CEM is also biased by WIMP interfaces and desktop GUIs, not questioning the relation between interface and input modality. However, in multi-touch applications, the use and communication of gestures is an important issue. Gesture-related breakdowns can have various causes; examples include users not being able to find the appropriate gesture, and users trying out gestures that do not produce the intended effect in the interface. Instead of adding a series of gesture-specific low-level categories, a higher-level category was added to the interpretation taxonomy. This *Gestures* category distinguishes breakdowns concerning communication about *input methods* from breakdowns related to e.g. *meaning assignment* (interpretation of interface elements) and *navigation*. During analysis, this high-level category allows to attribute a diverse set of breakdowns to the communication about gestures the designers have embedded in the user interface. The *Gestures* category can be attributed primarily to the temporary failure tags *Why doesn't it?*, *What now?*, and *Oops!* (see also Table 2). These tags can cover a wide range of gestural issues: users not finding the right gesture (*What now?*), users realising they used the wrong interaction after seeing an unintended change in the interface (*Oops!*), and users not understanding why the interface does not react as expected (*Why doesn't it?*). The *Gestures* category was limited to these three tags: they are appropriate in a gesture-related context and cover a wide range of breakdowns.

A third, minor extension to the CEM framework consists of extending the *Navigation* category to include the *Why doesn't it* tag. The tags in the original *Navigation* category cover breakdowns that are likely to occur in a WIMP interface: users not finding the desired functionality

(*Where is it?*) and not knowing where to look next (*What now?*), users realising they are following the wrong navigational path (*Oops!*), etc. Extending the *Navigation* category to include the *Why doesn't it?* tag acknowledges the fact that navigation in multi-touch applications is less standardised than in WIMP interfaces. In this sense, *Why doesn't it* covers specific breakdowns that occur in the navigation of the user interface: users trying to open, close, or navigate between applications or screens, but failing to do so because they are not familiar with the navigation interaction.

3. Case study

3.1. MuTable

3.1.1. Concept

MuTable, the multi-touch application platform used as a case study in the remainder of this paper, is a collection of tabletop applications (widgets) which can be used in a variety of public spaces, including museums, public events, libraries, and schools. Depending on the context of use, specific widgets can be used to create a context-specific application. While many multi-touch applications typically focus on viewing video, browsing photographs and playing music (Apted et al., 2006; Kristensson et al., 2008; Maximo et al., 2009; Peltonen et al., 2008), MuTable focusses on productivity tools for collaborative content creation. The MuTable application widgets discussed in this paper were developed to allow adolescents to collaboratively use a tabletop for school-related work. Students can search for content (text documents, pictures, movies,...), and integrate this content in a presentation. The central use scenario integrates searching and browsing for content with the creation of new content.

MuTable was conceived to run on a multi-touch surface that is positioned horizontally, or at a small angle, convenient for people to work at while standing up. The MuTable applications were designed to run on a touch table with a resolution of 1280 × 800 pixels; the dimensions of the touch table used during testing were 110 cm × 55 cm. These dimensions allowed two to four persons to collaborate at the MuTable multi-touch table comfortably, depending on the tasks to be completed. In addition, the applications were designed to have users standing at one particular side of it, unlike some other multi-touch hardware and software designed specifically for access from all four sides of the surface (e.g. Microsoft Surface). This consideration also played its part in UI design, as the software was designed specifically with users in mind standing in one specific orientation at the table. This setup makes the UI design of the MuTable software particularly suitable for touch tables that are placed against a wall, or are wall-mounted. Depending on the use context, the table can be positioned at a small angle in one direction, making the table position ergonomically better to work at compared to a flat surface. Moreover, the MuTable design specifically

incorporates a full-screen presentation mode in which slides created on the table can be presented. This functionality is especially suitable for wall-mounted tables that can be tilted from a small angle (in ‘working mode’) to a vertical position, to serve as a presentation surface (‘presentation mode’).

3.1.2. User interface

The MuTable UI offers a number of small, separate, multi-touch application widgets. These widgets can be accessed from the main ‘central ball’ menu, that allows users to search, open and delete content, and to open a Create submenu, from which a typewriter tool, a presentation creation tool and a drawing tool can be opened (see Fig. 1). The widgets, used together, can be used create and search for content, and (re-)use that content in a presentation. Several widgets can be opened at the same time, supporting collaborative work: when collaborating, one user can, for instance, search for content while a second user is typing a text in another widget. In this way, MuTable especially facilitates collaborative research on a specific subject, and creating new content (text, a presentation) on that subject. Therefore, the platform is especially useful in educational or library contexts, although

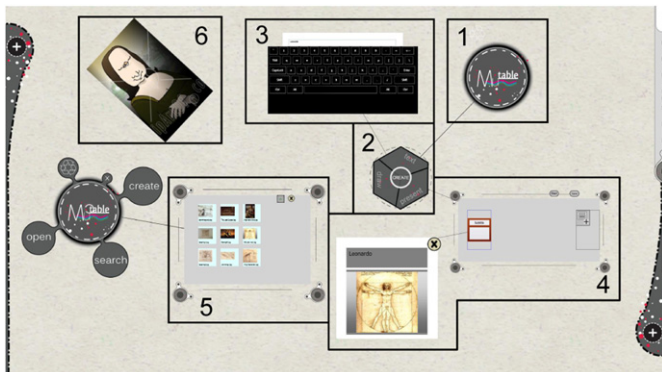


Fig. 1. Main MuTable functionality. The functionality includes navigation elements and several widgets: 1a. the central ball menu (collapsed), 1b. the central ball menu (expanded), 2. the Create submenu, 3. the typewriter tool, 4. the presentation creation tool (with a slide opened), 5. a file browser tool, 6. a piece of content (picture) opened.

the (individual) widgets are also relevant for other contexts (searching for content, e.g., can be relevant for both event and museum contexts). Depending on the context and availability of content the functionality of the platform can be changed.

The interaction techniques used to manipulate the application widgets are intentionally kept as simple as possible, and used across all MuTable widgets. Fig. 2 presents the most important interaction techniques used in the MuTable interface. Widget handles are used to allow users to *interact with application widgets as a whole* (Fig. 2a; see also Nacenta et al. (2009) and Schneider et al. (2010) for more information on gestures differentiating interaction with widgets as a whole vs. widget components). *Dragging content* from one widget to another is done by touching the content with two fingers, and moving one or both of them to the place where the object should be dropped (Fig. 2b). *Flipping pages* in books can be done by touching the book with three fingers, with two fingers staying in place, and one moving to the left to see the next page, or to the right to see the previous page (Fig. 2c). *Cropping images* is done by touching them with four fingers. A rectangle in the image gives users visual feedback showing the area to be cropped, and allows users to change the area by moving their fingers—releasing them creates an image of the cropped region (Fig. 2d).

3.2. Test procedure

The research goal of the user sessions was twofold: to assess the overall usability of the MuTable application framework, and to assess the interpersonal interaction and collaboration at the multi-touch table. To this end, 16 adolescents aged 16–18 were recruited (9 female, 7 male participants) to test the MuTable framework. They participated in small groups (2 groups of 3, 5 groups of 2), in which all members of the groups already knew each other before the test: they were friends, classmates, or family. The participants were asked to work at the interactive table, their task being to create a presentation in 45 min on the life and works of Leonardo da Vinci. One of the test

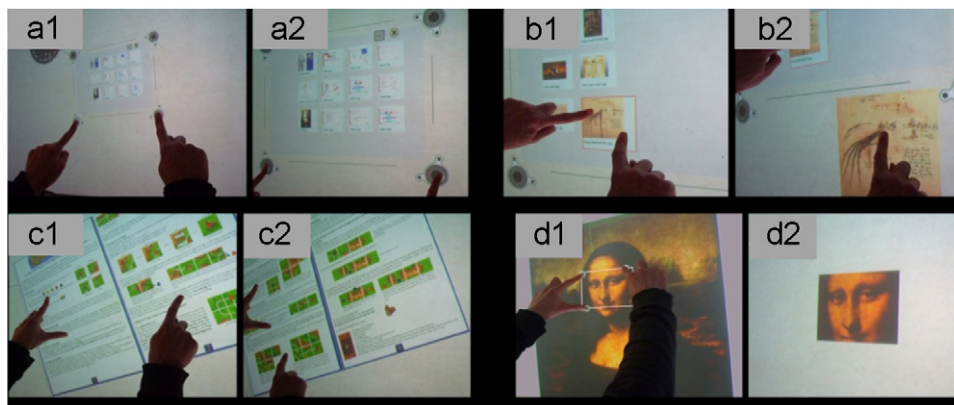


Fig. 2. MuTable interaction techniques. (a) Whole widget manipulation, (b) Drag and drop, (c) Flip pages and (d) crop.

observers gave the assignment to the participants, in which the participants were given a single test task: creating a presentation for a travelling exhibition on Leonardo da Vinci, using the functionality and content available in the multi-touch system. The assignment was quite ‘open’, in that no explicit, ordered breakdown of partial tasks was offered: test participants were asked to explore the system themselves. Additionally, no explicit think-aloud protocol was used, as the discussions among the test users provided ample feedback. Similar user tests with co-participating test users have been described in literature (Kahler, 2000; Shrimpton-Smith et al., 2008).

The user tests consisted of some ten minutes in which the participants were allowed to explore the table themselves. Afterwards, all MuTable functionality was demonstrated to the test participants in a short presentation. Pilot testing pointed out that the short demonstration was necessary, because test participants were not able to discover all gestures and functionality by themselves. (This observation in itself is, of course, highly relevant to the communicability analysis of the user interface. This issue will be dealt with in more detail in Section 3.3.2) After the short demonstration, the test users were allowed to continue working on their presentation for 30 min. After completing the test assignment, participants presented the work they created. Before starting the test, participants were given a pre-test questionnaire asking about their background; after the test, they were given a post-test questionnaire asking how the assignment went. At the end of the test session, the participants received a reward, in the form of a €20 gift voucher.

For analysis purposes, the test sessions were recorded in a usability lab. The video tapes were reviewed several times, in a qualitative usability evaluation (focussing on e.g. table division between test participants and territoriality issues, see Derboven et al. (2010)). For a detailed analysis of the usability and clarity of the interface, the CEM methodology was used. For this analysis, the test videos were treated as described in Sections 2.3.1–2.3.3: the three CEM stages were executed, with each level reaching higher levels of abstraction. Where relevant, the next section will show where the CEM semiotic analysis provided additional detail, compared to the qualitative usability evaluation.

3.3. Results

3.3.1. Interpretation of individual tags

In the first step of CEM, tags were attributed to the test videos of all seven groups. Although some groups had more tags of a specific type than other groups, the general picture was stable: a few tags were given very often, while others were given only a few times. A tag summary of all test groups is visualised in Table 3. We will, however, not use these results to draw statistically relevant conclusions, as the sample size is quite small (seven test groups), and CEM is a primarily qualitative method (De Souza and Leitão, 2009). The results, however, clearly show a number of trends that will be discussed in the next paragraphs.

Table 3

Cumulative summary of tags. The numbers of occurrences are cumulative numbers, adding the occurrences of all seven test groups.

Tag	Number of Occurrences	Percentage (%)
What's this?	21	9.3
What happened?	18	7.9
Who did that?	3	1.3
Help!	2	0.9
Why doesn't it?	83	36.6
Where is it?	16	7
What now?	8	3.5
Oops!	25	11
Where am I?	2	0.9
I can't do it this way.	5	2.2
Looks fine to me.	11	4.8
I give up.	9	4
Thanks, but no, thanks.	10	4.4
I can do otherwise.	14	6.2
Total	227	100

Table 3 clearly shows that a limited number of tags occurred very often. The top three tags that were attributed most often, all occurring more than 20 times, were: *Oops!*, *What's this?*, and *Why doesn't it?*—all temporary failures (see Table 1).

The *Oops!* tag occurred 25 times in total. Prates et al. (2000b) describe this tag as: ‘a user performs some action to achieve a specific state of affairs, but the outcome is not what he expected’ (p. 34). The user, in other words, makes a mistake, but immediately realizes his mistake. The occurrence of this tag was quite related to the *I can't do it this way* (5 occurrences) and the *What's this?* (21 occurrences) tag. These tags all relate to the expectations users of the system have. While the *Oops!* and the *I can't do it this way* tags refer to cases where the users realise their intended interactions were wrong, the *What's this?* tag refers to users not knowing what to expect (see also Table 1). These tags all relate to the same characteristic of the MuTable interface: the fact that the interface offers little explicit user guidance, but leaves the functionality to be explored by the end user. While exploring a multi-touch system can be fun and instructive, this enjoyable exploration can turn into frustration when users cannot find the functionality or content they are looking for. In effect, during the MuTable tests, users did get frustrated to some extent after prolonged searches for a specific functionality.

The exploration also triggered the occurrence of another tag: *Thanks, but no, thanks*. In exploring and trying out functionalities, users sometimes invented unintended solutions to their problems, leaving the intended functionality unused. An example of this is the use of the ‘recycle bin’ to delete objects: some users did not like this functionality, and invented other ways of getting rid of content. Users resized the objects until they became unnoticeably small, or simply moved them across the borders of the multi-touch table, so they were still open, but not ‘in view’. As these unintended solutions from the *Thanks, but no, thanks*

tag were quite surprising to the designers of the interface, they were also uncovered by the qualitative usability study (see also Derboven et al. (2010)).

The *What's this?* tag occurred 21 times in total. Prates et al. (2000b) describe this tag as ‘The user seems to be exploring the possibilities of interaction to gain more (or some) understanding of what a specific function achieves. He lingers on some symbol waiting for a tool tip and/or explicitly calls for help about that symbol, or he hesitates between what he thinks are equivalent options’ (p. 34). In other words, the user expects an explanatory tip about what he sees in the interface. The fact that users seem to expect explanations points to the fact that the MuTable interface offers too little (textual) explanation. While the interface does offer icons and symbols to explain the functionality, it is apparent that this does not suffice. To allow novice users to use the MuTable widgets fluently, more explicit user guidance is needed.

The *Why doesn't it* tag occurred 83 times in total. Prates et al. (2000b) describe this tag as ‘the user expects some sort of outcome but does not achieve it. The subsequent scenario is that he then insists on the same path, as if he were so sure that some function should do what he expects that he simply cannot accept the fact that it doesn't’ (p. 34). While the number of 83 occurrences seems extremely high compared to other tag occurrences, this picture has to be nuanced. The repetitive behaviour related to the *Why doesn't it* tag was observed in three different contexts: these contexts are represented in the higher-level taxonomy used to categorise the tags in (see also Section 2.3.2). Situation 1 is a situation in which a specific problem occurred in the navigation of the user interface, in which the user did not achieve the expected outcome in trying to open or navigate between widgets. An example of situation 1 is when a user did not notice a widget was already open, and tried to open a second instance of that widget in vain. Situation 2 occurred when a gesture problem occurred, with no, or an unexpected outcome. An example of situation 2 is a user trying to move text from the typewriter to a slide, not using the correct gestures. Situation 3 occurred when the user's expectations about a sign in the interface were violated, and the users could not understand why the function did not work the way they wanted to (which amounts to a problem in the assignment of meaning to the interface). Situation 3 occurred when, for example, a user tried to move an image to a presentation template intended for text, not for images. While this subdivision brings down the occurrence numbers, one clear peak remains (see Table 4): *Why doesn't it* tags related to gesture issues still had 54 occurrences. These figures clearly indicate that the widgets suffered from a gesture usability problem, in Section 3.3.2, we will come back to these gesture-related usability problems in more detail.

Apart from looking at the tags which were used most frequently, it is also useful to look at the tags which were used the least. The least-used tags were the *Who did that?*, *I can't do it this way*, *Where am I?*, and *Help!* tags—all temporary failures (see Table 1).

The *Who did that?* tag was added by the authors, in order to account for problems that can be encountered in

Table 4

Why doesn't it subdivision. The *Why doesn't it* tag was attributed in three different circumstances.

Interpretation taxonomy	Number of occurrences	Percentage (%)
Gestures	54	65
Meaning assignment	16	19.3
Navigation	13	15.7
Total	83	100

multi-user settings. The *Who did that?* tag did not occur frequently in the test results, but was necessary to account for the few occasions where an action's outcome was not near to the user's fingers, and where confusion arose about which user was responsible for a certain outcome. As most system feedback was located close to the user's initial action, the *Who did that?* tag did not occur very frequently.

The *I can't do it this way* tag occurred only 5 times. Prates et al. (2000b) describe this tag as ‘Sometimes the user follows some path of action and then realizes that it's not leading him where he expected. He then cancels the sequence of actions and chooses a different path. In this case the associated utterance is *I can't do it this way*’ (p. 34). The close relationship between the *I can't do it this way* and the *Oops!* tag have already been discussed before: while this tag focusses on long navigation paths, the MuTable framework, making use of several small widgets, was conceived in a way that stimulates direct interaction with the widgets, and avoids long navigation paths. Therefore, the *I can't do it this way* tag occurred very infrequently: if the user made a mistake in the MuTable widgets, they simply touch another object or window to continue working, without having to abandon a multi-step navigation path. The *Oops!* tag can be seen as an alternative to *I can't do it this way*, as it describes a similar problem (the user realizing the intended interaction is wrong; see also Table 1), but targets momentary mistakes, while *I can't do it this way* targets longer navigation paths.

The *Where am I?* tag occurred only 2 times. Prates et al. (2000b) attribute this tag when: ‘the user performs some action that is appropriate in another context but not in the current one’ (p. 34). This tag hardly occurred for similar reasons the *I can't do it this way* tag did not occur frequently: the MuTable widgets are very basic in their functionality. As the widgets do not have elaborate functionality, long menu lists and different modes, the user does not really risk interpreting icons and signs in the wrong context.

De Souza and Leitão, 2009, in their description of CEM, also mention the order in which tags occur as potentially important information for interpretation. However, apart from the relation between the *Oops* tag and the *What's this* and *I can't do it this way* tags discussed earlier, it proved to be very difficult to interpret tag orders. This was due to the fact that the MuTable framework is a multi-user framework, and that the users often tended to switch in their

Table 5

Tag occurrences before and after explanation of the functionality. The tag occurrences are cumulated according to the interpretation taxonomy.

Tag	Number of occurrences before explanation	Percentage (%)	Number of occurrences after explanation	Percentage (%)
Meaning assignment	54 (0.82/min)	47.3	32 (0.14/min)	24.8
Navigation	15 (0.23/min)	13.2	9 (0.04/min)	11.5
Task accomplishment	9 (0.14/min)	7.9	20 (0.09/min)	17.7
Declining/missing affordances	8 (0.12/min)	7.0	12 (0.05/min)	10.6
Gestures	28 (0.42/min)	24.6	40 (0.18/min)	35.4
Total	114 (1.72/min)	100	113 (0.50/min)	100

collaboration methods. Users changed constantly between individual work, collaboration, watching another user work, etc. This constant switching of collaboration mode made it difficult to identify patterns in the users' efforts to use the widgets. For a detailed description of different modes of collaboration, see also Derboven et al., (2010). For this reason, the tag orders are not included in the further analysis.

3.3.2. Interpretation of high-level taxonomy

As already mentioned in Section 3.2, the fact that after some ten minutes the users were interrupted by the test facilitators in order to give a short explanation about the MuTable functionality is highly relevant in relation to the test results. During pilot test sessions prior to the actual tests, it became clear that the user tests would not return valuable results if such an explanation was not given. Test users would not be productive enough to finish the test task in a satisfactory manner. In terms of semiotic tagging, this short explanation more or less is the equivalent of the 'Help!' tag. None of the test groups were able to find out sufficient information about the MuTable functionality in order to finish the test task, so each user group needed some form of 'metalinguistic metacommunication in order to restore productive interaction' (De Souza and Leitão's, 2009, p. 40 description of the Help! tag). This was clear even when the test users did not call for it explicitly. The necessity of the short explanation in the first part of the user tests reinforces the observation made in Section 3.3.2 that the interface in itself offers too little cues for users to make sense of the interface on their own. For further analysis of the user test data, it is important to make a distinction between tags attributed *before*, and the tags attributed *after* the explanation was given. Table 5 clearly shows that both before and after the explanation, meaning assignment and gesture problems were the main issues. Therefore, the next paragraphs will discuss the *Meaning Assignment* and *Gestures* categories in detail, starting with the *Gestures* category.

Although the gestures needed to navigate the interface were quite straightforward (see also Section 3.1.2), the tagging and categorisation results discussed in the previous section clearly show that this gesture set proved to be an

important hurdle to take for the users, even after³ an explanation concerning the functionality and gestures was provided. With 41 occurrences, gesture tags even become the most important problem that occurred after the explanation. This suggests that gestures that are more intricate than the already common pinching, flicking and swiping should be avoided, or used with care. The use of intricate gestures should, for instance, be limited to short-cuts for actions that are also available in other ways.

The gestures used to navigate the MuTable interface had somewhat mixed results. All of the test participant groups figured out the 'pinch to zoom' gesture on their own, before the short MuTable demo was given—even participants that had not used multi-touch technology before managed to find out about it. On the other hand, the more elaborate gestures proved to be quite difficult. When investigating the individual tags, it was found that different kinds of gestural issues could be distinguished. A first gestural issue is the use of specific, more elaborate gestures to perform an action (for instance, the gesture for dragging text from one widget to another). These more elaborate gestures proved to be difficult to remember and to perform correctly. Secondly, even simple gestures were often performed incorrectly. For instance, the 'pinch to zoom' gesture was often executed with two hands (multiple fingers per hand), instead of with two fingers. Thirdly, as the gestures were not visualised in the user interface, they sometimes *obscured* the availability of functionality. Participants sometimes tried out gestures to use functionality that was not available, and, the other way round, failed to discover functionality that was available. A fourth gestural issue amounted to a technical issue, in which users trying to drag an object from one place to another, performed the drag gesture too quickly, 'losing' the dragged object on the way, and sometimes dropping it in unwanted locations.

³The absolute numbers presented in Table 5 need to be nuanced: the occurrence numbers after explanation may seem high compared to the numbers before explanation, but this is due to the fact that users were allowed to work much longer *after* the explanation than *before*. To add this nuance to the data, mean tag occurrence has been added to Table 5. Overall mean tag occurrence is 1.72 tags per minute before the explanation was given, and 0.5 tags per minute after: after the explanation, user problems occurred three times less often on average.

While the last issue is a technical issue that could be solved using other hardware, the first three issues are inherent to gestural interfaces. The first two gestural issues suggests that it is imperative that gestures be very simple, making them easy to learn and memorise, as users forget about gestures even after they have seen a demo, or do not pay sufficient attention to the *exact* way of executing gestures. Ideally, gestures should allow for a margin of ‘error’, as users often are not very precise in the execution of gestures. The third issue suggests it should be made clear to users which functionality they can expect to operate using gestures, and which they cannot—gestures should enable users to use a system more fluently, they should not obscure its functionality.

While qualitative usability testing also uncovered the fact that a lot of user problems, both before and after the functionality explanation were gesture-related, the CEM high-level taxonomy analysis adds detail to this picture. It was expected that after the system’s functionality was explained, users would have less problems in using and understanding the basic functionality of the system (assigning meaning to interface elements, and interacting with the system, i.e., the *Meaning assignment* and *Gesture* categories), and more problems in combining basic skills to get their actual work done (*Navigation*, *Task Accomplishment*). In that view, the decline of *Meaning assignment* tags after the explanation of functionality was to be expected: as users know more about the system’s functionality, they can be expected to have less problems in assigning the correct meaning to interface elements. On the other hand, while gestures were also demonstrated, gesture issues make up the majority of user problems after the system explanation was given. While the share of *Meaning assignment* tags dropped 22.5% after the explanation, the share of *Gesture* tags rose 10.8%. This observation suggests that while the functionality explanation was effective in explaining the *visual* part of the user interface, users continued to struggle with the *gestural* part of the interface. In this way, CEM adds detail to the analysis, making a clear-cut distinction between basic visual interface issues (making sense of the interface cues) that were alleviated by the functionality explanation, and basic gestural issues that remained important problems.

Although dropping with more than 20% after the functionality explanation, the *Meaning Assignment* category is one of the important communication breakdown categories, apart from the *Gestures* category. The high initial frequency of the *Meaning assignment* category indicates that users struggled in assigning the correct meaning to visual interface elements in the MuTable framework. Comparing the *Meaning assignment* tags before and after functionality explanation (47.3%–24.8%) suggests that a limited explanation can suffice to allow users to be more effective in completing their tasks. While one of the focus points of the MuTable interface design was offering sufficient, but often subtle user cues and affordances in the interface (such as widget handles to

move and to resize widgets; text labels were avoided in general—see Fig. 1), it was apparent that users often overlooked these cues. The decline of *Meaning assignment* tags indicates that the user cues clearly were not explicit enough: users needed the explanation of the MuTable interface in order to complete their test assignment. This illustrates the difficulty of creating a clear UI without using text or descriptive labels. Icons, without the right context, are often hard to interpret. A standardised GUI environment, with well-known conventions and icons, is clearer in this respect (see De Souza (1993) for the importance of using conventional symbols in interface design): there are clear conventions on which meaning should be attributed to specific, ‘standardized’ icons. In a multi-touch environment, however, there is no coherent UI language available, making even (re-used) GUI conventions less clear-cut and ambiguous: during user tests, several users accidentally closed a widget by touching the cross in the upper right corner of the widget. While this is a very well-known GUI convention, users did not always expect the widget to close in the multi-touch environment. Some users need explicit guidance in making sense of the interface controls presented on a multi-touch user interface, even if the interface re-uses well-known GUI conventions.

3.3.3. Semiotic profiling

While the previous sections discussed the CEM tagging results at the level of the individual tags and tag taxonomy, the current semiotic profiling section reaches a higher level of abstraction. Semiotic profiling distills overall conclusions based on the previous tagging and interpretation phases.

Analysis of both the individual CEM tags (*Oops!*, *What’s this?*, *Why doesn’t it?*) as well as the higher-level categorisation (*Gestures*) indicate that the main issues in the MuTable interface were related to gestural interaction. A comparison between the tags *before* and *after* functionality explanation indicate that even a detailed demonstration of gestures does little to alleviate the difficulties users have with gestures that go beyond the standard pinching and flicking. This is mainly due to the lack of *visibility* of gestures (gesture availability is not signalled visually in the interface) and the lack of *precision* with which the gestures are executed (using multiple fingers to execute a two-finger gesture). While moving objects around and resizing them was quite straightforward, the interface does not provide user support for the usage of more intricate gestures.

Apart from gestural issues, the CEM analysis clearly showed that the interface design relies too heavily on implicit cues. Despite design decisions that were targeted at allowing users to explore the MuTable UI (e.g., moving parts in the interface that were designed to catch the users’ attention and be touched), the CEM results show that the UI suffers from a lack of user support. The fact that the *Meaning assignment* category was very prominent is mainly due to that lack of user guidance. Assigning the correct meaning to the MuTable interface components was not

straightforward for MuTable's users, as there is very little explanatory text in the interface. Even though the interface controls had been graphically designed to be as clear as possible (e.g., presentation slides are presented as sheets of paper with text placeholders), this graphic design often proved to be insufficient for the users to interpret the details in the intended way. While the basic message was clear (e.g., a presentation slide can be filled up with text or pictures), users could not infer the interaction details from the design (e.g., which slide template should be used for text, and which for pictures).

Combining the above observations on gestures and meaning assignment, CEM analysis suggests that the MuTable interface, at a superficial level, is quite easy to understand and 'natural' due to its use of simple gesture pinching and dragging gestures, and its reliance on familiar metaphors such as pictures and presentation slides. However, the details of the interaction (which more intricate gestures and functionality are available) are insufficiently communicated to users, and often need additional explanation. Superficially, however, the MuTable UI does lend itself for user exploration: using the system for the first time, users can spend some time exploring it, getting to know its functionalities before doing actual work or performing tasks. The analysis presented above suggests that the 'design for exploration' approach was, in fact, successful. The high occurrence of the *Oops!* tag, co-occurring with the *What's this* tag, suggests that users try out functionalities, often without being able to predict what the outcome of their actions will be, and without thinking they 'failed' when the outcome was not what they expected. This exploration also inspired the unexpected user behaviour as described in Section 3.3.1. A related question that requires extra research is when this exploration phase tips over to frustration because the desired functionality is not found, or not found fast enough. In the post-test questionnaire, test users did report that they liked using the multi-touch widgets, but during the tests, it was obvious that the line between exploration and frustration was crossed at some points.

4. Discussion

4.1. Case study: Implications for user guidance in multi-touch applications

The high-level semiotic profiling of the MuTable widgets suggests that although at a superficial level, multi-touch interaction with well-chosen gestures (selected with the social and interaction context in mind, see also Hinrichs and Carpendale (2011)) can be quite straightforward and 'natural', users need guidance in understanding the detailed interaction with the system. For walk-up-and-use multi-touch systems in public spaces, user exploration without any guidance can be a fun part of the application experience, when learning to use it. This points to an intricate balance between letting users explore multi-touch

systems on their own on one hand, and guiding users, explaining how to use and interpret the user interface, on the other. Nevertheless, we believe that guidance often is necessary to allow users to understand use all functionality, because of the lack of standardisation both in the UI language and in the use of gestures on the one hand, and because no interface design can be entirely unambiguous on the other.

Although help systems for gestural interfaces do exist, such as the experimental TouchGhosts interface (Vanacken et al., 2008), 'self-revealing gestures' in the form of tooltips (Brandl et al., 2008) or the often text-ridden instructional screens found in commercial iPad apps such as Project Magazine app (Virgin Digital Publishing, 2011), a good way to provide users with correct, non-intrusive instructions on which gestures to use has not yet been found. Existing systems tend to break the flow of use, or offer separate instructional screens that try to explain the entire system, but are not truly embedded in it. Based on the case study presented, a number of generic characteristics for a multi-touch user guidance system can be proposed. Such a system should:

- **Adapt to the user.** User guidance should be available but non-intrusive. More specifically, users should be allowed to explore the multi-touch system freely without being interrupted by user assistance. User exploration allows users to be creative in trying out the application, and potentially discovering new, creative solutions to their problems. However, while multi-touch applications are often conceived to offer limited functionality, and to be self-explanatory in their simplicity (Vinh, 2011), user guidance should be available whenever necessary. While multi-touch user interfaces can be designed to be as straightforward as possible, they still are the 'designer's deputy': they have to communicate the intended meaning to the users when they are interacting. If the initial communication breaks down, an alternative should be available. Users can encounter problems in unexpected ways, and when they do, users should be able to find the help they need, without hindering other users in multi-user environments.
- **Explain gestures.** User guidance should explain which gestures are used throughout the application. As most gestures can be performed in ways that are quite similar (e.g. dragging with one finger, multiple fingers, or an entire hand), the way of executing gestures should be dealt with in detail. In any case, the number of gestures used in the application should be kept to a minimum, and they should be kept as simple as possible. For designers and developers, the challenge is to be creative in the interface and interaction design with this very limited set of gestures. Whenever more intricate gestures are necessary, their use and execution should be explained in a non-obtrusive way, or an alternative interaction (using a more common interaction style, e.g. buttons) should be offered.

- **Explain functionality.** User guidance should explain which functionality is available. In general, the fact that multi-touch interfaces often use real-world analogies makes it possible for users to draw upon their real-world knowledge to discover high-level functionality—it is obvious that a typewriter can type text. Unfortunately, real-world analogies and metaphors inevitably break down at some point. At the point where the real-world analogy breaks down, the interface, as the designer's deputy, should become more explicit; i.e., when behaviour can no longer be inferred from real-world analogies, explicit communication about the nature of the functionality and the interaction should become more prominent. User assistance should provide clarity on the exact functionality provided by the application. The fact that the use of gestures tends to obscure functionality makes it even more important to be explicit about which functionality is available, and which is not.
- **Explain the UI language.** User guidance should explain which conventions are used in the visual language of the application. Since multi-touch applications cannot draw upon a rich tradition of graphical interface design such as GUIs, and even well-known GUI conventions tend to lose their familiarity in a new environment, it is important that the user interface is consistent in its behaviour, and that a user support system is explicit about which graphical elements are used, and what they mean (see also De Souza (1993)).

4.2. Communicability evaluation for multi-touch applications

As CEM has been used primarily as an evaluation method to evaluate WIMP user interfaces, it is worthwhile to evaluate the method's usefulness in evaluating other types of user interfaces. The CEM framework has been extended with multi-touch-specific additions: they were necessary to adjust the evaluation method to the specifics of multi-user, multi-touch applications such as MuTable. Especially the additional *Gestures* category was instrumental in analysing the difference between the effectiveness of additional explanation of the visual aspect of the user interface (*Meaning Assignment* category) and the gestural aspect of the interface (*Gestures* category)—see Section 3.3.2. Besides the *Gestures* category, the additional low-level *Who did that?* tag – although not very prominent in the case study analysis – was instrumental in analysing the user problems due to simultaneous, co-located work.

The analytic power of the CEM framework depends on the appropriateness of the low-level tags, and the interpretation taxonomy. The multi-touch case study results show that well-chosen extensions can allow for more meaningful analysis and relevant results. For other types of applications, other tags and interpretation categories will be more suitable: Prates et al. (2000b) offers the example of artificial intelligence, 'for which

utterances related to the system's cognitive abilities are likely to occur (e.g., *Do you know this? Can you learn this?*)' (p. 37, original emphasis). However, it is not always self-evident to anticipate beforehand which user problems 'are likely to occur', and need a new, separate tag, especially when applying the framework to new interface paradigms. This requires at least some experience with the technology before one can make informed extensions to the CEM framework. Less appropriate tags and interpretation taxonomies will significantly reduce the framework's analytic power, by failing to point out the important communicative breakdowns.

With the appropriate tags and interpretation categories, the important strength of CEM is that it reveals nuances in user behaviour that would probably remain unnoticed in more 'standard' usability testing. The exact nature of the users' mistakes can be determined in a more precise, fine-grained way. For instance, in our case study, the method has shown that the *Oops!* tag (users realising they made a mistake) was related to the *What's this?* tag (users wondering about the meaning of a certain interface component). This co-occurrence of tags lead to the observation that the high frequency of mistakes is due to the user's exploratory behaviour with easy recovery from mistakes, and not only from the lack of clarity or guidance in the user interface. In addition, CEM explicitly 'codifies' unexpected user behaviour in the *Thanks, but no, thanks* tag, in which users are aware of the 'standard' way of solving a problem, but consciously choose to use another solution. Especially in systems in which user discovery and exploration is an important factor, it is better to take up unexpected user behaviour in the basic analysis, rather than considering it as an anomaly.

On a higher level, CEM allows for well-motivated general interpretations of low-level problems. Instead of stopping at a discussion of individual incidents during user testing, CEM offers a solid framework to group individual incidents in higher-level categories, and arrive at more generic conclusions. For instance, the CEM high-level categorisation in the case study showed a clear difference between the high frequency of the *Meaning Assignment* and the *Gestures* categories on the one hand, and the lower frequency of other categories on the other: most user issues occurred while trying to make sense of unfamiliar interface elements and using gestures. Moreover, more detailed analysis showed that while user issues related to the interpretation of unfamiliar interface elements declined after the interface explanation was given, the decline in gesture-related issues was clearly less pronounced. In this respect, the CEM analysis suggests that gestures are more difficult to learn and memorise than visual interface elements.

5. Conclusion and future work

In this paper, we have argued that semiotics, especially the Communicability Evaluation Method (CEM), can offer a valuable framework for the evaluation of multi-touch interfaces. The method offers an analytic tool that allows evaluators to analyse user test data in detail, and arrive at well-motivated general conclusions based on

empirical data. As such, CEM is a specialised evaluation method that has a limited scope (interaction problems), but that helps evaluators shed more light on the exact nature of interactive breakdowns. The level of detail offered by CEM analyses is especially interesting in evaluating innovative interface paradigms, such as multi-touch interaction.

The MuTable case study presented a practical application of an extended version of the CEM framework, tailored to the specifics of multi-touch interaction. The successful extension of the CEM framework suggests that in future research, the CEM framework can be used and further extended to evaluate new developments in interface design. As multi-touch interaction continues to evolve, CEM can aid the process of convergence to a common user interface language by offering an analytic framework to evaluate alternative multi-touch designs, and helping designers decide which interaction style offers the best system-user metacommunication. Moreover, in addition to multi-touch-specific evolutions, the natural user interface (NUI) paradigm as a whole continues to expand beyond multi-touch interaction to e.g. voice control of devices such as TVs and phones. Such new NUI systems pose new interaction design challenges, and it will be interesting to see how the CEM framework can be extended and used to evaluate such innovative interaction designs. In this way, semiotic analysis can continue to play a significant role in the development and evaluation of new, innovative user interaction.

Acknowledgements

The IBBT MuTable project is a project cofunded by IBBT (Interdisciplinary institute for Technology), a research institute founded by the Flemish Government. Companies and organisations involved in the project are CUO (KULeuven), EDM (UHasselt), SMIT (VUB), Kunstencentrum Vooruit, Artec, STAM, Ravi, and University Library of Ghent, with project support of IWT.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at <http://dx.doi.org/10.1016/j.ijhcs.2012.05.005>.

References

- Andersen, P.B., 1990. *A Theory of Computer Semiotics*. Cambridge University Press, Cambridge.
- Andersen, P.B., 2001. What Semiotics can and cannot do for HCI. *Knowledge-Based Systems* 14 (8), 419–424.
- Apple Inc., 2011. *iOS Human Interface Guidelines*. User Experience. Available from: <<http://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/MobileHIG.pdf>> (accessed 20.07.11).
- Apted, T., Kay, J., Quigley, A., 2006. Tabletop sharing of digital photographs for the elderly. In: Grinter, R., Rodden, T., Aoki, P., Cutrell, E., Jeffries, R., Olson, G. (Eds.), *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '06. ACM, New York, NY, pp. 781–790.
- Bardzell, J., Bardzell, S. 2008. Interaction criticism: a proposal and framework for a new discipline of hci. In: *CHI '08 Extended Abstracts on Human Factors in Computing Systems (CHI EA '08)*. ACM, New York, NY, USA, pp. 2463–2472.
- Benyon, D., 2001. The new HCI? Navigation of information space. *Knowledge-Based Systems* 14 (8), 425–430.
- Blackwell, A., 2006. The reification of metaphor as a design tool. In: *ACM Transactions on Computer-Human Interaction* 13:4. pp. 490–530.
- Brandl, P., Forlines, C., Wigdor, D., Haller, M., Shen, C., 2008. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In: *Proceedings of the Working Conference on Advanced Visual Interfaces*. AVI08, New York, NY, USA, 2008, pp. 154–161.
- Chandler, D., 2002. *Semiotics. The basics*. Routledge, London, UK.
- Derboven, J., De Roeck, D., Verstraete, M., Geerts, D., Schneider-Barnes, J. Luyten, K., 2010. Comparing user interaction with low and high fidelity prototypes of tabletop surfaces. In: *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordCHI '10)*, ACM, New York, NY. 148–157.
- De Souza, C.S., 1993. The semiotic engineering of user interface languages. *International Journal of Man-Machine Studies* 39 (5), 753–773.
- De Souza, C.S., 2001. Semiotic approaches to user interface design. *Knowledge-Based Systems* 14 (8), 415–418.
- De Souza, C.S., 2005. *The Semiotic Engineering of Human-Computer Interaction (Acting with Technology)*. The MIT Press, London, UK.
- De Souza, C.S., Leitão, C.F., 2009. *Semiotic Engineering Methods for Scientific Research in HCI*. Morgan & Claypool, San Francisco, CA.
- De Souza, C.S., Leitão, C.F., Prates, R.O., da Silva, E.J., 2006. The semiotic inspection method In: *Proceedings of VII Brazilian Symposium on Human Factors in Computing Systems, IHC '06*, 323, ACM, New York, NY. 148–157.
- Hinrichs, U., Carpendale, S., 2011. Gestures in the wild: studying multi-touch gesture sequences on interactive tabletop exhibits. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11*. 3023–3032.
- Kahler, H., 2000. Methods & tools. Constructive interaction and collaborative work: introducing a method for testing collaborative systems. *Interactions* 7 (3), 27–34.
- Kjeldskov, J., Paay, J., 2010. Indexicality: understanding mobile human-computer interaction in context. *ACM Transactions on Computer-Human Interaction* 17 (4), 28.
- Kristensson, P.O., Arnell, O., Björk, A., Dahlbäck, N., Pennerup, J., Prytz, E., Wikman, J., Aström, N., 2008. InfoTouch: an explorative multi-touch visualization interface for tagged photo collections. In: *Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges*. NordiCHI '08, ACM, New York, NY. 491–494.
- Mazalek, A., Winegarden, C., Al-Haddad, T., Robinson, S. J., Wu, C., 2009. Architales: physical/digital co-design of an interactive story table. In: *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction, TEI '09* ACM, New York, NY. 241–248.
- Maximo, A., Saba, M.P., Velho, L., 2009. CollecTable: a natural interface for music collections. In: *SIGGRAPH '09: Posters (SIGGRAPH '09)*. ACM, New York, NY.
- Merriam-Webster Dictionary, 2011. Metonymy. <www.merriam-webster.com/dictionary/metonymy> (accessed 19.07.11).
- Nacenta M.A., Baudish P., Benko H., Wilson A., 2009. Separability of spatial manipulations in multi-touch interfaces. In: *Proceedings of the Graphics Interface 2009 (GI '09)*. 175–182.
- Norman, D.A., 2010. Natural user interfaces are not natural. *Interactions* 17 (3), 6–10.
- Norman, D.A., Nielsen, J., 2010. Gestural interfaces: a step backward in usability. *Interactions* 17 (5), 46–49.

- O'Neill, S., 2008. *Interactive Media: The Semiotics of Embodied Interaction*. Springer, London, UK.
- Pavlus, J., 2010. Synthesizer 76 iPad App Shows Delights and Pitfalls of “Skeuomorphic” UI’s. <fastcodesign.com> (accessed 19.07.11).
- Peltonen, P., Kurvinen, E., Salovaara, A., Jacucci, G., Ilmonen, T., Evans, J., Oulasvirta, A., Saarikko, P., 2008. It’s Mine, Don’t Touch! Interactions at a large multi-touch display in a city centre. In: *Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems, CHI ‘08*. ACM, New York, NY. 1285–1294.
- Pinelle, D., Gutwin, C., 2008. Evaluating teamwork support in tabletop groupware applications using collaboration usability analysis. *Personal and Ubiquitous Computing* 12 (3), 237–254.
- Piper, A.M., Hollan, J.D., 2009. Tabletop displays for small group study: affordances of paper and digital materials. In: *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI ‘09*. ACM, Boston, MA. 1227–1236.
- Piper, A.M., O’Brien, E., Morris, M.R., Winograd, T., 2006. SIDES: a cooperative tabletop computer game for social skills development. In: *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work (CSCW ‘06)*, ACM, New York, NY. 1–10.
- Prates, R.O., Barbosa, S.D., de Souza, C.S., 2000a. A case study for evaluating interface design through communicability. In: *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, ACM, New York, NY. 308–316.
- Prates, R.O., de Souza, C.S., Barbosa, S.D., 2000b. Methods and tools: a method for evaluating the communicability of user interfaces. *Interactions* 7 (1), 31–38.
- Prates, R.O., Souza, C.S., Assis, P.S., 2001. Categorizing communicability evaluation breakdowns in groupware applications. *CHI-SA2001*. Pretoria, RSA.
- Saffer, D., 2008. *Designing Gestural Interfaces: Touch Screens and Interactive Devices*. O’Reilly Media, Sebastopol, CA.
- Schneider, J., Derboven, J., Luyten, K., Vleugels, C., Bannier, S., De Roeck, D., Verstraete, M., 2010. Multi-user multi-touch setups for collaborative learning in an educational setting. *Lecture Notes in Computer Science* 6240, 181–188.
- Shrimpton-Smith, T., Zaman, B., Geerts, D., 2008. Coupling the users: the benefits of paired user testing for iDTV. *International Journal of Human-Computer Interaction* 24 (2), 197–213.
- Spindler, M., Stellmach, S., Dachselt, R., 2009. PaperLens: advanced magic lens interaction above the tabletop. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS ‘09)*, ACM, New York, NY. 69–76.
- Tuddenham, P., Robinson, P., 2009. Territorial coordination and workspace awareness in remote tabletop collaboration. In: *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI ‘09 ACM*, Boston, MA. 2139–2148.
- Vanacken, D., Demeure, A., Luyten, K., Coninx, K., 2008. Ghosts in the interface: meta-user visualizations as guides for multi-touch interaction. In: *Proceedings of the IEEE Tabletops and Interactive Surfaces 2008*, Amsterdam, NL. 87–90.
- Vinh, K., 2011. Unnecessary Explanations. <subtraction.com> Feb 01, 2011 (accessed 25.07.11).
- Virgin Digital Publishing, 2011. Project Magazine. Available from: <http://itunes.apple.com/us/app/project/id404942717?mt=8> (accessed 28.07.11).
- Weiss, M., Wagner, J., Jansen, Y., Jennings, R., Khoshabeh, R., Hollan, J.D., Borchers, J., 2009. SLAP widgets: bridging the gap between virtual and physical controls on tabletops. In: *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI ‘09)*, ACM, New York, NY. 481–490.
- Westerman, 2008. How People Really Use the iPhone. Create with Context Research Report. Available from: <www.createwithcontext.com/how-people-really-use-the-iphone.html> (accessed 15.01.12).
- Wigdor, D., Wixon, D., 2011. *Brave NUI World*. Morgan Kaufman, Burlington, MA.
- Wigdor, D., Williams, S., Cronin, M., Levy, R., White, K., Mazeev, M., Benko, H., 2009. Ripples: utilizing per-contact visualizations to improve user interaction with touch displays. In: *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST ‘09)*, ACM, New York, NY. 3–12.
- Wixon, D., 2008. The challenge of emotional innovation. *UX Week*. Available from: <http://vimeo.com/2893051> (accessed 19.07.11).
- Wobbrock, J.O., Morris, M.R., Wilson, A.D., 2009. User-defined gestures for surface computing. In: *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI ‘09 ACM*, New York, NY. 1083–1092.