Multi-User Multi-Touch Setups for collaborative Learning in an Educational Setting

Jan Schneider, Jan Derboven, Kris Luyten, Chris Vleugels, Stijn Bannier, Dries De Roeck, Mathijs Verstraete

Hasselt University – tUL – IBBT
Expertise Center of Digital Media, Wetenschapspark 2, 3590 Diepenbeek, Belgium.
IBBT - KULeuven
Centre for User Experience Research, Parkstraat 45 / 3605, 3000 Leuven, Belgium.
IBBT- VUB
Studies on Media Information & Telecommunication, Pleinlaan 9, B-1050 Brussels, Belgium.

{jan.schneiderbarnes; kris.luyten}@uhasselt.be, {jan.derboven; dries.deroeck; mathijs.verstraete}@soc.kuleuven.be
{chris.vleugels; stijn.bannier}@vub.ac.be

Abstract. In educational settings, current digital technologies often work counter-productive because people using them experience separation and isolation. This paper describes a set of multi-touch multimedia interaction applications that especially were designed to enhance co-located collaboration between users. We present the underlying framework for creating such applications. Our applications were created for supporting typical collaborative tasks performed by secondary students. We present our findings on the usage of these applications by these users in the settings of a secondary school classroom.

Keywords: CSCL, multi-touch screens, interaction techniques

1 Introduction

Collaboration in educational settings can be supported by a variety of digital technologies offering opportunities to connect learners and teachers in shared spaces for creating and developing work. Examples of technologies that can enhance collaboration are, amongst others, wikis, blogs, learning management systems, file sharing applications or online collaboration workspaces. The domain of 'Computer Supported Collaborative Learning' (CSCL) [1,2,3,4,5,6] conceptualizes collaboration as a process of shared meaning construction; i.e. through processes of interaction, meaning is achieved. The focus is on learning through collaboration with other learners rather than directly from a teacher. Furthermore, it is stressed that in this process of collaboration, learners learn by expressing their questions, pursuing lines of inquiry together, teaching each other and seeing how others are learning. Therefore

the role of technology is to support collaboration by providing media of communication and scaffolding for productive interaction between learners.
In this article we present a multi-touch multimedia interaction framework, whose applications were designed to enhance collocated, real-time collaboration between learners. The design and development of our multi-touch applications was based on the scenario of a school assignment where the class performs research on a specific topic in small groups. Later, their findings are presented to each other in front of the class. Both the research as the presenting activity is performed on a multi-touch surface.
The main contribution of this paper is to elaborate on the acceptance of a multi-touch setup inside of a secondary school classroom, showing how this can be used as a tool that facilitates collaboration among students.

2 Software Architecture for Multi-Touch

2.1 Basic Architecture

Our basic architecture has been designed based on previous experiences in developing multi-touch applications. The initial software framework, Eunomia [7], was developed for interaction with multimedia objects on multi-touch surfaces. This framework was written in C++which meant that adding new elements to the graphical interfaces built on top of Eunomia required a profound knowledge of both the framework and C++. Based on this experience, we created an alternative software framework, *MuTable*, that is both simplified with respect to programming new behavior and more accessible for designers to add user interface designs. This section provides an overview of the architecture of the software framework, decomposed in functional components. We use the term software framework for a layered set of components that can be used to build multi-touch applications
Input Layer: There is a wide diversity of multi-touch hardware available on the market which is still evolving at a fast pace. This implies we need an input layer that can collect touch points regardless of the hardware it is running on. We provide an input layer component that uses a client-server setup in which the hardware acts as the server sending out touch points, and the multi-touch application acts as a client receiving these touch points. The way the server works is dependent on the hardware, but we provide a set of generic implementations so the server can run on Windows 7 and use its multi-touch capabilities or at any TUIO compliant system [8]. There is a fixed protocol that is used between the client and the server, so clients know what to expect and can process this input.
Core Handler: Our Framework uses the Windows Presentation Foundation libraries which belong to the .Net framework. The Core Handler of this framework is the main entry for MuTable applications. It receives the touch-events raised from the Input Layer and passes them to the touch point controller.
Touch-Point Controller: Gestures that occur over a multi-touch screen are context dependent. The simple gesture of moving two touch-points away from each other could be translated into enlarging an object, or moving to objects away from each

other. The Touch-Point Controller is in charge to identify the context of each touch-event and assigns it to the correct multi-touch widget.
Multi-Touch Widget Container: All displayed widgets are contained in the multi-touch widget container. It allows adding new widgets and deleting old ones, and serves as the top-level container.
Multi-Touch Widgets: Are all components in the screen which can be manipulated by users through touch-events. They all derive from our MultiTouchObject abstract class. Multi-touch widgets are responsible for interpreting the multi-touch gestures they can be controlled with. This makes it easier to add new widgets that might work radically different. This choice has proven to be very useful for experimenting with prototypes to find the most appropriate way of interaction. Multi-touch widgets can be composed from other multi-touch widgets. This hierarchical approach allows combining several existing widgets to obtain more complex widgets. An example is a multi-touch virtual keyboard, this keyboard consist of one container and buttons that by themselves are again multi-touch widgets.

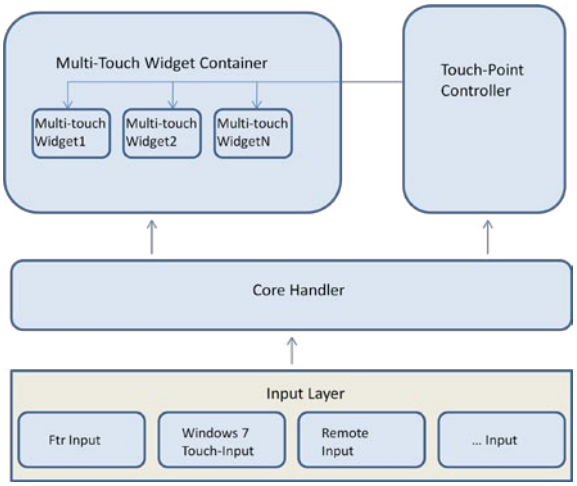

Fig. 1. Software Architecture Diagram

2.2 Common User Interactions

Multi-touch surfaces offer a different type of interaction with respect to traditional mouse-keyboard interaction. They open the possibilities for new types of interaction techniques, but also present some issues that require to be tackled [9,10] We created some common multi-touch interactions techniques for collaborative applications that were kept simple intentionally. Four basic interaction techniques are listed here:
Whole widget manipulation: For the case of composed multi-touch widgets, it is important to know if the user wants to interact with a specific component of the Widget (button, item on a list, slider, etc) or if the user wants to interact with the

widget as a whole. In order to differentiate that, as written in [10] we also use handles. Through the handles it is possible to manipulate the widget as a whole by using some common gestures that have begun to emerge for multi-touch devices [11].
Drag and Drop: Composed widgets usually contain items that can be dragged out of them and be placed somewhere else. This action is done by pressing the drag able object with two contact points (fingers) and moving one or both of them to the place where the object should be dropped
Flip Pages: Book widgets are represented visually as open books. Navigating through its pages is done by pressing the widget with three fingers. Two fingers stay still and the third one moves to the left to see the next to pages or to the right to see the previous ones.
Crop: It is possible to copy extracts of media files. To get these fragments, the user touches the media object with four fingers. A rectangle appears in the screen giving the user a visual feedback showing the area to be cropped. The user can change this area by moving the touch-points, and by releasing them an image copy of the selected region is created.

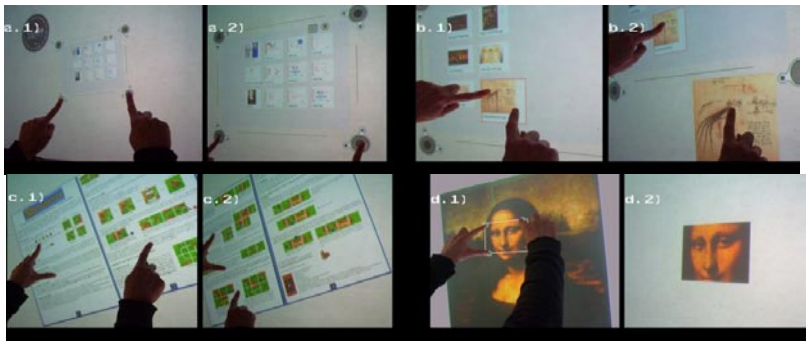

Fig. 2. a) Whole widget Manipulation b) Drag and Drop c) Flip Pages d) Crop

2.3 From Prototype to Deployment

Our applications were the result of a close collaboration between interaction designers and software developers. First, interaction designers created several low fidelity mock-ups that were iteratively tested. Afterwards, high fidelity visualizations were created based on the feedback of the low fidelity end-user tests. In a constant feedback loop, both interaction designers and developers interacted with each other on the progress made in the development and design process.
Before software development started, several cardboard mockups of the crucial low-level interface elements were created. Using tangible materials to create a “natural user interface” helped the team to make the link between how objects were manipulated in real life and how they could be transposed to the digital realm. In a second phase of low-fidelity prototyping, a tabletop prototype allowed the design team to focus on high-level user interaction: to observe how groups of users interact with each other, with the material available, and with the surface they work on in a controlled environment. This second prototype involved a physical table at which test

users were asked to collaborate. They were asked to create a presentation on a given subject, using physical materials such as an analogue typewriter, material to paste, to draw, etc. The results from the second prototyping phase allowed the interaction designers to integrate the lower-level interface elements of the first phase into a larger application structure in an optimal way.
In the last phase, the lessons learned during the low fidelity tests were transferred to high-fidelity visualizations. The challenge here was matching the ideas and designs of the interaction designers (resulting from the low-fidelity tests) to a feasible technical implementation. Our framework made it straightforward to integrate the high fidelity visualizations (designs created in Adobe Illustrator) into the applications being developed: the Illustrator files were exported to XAML files using Microsoft Expression Blend, which were immediately usable in the final applications.

3 Application Support for Educational Tasks

3.1 The Collaborative Workspace

In a collaborative workspace it is important to recognize the users and let them work on individual tasks as well as on mutual ones. Platforms like the DiamondTouch are able to detect a limited amount of users [12]; still most of the multi-touch hardware technologies nowadays identify many touch-points, but not the users associated with these. We dealt with this issue by assigning each user with a delimited space on the touch-screen where he/she is allowed to work on his/her individual tasks: an individual workspace. It is, however, not possible to predict exactly how many users will be using the touch-screen at the same time. Considering the size of the table, the number of users could vary between one and four. Furthermore, size, shape and location of the workspace are intricate issues: numerous experiments have already considered this issue of work spaces, territoriality, and 'ownership' of digital artifacts [13,14,15,16].
The Mutable platform presents a workspace by a small circle on the screen, a 'central ball'. Once this central ball is touched, the options available to the user appear round the circle (which amounts to an implementation of sorts of a pie menu [17]). When selecting one of these options, the relevant application opens, connected to the central ball with a line. Users can always relocate and resize each of the applications that were opened from their workspace. In addition, the central balls themselves can also be moved around on the touch-table
New workspaces can be created by touching dedicated buttons on the sides of the multi-touch screen. This location allows all users to create a new workspace, without having to reach too far. At first, another strategy was implemented: touching the screen border in any place opened a new work space - no matter where users are located around the multi-touch table, they are always close to an edge. However, this strategy ran into some issues: a lot of unintended workspaces were created, as people tend to rest their non-dominant hand (or elbows, if the table is placed at the appropriate height [18]) on the edge of the touch surface - even when the table has a framing that is not touch-sensitive.

'Ownership' of workspaces is not strictly organized by the Mutable platform: users can work alone on their own workspaces and applications, or they can share one workspace among several users.

3.2 Functionality Overview

The functionality consists of a number of separate, dedicated applications. In general, the applications can be divided in three functional groups: (i) searching and browsing content, (ii) composing and creating content (the main focus of the Mutable framework), and (iii) presenting content. An instance of all applications is available for each personal workspace; all users can collaborate using a common workspace, or each one of them can use their private one. Content can be moved from one workspace to another without restrictions, making for a very flexible framework.
(i) Searching for content. This application contains two views. The first one shows a list with thumbnails of the media files that are available. And the second one contains a virtual keyboard that can be used to type for keywords in order to search for specific content and narrow down the information displayed
(ii) Composing and creating content. The content of this application is threefold; it contains widgets to create text, free hand paintings and presentations. The widget used to create text simulates an old typewriter. It contains a virtual keyboard and a list of textboxes. With it, users can create separate 'snippets' of text that can be dragged and dropped in other documents. The free-hand painting widget contains an area that accepts user's paint strokes: they can select a color, start painting and later drag their creation and drop it into another document. The presentation widget allows students to create slides and arrange them for a presentation. It contains a list showing the thumbnail of the slides inside the presentation. Users can create, edit or delete slides. Slide layouts can be selected from a limited number of templates. After selecting a template, users can start dragging and dropping content into the slide.
(iii) Presenting content. For this the interface turns into 'presentation mode'. A theater stage image becomes the background and a list showing the thumbnails of the slides appears on the bottom of the screen. By touching the thumbnails the moveable, scalable, and rotatable slides appear at their original size in the middle of the screen.

4 In-Situ Evaluation

The location for the education test was at the Leonardo Lyceum in Antwerp, Belgium. Within the education context of the project, this secondary school was able to provide a 2nd form class and a classroom. The school has an advanced ICT infrastructure. For every two students there is one computer available. The school has four ICT classrooms, each equipped with 13 computers and a smart board. The multi-touch setup was positioned in the middle of the ICT classroom, also known as the open learning center. With an extra beamer connected to the multi-touch setup, other students were able to see the activities performed on the setup. The students got an assignment to make a presentation about Leonardo Da Vinci. After a brief introduction of the setup we provided including the applications and interaction, three

students made the assignment. The other students created a presentation, two by two, on normal computers using standard web search. After one hour these students were invited to accompany the first three students and continue the assignment. At this moment the students introduced already working with the multi-touch setup transferred their knowledge on the system to the other three. The students had two hours to complete the Da Vinci presentation. The third hour was used as an epilogue to discuss and compare the multi-touch setup versus the computer.
A brief explanation of the main operations and possibilities were sufficient for the three students to start the assignment. They had no additional help during the creation and instructed the other students easily at using the table. In particular, the peer-to-peer learning and collaborative working skills were frequently addressed during this test. When asked for comparing collaboration using computers versus a shared multi-touch surface, the students primarily noted that when using a mouse one person controls the whole collaboration while the multi-touch setup did not impose such leadership. Note the tasks at hand (research, creation and presentation) are well suited for collaboration, but traditional tools fail to exploit these

Fig. 3. Students working on the multi-touch setup.

5 Conclusions

In this paper we presented our multi-user multi-touch setups for together learning in an educational setting. We started with creating an underlying framework that supports the design of collaborative multi-touch applications. Our focus was on the collaborative aspect, where multiple users should be able to use the applications simultaneously. To validate and evaluate our framework, a set of applications were designed and developed for secondary school students. Three typical collaborative tasks were selected: searching for information, composing this information in a presentable format and presenting this information to others. The valuation showed collaboration, and thus together learning, were improved using a large multi-touch surface accompanied with applications that were designed for collaboration. Concerning functionality offered, the students asked for more personalization features of the interface, also a clear ‘undo’ function was missing. The test at the Leonardo

Lyceum showed the MuTable and its interface to be a valuable and well deployable new media technology for the educational context.

6 References

1. Dillenbourg, P.: What do you mean by "collaborative learning"? In P. Dillenbourg (Ed.), Collaborative learning: Cognitive and computational approaches, pp. 1-16. Amsterdam, NL (1999).
2. Loveless, A. M.: Creativity, technology and learning – a review of recent literature University of Brighton, School of Education. Available at www.futurelab.org.uk/litreviews
3. Meijer, J., van Eck, E., Charles F.: Leren met meer effect; rapportage van het onderzoek. Amsterdam, Universiteit van Amsterdam, SCO-Kohnstamm Instituut. Available at http://onderzoek.kennisnet.nl(2008)
4. Saab, N. (2005). Chat and explore. The role of support and motivation in collaborative scientific discovery learning. Amsterdam: University of Amsterdam.
5. Stahl, G.: Group cognition: Computer support for building collaborative knowledge. Cambridge, MA: MIT (2006)
6. Baker, M., Lund, K.: Promoting reflective interactions in a CSCL environment. In, JCAL 1997, pp. 175-193.
7. Cuypers T., Schneider J., Taelman J., Luyten K., Bekaert P.: Eunomia: Toward a Framework for Multi-touch Information Displays in Public Spaces. In Proc. BCS-HCI 2008, pp. 31-34
8 Kaltenbrunner, M; Bovermann, T; Bencina, R; Costanza, E. TUIO: A Protocol for Table-Top Tangible User Interfaces. In Proc GW 2006
9. Sousa C., Matsumoto M.: Study on Fluent Interaction with Multi-Touch in traditional GUI Enviroments. In TENCON 2007 -, pp. 1--4
10. Nacenta M., Baudish P., Banko H., Wilson A.: Separability of Spatial Manipulations in Multi-Touch Interfaces. In Proc. GI 2009, pp. 175--182.
11. Wigdor D., Fletcher J., Morrison G.: Designing User Interfaces for Multi-Touch and Gesture Devices. In CHI 2009, pp. 2755-2758.
12. Dietz, P.H.; Leigh, D.L.: DiamondTouch: A Multi-User Touch Technology. In UIST 2001, pps 219-226.
13. Scott, S.D., Carpendale, M.S.T., Inkpen, K.M.: Territoriality in Collaborative Tabletop Workspaces. In CSCW 2004.
14. Tuddenham, P., and P. Robinson: "Territorial coordination and workspace awareness in remote tabletop collaboration. In CHI 2009, pp. 2139-2148.
15. Scott, S.D., Grant K.D., Mandryk R.L.: System Guidelines for Co-located, Collaborative Work on a Tabletop Display. In ECSCW 2003.
16. Peltonen, P., E. Kurvinen, A. Solovaara, G. Jacucci, T. Ilmonen, J. Evans, A. Oulasvirta, and P. Saarikko: It's Mine, Don't Touch!: interactions at a large multi-touch display in a city centre. In CHI 2008
17. Callahan, J., Hopkins, D., Weiser, M., Shneiderman, B.: An empirical comparison of pie vs. linear menus. In Proc. SIGCHI 1988, pp. 95-100.
18. Varcholik P., Laviola, J., Nicholson, D.: TACTUS: A Hardware and Software Testbed for Research in Multi-Touch Interaction. In Proc. HCI International 2009.