Craig Anslow · Pedro Campos
Joaquim Jorge *Editors*

Collaboration Meets Interactive Spaces

Springer

Collaboration Meets Interactive Spaces

Craig Anslow · Pedro Campos
Joaquim Jorge
Editors

Collaboration Meets Interactive Spaces

Springer

Editors
Craig Anslow
Department of Computer Science
Middlesex University
London
UK

Joaquim Jorge
Department of Computer Science and Engineering, Instituto Superior Técnico
University of Lisbon
Lisbon
Portugal

Pedro Campos
Madeira Interactive Technologies Institute
University of Madeira
Funchal
Portugal

ISBN 978-3-319-45852-6 ISBN 978-3-319-45853-3 (eBook)
DOI 10.1007/978-3-319-45853-3

Library of Congress Control Number: 2016952912

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Foreword

Multitouch surfaces are irresistible. People readily walk up to and press their fingers on the surfaces of tabletops and wall displays in museums, galleries, libraries, and stores, anticipating what will happen—when swiping, tapping, pushing, dragging, and stretching the digital content. At the same time, babies are learning, from just a few months old, to swipe before learning any other kind of interaction. It is what they instinctively do now when encountering anything new. Just look at the countless videos online of babies swiping at books, trees, and other objects—with the learned expectation, it will cause an effect.

It is without question that interactive surfaces have come of age—especially for the individual user. But they offer much more—especially the opportunity for multiple people to collaborate around and through them. While early research made in-roads into how to enable this to happen, many questions remain still unanswered. Of central concern is optimizing ways for groups to work together, co-located, or apart, when using shared surfaces of one form or another—be it videoconferencing; sharing of screens in real time; moving between multiple devices in the same place; or using a single shared display. This book covers new research and observations that address the challenges and opportunities of working across surfaces.

Since the early days of the Diamond Touch technology and other customized interactive shared surfaces, there has been much research investigating how to support intuitive interactions. The lightweight and parallel action of touching, the mobility of users, and the increased ability for natural expressions of behavior such as gesture and posture extend the possibilities for communication and collaboration.

Core issues that are covered include the best protocols and norms for enabling people to work together using multiple technologies or shared surfaces and how can they be managed fluidly and effortlessly. People's actions, comments, and gestures can all be seen, heard, and experienced by others using shared surfaces. While such actions may become largely invisible to those executing them, as they are so familiar, their enaction, in contrast, remains visible to others. How do groups exploit these in order to coordinate their actions and interactions? Another feature of surfaces, tangibles and shareable public displays is that they enable simultaneous control by multiple users. These technologies, therefore, offer new opportunities for situational awareness—gesture, body orientation and more so-called 'natural' means of communication, for making salient in displays the availability of information supported by the public space provided, and for equitable simultaneous control, such as 'entry points' to the technology. Such possibilities, however, raise further questions: What are the best ways to indicate where people are looking, what each other is doing, what other would like you to do, and so on? Flashing cursors, eye gaze marks, haptic buzzes sounds, or other? How should content be downloaded and uploaded to public and shared displays? If gestures are to be used, what kinds and how many can people be reasonably expected to remember?

The chapters in the book show that there are many collaborative practices that lend themselves to being supported by the use of shared surfaces, including emergency response management, rehabilitation, rural areas, videoconferencing, and education. But for every application, different factors need to be considered as to what is the optimal way to support, promote, and augment them. For example, what are the best size, shape, and orientation of the kinds of displays that are used? If a number of displays are available, how should content flow between them so that people understand and manage what is happening? Should everyone be able to interact at the same time or should constraints be put in place to force turn-taking and enable better situation awareness—rather than simply have a free-for-all form of interaction?

The 19 chapters in this book cover a range of topics. In Part I, there are a number of chapters that cover interaction techniques, large displays, and the way other technologies might be used with them, such as wearables. In Part II, case studies and applications are covered that consider different models, frameworks, and software methods for designing and implementing various configurations. Together, they offer new understandings, methods, and frameworks for researchers and designers as a way of generating ideas, codifying observations, and reflecting on how to support collaboration around interactive surfaces.

London
July 2016

Prof. Yvonne Rogers
University College London

Acknowledgements

This book would not have been possible without the voluntary effort of the reviewers who dedicated their renowned expertise to selecting and improving the chapters this book is made of. We list them below as a symbolic way of thanking them.

Reviewers:
Jason Alexander, Lancaster University, UK
Pedro Antunes, Victoria University of Wellington, New Zealand
Mirjam Augstein, University of Applied Sciences Upper Austria, Hagenberg, Austria
Jakob Bardram, Technical University of Denmark, Denmark
Robert Biddle, Carleton University, Canada
Jacob Biehl, FXPal, USA
Mark Billinghurst, University of South Australia, Australia
John Brosz, University of Calgary, Canada
Diogo Cabral, University of Madeira, Portugal
Nuno Correia, Goldsmiths, University of London, UK
Raimund Dachselt, Technical University of Dresden, Germany
Florian Daiber, German Research Center for Artificial Intelligence (DFKI), Germany
Bruno De Araujo, University of Toronto, Canada
Sebastian Doeweling, SAP, Germany
Morten Esbensen, IT University of Copenhagen, Denmark
Denzil Ferreira, University of Oulu, Finland
Joel Fisher, University of Nottingham, UK
Laurent Grisoni, University of Lille, France
Tom Gross, University of Bamberg, Germany
Mark Hancock, University of Waterloo, Canada
Steven Houben, University College London, UK
Pourang Irani, University of Manitoba, Canada
Tobias Isenberg, INRIA, France
Judy Kay, University of Sydney, Australia

Clemens Kolkmose, Aarhus University, Denmark
Per Ola Kristensson, University of Cambridge, UK
Antonio Kruger, University of Saarland, Germany
Markus Lochtefeld, Aalborg University, Germany
Andrés Lucero, University of Southern Denmark, Denmark
Frank Maurer, University of Calgary, Canada
Daniel Mendes, University of Lisbon, Portugal
Alex Nolte, University of Bochum, Germany
Ian Oakley, Ulsan National Institute of Science and Technology, South Korea
Beryl Plimmer, University of Auckland, New Zealand
Harald Reiterer, University of Konstanz, Germany
Markus Rittenbruch, Queensland University of Technology, Australia
Stacey Scott, University of Waterloo, Canada
Ed Tse, Smart Technologies, Canada
Jo Vermeulen, University of Calgary, Canada
Jim Wallace, University of Waterloo, Canada

Contents

Contributors

Craig Anslow Department of Computer Science, Middlesex University, London, UK; University of Calgary, Calgary, AB, Canada

Mirjam Augstein University of Applied Sciences Upper Austria, Hagenberg, Austria

Jakob E. Bardram Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby, Denmark

Andrea Bellucci Universidad Carlos III de Madrid, Madrid, Spain

Robert Biddle School of Computer Science, Carleton University, Ottawa, ON, Canada

Mark Billinghurst Empathic Computing Laboratory, University of South Australia, Adelaide, Australia

Jens Bornschein Human-Computer Interaction, Technische Universität Dresden, Dresden, Germany

Ben Brown Faculty of Engineering, University of Auckland, Auckland, New Zealand

Judith M. Brown School of Computer Science, Carleton University, Ottawa, ON, Canada

Ross Brown Information Systems School, Queensland University of Technology, Brisbane, Australia

Neil Bruce Department of Computer Science, University of Manitoba, Winnipeg, Canada

Pedro Campos Madeira Interactive Technologies Institute, University of Madeira, Funchal, Portugal

Alan Chamberlain University of Nottingham, Nottingham, UK

Edwin Chan University of Calgary, Calgary, AB, Canada

Y.-L. Betty Chang Games Institute/Systems Design Engineering, University of Waterloo, Waterloo, ON, Canada

Raimund Dachselt Interactive Media Lab, Technische Universität Dresden, Dresden, Germany

Sabine Dielacher Neuromed Campus Kepler Universitätsklinikum, University of Applied Sciences for Health Professions Upper Austria, Linz, Austria

James Diprose Department of Computer Science, University of Auckland, Auckland, New Zealand

Alan Dix Talis and University of Birmingham, Birmingham, UK

Sebastian Döweling SAP SE, Strategic Projects, Darmstadt, Germany

Simon Du Preez Faculty of Engineering, University of Auckland, Auckland, New Zealand

Barrett Ens Department of Computer Science, University of Manitoba, Winnipeg, Canada

Morten Esbensen Pervasive Interaction Laboratory, IT University of Copenhagen, Copenhagen, Denmark

Alfredo Ferreira INESC-ID/Tcnico Lisboa/University of Lisbon, Lisbon, Portugal

Luca Frosini CNR-ISTI, Pisa, Italy

Steve Gill Cardiff Metropolitan University, Cardiff, UK

Stevenson Gossage School of Computer Science, Carleton University, Ottawa, ON, Canada

Laurent Grisoni Mint, University of Lille Science and Technology, CNRS, INRIA, Villeneuve-d'Ascq, France

Kunal Gupta The HIT Lab NZ, University of Canterbury, Christchurch, New Zealand

Mark Hancock Games Institute/Department of Management Sciences, University of Waterloo, Waterloo, ON, Canada

Thomas Herrmann Information and Technology Management, Ruhr-University Bochum, Bochum, Germany

Tom Horak Interactive Media Lab, Technische Universität Dresden, Dresden, Germany

Pourang Irani Department of Computer Science, University of Manitoba, Winnipeg, Canada

Tobias Isenberg Inria, Saclay, France

Joaquim Jorge Department of Computer Science and Engineering, Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal; INESC-ID/Tcnico Lisboa/University of Lisbon, Lisbon, Portugal

Masai Katsutoshi Keio University, Minato, Tokyo, Japan

Martin Kropp University of Applied Sciences Northwestern Switzerland, Windisch, Switzerland

Kai Kunze Keio University, Minato, Tokyo, Japan

Ricardo Langner Interactive Media Lab, Technische Universität Dresden, Dresden, Germany

Gun Lee The HIT Lab NZ, University of Canterbury, Christchurch, New Zealand

Youngho Lee Empathic Computing Laboratory, University of South Australia, Adelaide, Australia; Mokpo National University, Mokpo, South Korea

Gareth Loudon Cardiff Metropolitan University, Cardiff, UK

Andrew Luxton-Reilly Department of Computer Science, University of Auckland, Auckland, New Zealand

Alessio Malizia Brunel University, Uxbridge, UK

Magdalena Mateescu University of Applied Sciences Northwestern Switzerland, Windisch, Switzerland

Frank Maurer University of Calgary, Calgary, AB, Canada

Daniel Medeiros INESC-ID/Tcnico Lisboa/University of Lisbon, Lisbon, Portugal

Daniel Mendes INESC-ID/Tcnico Lisboa/University of Lisbon, Lisbon, Portugal

Miran Mirza School of Computer Science, Carleton University, Ottawa, ON, Canada

Annett Mitschick Interactive Media Lab, Technische Universität Dresden, Dresden, Germany

Richard Morris Cardiff Metropolitan University, Cardiff, UK

Max Mühlhäuser Telecooperation Group, TU Darmstadt, Darmstadt, Germany

Thomas Neumayr University of Applied Sciences Upper Austria, Hagenberg, Austria

Alexander Nolte Information and Technology Management, Ruhr-University Bochum, Bochum, Germany

Eyal Ofek Microsoft Research, Redmond, USA

Fabio Paternò CNR-ISTI, Pisa, Italy

João Madeiras Pereira INESC-ID/Tcnico Lisboa/University of Lisbon, Lisbon, Portugal

Beryl Plimmer Department of Computer Science, University of Auckland, Auckland, New Zealand

Artem Polyvyanyy Information Systems School, Queensland University of Technology, Brisbane, Australia

Denise Prescher Human-Computer Interaction, Technische Universität Dresden, Dresden, Germany

Yosra Rekik Mint, University of Lille Science and Technology, CNRS, INRIA, Villeneuve-d'Ascq, France

Jan Riemann TU Darmstadt, Telecooperation Group, Darmstadt, Germany

Markus Rittenbruch Queensland University of Technology, Brisbane, QLD, Australia

Renate Ruckser-Scherb Neuromed Campus Kepler Universitätsklinikum, University of Applied Sciences for Health Professions Upper Austria, Linz, Austria

Stacey D. Scott Games Institute/Systems Design Engineering, University of Waterloo, Waterloo, ON, Canada

Teddy Seyed University of Calgary, Calgary, AB, Canada

Peter Simonyi School of Computer Science, Carleton University, Ottawa, ON, Canada

Maurício Sousa INESC-ID/Tcnico Lisboa/University of Lisbon, Lisbon, Portugal

Maki Sugimoto Keio University, Minato, Tokyo, Japan

Aurélien Tabard Université de Lyon, CNRS Université Lyon 1, LIRIS, Lyon, France

Tarik Tahiri SAP SE, Strategic Projects, Darmstadt, Germany

Tommaso Turchi Brunel University, Uxbridge, UK

Radu-Daniel Vatavu MintViz Lab—MANSiD Research Center, University Stefan cel Mare of Suceava, Suceava, Romania

Ulrich von Zadow Interactive Media Lab, Technische Universität Dresden, Dresden, Germany

Moritz Wiechers Information and Technology Management, Ruhr-University Bochum, Bochum, Germany

Jeff Wilson School of Computer Science, Carleton University, Ottawa, ON, Canada

Chapter 1
An Introduction to Collaboration Meets Interactive Spaces

Craig Anslow, Pedro Campos and Joaquim Jorge

Abstract Interactive Surfaces and Spaces have become ever more pervasive in the past decade. Indeed, the current explosion of media that pervades our everyday lives invades our senses through (increasingly) interactive displays surfaces in all sizes, shapes and formats. Indeed, interactive walls, tables, mobiles (tablets and phones), as well as wearables change the way human beings interact with information and collaborate with one another. At the same time, these surfaces and devices are redesigned and reinvented through new social protocols and collaborative work styles that arise from the experimentation and long-term usage of novel people/device ecologies. The book reflects a high interest among researchers and practitioners about this particular approach and the challenges it entails. It offers an up to date and comprehensive scientific overview of the new generation of devices and their myriad combinations. While pervasive display technologies are changing the way we relate to media, people and society are also shaping and adapting new techniques, methods and idioms. Our purpose is to update both researchers and practitioners with exciting new work around the emergence of social protocols that arise from the experimentation and long time usage of interactive surfaces and also includes numerous case studies, based on recent work.

C. Anslow (✉)
Department of Computer Science, Middlesex University, London, UK
e-mail: c.anslow@mdx.ac.uk

P. Campos
Madeira Interactive Technologies Institute, University of Madeira,
Reg. Aut. da Madeira, Funchal, Portugal
e-mail: pedro.campos.pt@gmail.com

J. Jorge
Department of Computer Science & Engineering, Instituto Superior Técnico,
University of Lisbon, Lisbon, Portugal
e-mail: joaquim.jorge@gmail.com

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_1

1.1 Part I: Devices and Techniques for Collaboration Through Interactive Spaces

This book offers an updated comprehensive scientific overview of the interplay between technological advances in collaborative technologies and the social interactions that occur when using interactive surfaces and devices. Existing books on interactive surfaces are either too focused on technological advances and their benefits, or just based on the social interactions that occur when using interactive surfaces and devices. With this new book, we propose to offer an updated comprehensive scientific overview of the interplay between these two factors. One possible way to best describe the book's idea is as follows: interactive walls, tables, mobiles (tablets and phones), and wearables change the way human beings interact with information and collaborate with one another in a co-located or distributed space. At the same time, these surfaces and devices are redesigned and reinvented through the emergence of new social protocols and collaborative work styles that arise from the experimentation and long-term usage of these surfaces. As far as our knowledge goes, no other book offers perspectives on this interplay. Moreover, there is a high interest among researchers and practitioners about this particular approach and the results it entails. Attesting there is a large participation and interest we have been receiving over recent years regarding the workshop series on Collaboration Meets Interactive Surfaces (CMIS): Walls, Tabletops, Mobiles, and Wearables that forms a foundation for this book.

This books starts out by outlining the degrees of freedom we currently find when starting to design interactive surfaces with the specific goal of improving human collaboration and cooperation. Naturally, this restricts the design space and consequently determines how much collaboration can happen in a specific domain, independently of what domain or problem we are addressing. This section of the book sets the stage for a second section about case studies and applications. Devices and techniques were clustered following an analysis of the accepted submissions we received, in the following manner: (i) interaction techniques, (ii) large displays, and (iii) wearables.

1.1.1 Interaction Techniques

Advances have been consistent in interaction techniques for improving collaborative activities mediated by interactive surfaces of all kinds. The first chapter in this book is authored by Plimmer et al. and covers tabletop 3D object manipulation through touch and tangibles. They demonstrate the advantage of combining touch and tangible regarding 3D object manipulation. The research was constructed on top of a 3D turn-based game and investigates how the users adapt to the bi-manual interaction using touch as well as the tangible.

In the next chapter, Rekik et al. describe and analyze the variety of movement patterns that participants use when performing multi-touch gestures. This chapter also gives understandings about the different strategies participants use to generate this variety of patterns, when they are asked to, as well as their mental models regarding preference. There were two experiments setup, from which the authors collected nearly 6,700 multi-touch gestures from nearly 50 participants. These experiments provide a qualitative analysis of user gesture variability and also derive a taxonomy for users' gestures that complements existing taxonomies.

Finally, and still inline with interaction techniques for collaborative spaces, Sousa et al. describe work around remote proxemics. As virtual meetings become increasingly common with modern technologies, they still add unproductive layers of protocol to the flow of communication between participants, rendering the interactions far from seamless. Therefore, Remote Proxemics is proposed as a technique to mitigate these negative aspects, as it is an extension of proxemics aimed at bringing the syntax of co-located proximal interactions to virtual meetings.

1.1.2 Large Displays

The interplay between mobile phones and large displays has become a trendy subject, since content sharing between large displays and personal mobile devices is central to many collaborative usage scenarios. This is the topic of the first chapter in this section. Langner et al. describe NiftyTransfer, a suite of 5 bidirectional transfer interaction techniques to move digital content between mobile devices and a large vertical display. The chapter provides a detailed overview of design goals and interaction techniques, and reports on a study exploring the usefulness of the techniques. Five techniques are presented that explore three main aspects: multi-item transfer and layout, the dichotomy of casual versus precise interaction, and support for physical navigation.

Isenberg presents a survey of the different approaches currently in use regarding the interaction with large surfaces. The analysis is performed in the context of scientific visualizations, which has traditionally been a domain where large scale and/or high resolutions displays are particularly useful. This chapter demonstrates that the reported systems are valuable giving a complete overview of 3D scientific visualization on interactive surfaces.

Finally, a very well-known interactive large display is presented: CubeIT, a multi-user presentation and collaboration system at Queensland University of Technology (QUT) in Australia. The output medium of CubeIT is composed of 48 multi-touch screens and projected displays above these. As functionalities, CubeIT allows students and academic staff to share their own multimedia content, allowing collocated and simultaneous screen interaction to explore its content, which is an interesting concept and an insightful evaluation. This system uses three interface mechanisms: the multitouch wall itself, a mobile app, and a website.

1.1.3 Wearables

Wearable systems are becoming more common ways to interact and collaborate through interactive surfaces. This book presents two different contributions in this field: using head-worn displays and using eye gaze.

Shared Façades, by Ens et al. is a new approach for distributing virtual information displays across multiple users. Their method is extremely sophisticated: it applies a random walk algorithm to balance multiple constraints, such as spatial constancy of displayed information, visual saliency of the background, surface-fit, occlusion and relative position of multiple windows, to produce layouts that remain consistent across multiple environments while respecting the local geometric features of the surroundings. Results show that the balanced constraint weighting schema produces better results than schemas that consider spatial constancy or visual saliency alone, when applied to models of two real-world test environments.

Head mounted displays (HMDs) and head worn cameras (HWCs) can be useful for promoting remote collaboration. In the chapter by Billinghurst et al. they describe explorations in using gaze cues for this type of activity. Overall, they conclude that showing gaze cues on a shared video is better than just providing the video on its own, and also that combining gaze and pointing cues is the most effective interface for remote collaboration.

1.2 Part II: Case Studies and Applications

1.2.1 Collaboration Aspects

Chang et al. present advances in collaborative aspects regarding the usage of interactive timelines in collaborative digital tabletops with automation. The techniques in this case study are particularly useful when dealing with highly complex scenarios, since the maintenance of situational awareness in the context of automated dynamic changes is paramount to keeping users making optimal decisions. They designed an interactive event timeline to enable exploration of historical system events. On average, the participant groups exhibited high scores of situation awareness for a cooperative tabletop game task.

Activity-based collaboration for interactive spaces is a new conceptual and technological framework for designing interactive systems with a better mapping between activities people conduct and the digital entities they use. Bardram et al. present this framework together with some applications in supporting collaboration across many interactive surfaces. This chapter provides a focus on the framework's support for collaboration (“activity sharing”) and multiple devices (“activity roaming”), after which two case studies are presented in order to illustrate its application.

1.2.2 Software Development

This book presents several application domains where collaboration is improved through the use of interactive surfaces of different shapes, sizes and capabilities. One of those domains is business process modelling. The chapter by Nolte et al. deals with the challenges of collaborative process modeling and makes the case for interactive spaces where different interactive technologies are combined in order to allow for orchestrating collaboration, since it is possible to form breakout groups on demand or work on a process model in solitude before coming back together.

A related topic is described in the chapter by Kropp et al., where the authors look specifically at cardwalls for agile software development. They present two studies, one on the general use of cardwalls and the second on a concrete tool called aWall (using a large interactive wall display) that supports agile team meetings. As with other case studies presented in this book, this chapter shows encouraging results in the way that team collaboration can be improved through properly designed large interactive surfaces.

1.2.3 Emergency Management

Emergency management is one of the most attractive application domains for collaborative surfaces, especially large ones, since users—under this scenario - make extensive use of large maps to take decisions about any crisis, and to establish a common understanding of a critical situation, in order to plan and coordinate appropriate countermeasures.

Döweling et al. present a comparative study, in which 30 participants performed tasks reflecting actual crisis management work on a tabletop system, classical paper maps and an off-the-shelf desktop geographical information system. They report encouraging results, in which users were most efficient using the tabletop and perceived its user experience as superior. In addition, the tabletop offered a team-work quality comparable to classical paper maps.

Chan et al. propose what they coin as the emergency operations center of the future, an exploration into "the integration of various novel technologies in EOC design, in an effort to make emergency response more efficient." They implemented a multi-surface system that includes display walls, tabletops, tablet devices, and mobile/wearable computing devices as a testbed for examining how proxemics, augmented reality and social media can be used to improve decision-making during emergencies.

1.2.4 Security

Collaboration within interactive spaces cannot happen without a proper technical infrastructure enabling fast communication between the different parts of the user interface, which needs coordination. However, this coordination poses privacy and security problems. Frosini and Paternò describe a solution for achieving secure user interfaces when these are distributed through dynamic sets of users and devices. Their solution has been designed to guarantee authentication, authorization, authenticity and data privacy in collaborative distributed user interfaces. It consists of a software architecture for this purpose and a related implementation. They also demonstrate the importance of authentication mechanisms in the security aspects faced by collaborative user interfaces.

Brown et al. investigate surface application design and development. Their research on security analysis has focused on work to understand information related to security, including both computer security and security in a more general sense. One of the original aspects that stems from their approach is the premise that sensemaking is a key activity. This raises significant challenges, as the intervening actors may be concealing information, and providing misleading or irrelevant information. In addition to the technical contributions, the chapter by Brown et al. is also interesting because it presents a review of several projects about surface computing for security analysis. Several issues were identified from this analysis, including the fact that analysts need to take away results, work alone, and bring back new ideas, and this influences the way surface computing should be implemented, in particular that it can support collaborative epistemic interaction, and they can be improved by support for guidance, interaction history, and annotation.

1.2.5 Medical, Accessibility, and Community

Interactive surfaces have been widely known to the medical domain for quite some time and this book provides two very interesting applications. One application by Augstein et al. about collaboration around an interactive tabletop in rehabilitation settings. The other application by Bornschein and Prescher, about a collaborative workstation for sighted and visually impaired users.

Therapy for patients who acquired brain injury (e.g. a stroke or accident) is quite tedious and difficult due to the repetitive nature of the tasks. Collaboration can be easily facilitated with tabletop computers because they can be interacted with by multiple people in parallel. The chapter by Augstein et al. propose an approach towards rehabilitation using an interactive tabletop in collaborative settings, covering the therapeutic motivation behind as well as aspects related to interaction design and modalities.

The chapter by Bornschein and Prescher about visually impaired users is also very interesting from a technical perspective, since blind users get both auditory and tactile feedback from a workstation through a dynamic planar tactile pin-matrix device.

Both this section and the book finish off with a very relevant chapter by Dix et al., which describes their experience in the design and installation of a low-cost multi-touch table in a rural island community. Among many interesting conclusions, this chapter notes that when installing collaborative surfaces in local communities it is particularly important to be sensitive to local needs and not simply impose a solution because it is the latest, trendy technology. Of course this creates equal challenges in interpreting the research data as each setting is unique with specific stakeholders and issues. We feel this chapter is a perfect closing to this book, as "collaboration meets interactive spaces" also implies that the context of this "meeting" should always take the best interests of the user community that is served by these interactive surfaces.

Acknowledgements This book would not have been possible without the voluntary effort of the reviewers who dedicated their renowned expertise to selecting and improving the chapters this book is made of. We list them below as a symbolic way of thanking them for their efforts.

Part I
Devices and Techniques for Collaboration Through Interactive Surfaces

Chapter 2
Tabletop 3D Object Manipulation with Touch and Tangibles

Beryl Plimmer, Ben Brown, James Diprose, Simon Du Preez and Andrew Luxton-Reilly

Abstract Tabletop environments are ideal for collaborative activities that involve moving and arranging objects. However, manipulating 3D virtual objects through the 2D interface is challenging because users' 2D actions must be translated into 3D actions. We explore the use of touch and tangibles to aid collaboration and 3D object manipulation. Our user study shows that using touch and tangible interaction together has advantages 3D object manipulation. While most users preferred touch due to its familiarity, the tangibles were favored for some tasks.

2.1 Introduction

Three dimensional object manipulation is a common task, which involves translating, rotating or scaling a selected object [11]. These tasks are difficult with 2D input devices because objects can be manipulated on nine dimensions, three dimensions each for translation, rotation and scale [17] but the 2D input device maps naturally to only two dimensions. Examples of tasks that involve 3D object manipulation include laying out animated film sets, furnishing virtual rooms in architectural concept plans and playing games. These tasks are often undertaken by small collaborative groups, thus when using a computer a large display is preferable.

Multi-touch interaction is the current default for large display interaction. An under-explored alternative is Tangible User Interfaces (TUIs). TUIs provide real physical objects with which the user can manipulate virtual objects. They provide a more direct method of interaction than mouse, pen or touch. A number of projects have explored using tangibles for Lego-style construction. However, to the best of

---

B. Plimmer (✉) · J. Diprose · A. Luxton-Reilly
Department of Computer Science, University of Auckland, Auckland, New Zealand
e-mail: b.plimmer@auckland.ac.nz

B. Brown · S. Du Preez
Faculty of Engineering, University of Auckland, Auckland, New Zealand

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_2

Fig. 2.1 Participants interacting with virtual Jenga [19]

our knowledge, there is no work that uses tangibles, with or without multi-touch, to directly manipulate generic objects in a 3D scene (Fig. 2.1).

Collaboration is a key design goal behind many TUI systems [22]; research has shown they are beneficial for collaboration e.g. Jordà et al. [12]. Properties of TUIs that support collaboration include: lowering the barrier to interaction due to the familiarity with real-world interactions [22]; they are more welcoming than mouse driven interfaces due to support for parallel interaction; and they physically embody facilitation, as they can be designed to guide collaboration between participants, for instance, a tangible can be used to give a particular participant control, or encourage equal participation [22]. While we often think of collaboration as working together to build something, competitive games are also a collaborative activity [20]. Players collaboratively agree on the rules of play and on what constitutes a win. They then challenge each other in a collaborative-competitive setting.

To explore combining touch and tangibles for interacting with a 3D world through a tabletop we created virtual Jenga [19], a turn based game where players are situated around a stack of rectangular blocks. The players take turns pulling blocks from the stack and placing them on the top of the stack. The first person to knock the stack over loses. Jenga was our selected context as it allowed us to focus specifically on selection and manipulation of 3D objects. We developed a set of 2D touch gestures and tangibles through an exploratory Wizard of Oz user study. The most promising gesture and tangible interactions were then implemented and iteratively refined in our virtual Jenga game on a Microsoft Pixel Sense tabletop. After usability testing and further refinement, the final evaluation was a Jenga tournament.

This naturalistic evaluation provides an insight into how users can quickly learn to manipulate 3D objects using a combination of touch and tangibles. A video of the project is available online (https://vimeo.com/diprose/tangibles).

2.2 Related Work

Three dimensional object selection and manipulation are common tasks performed in 3D environments. The object manipulation involves choosing the desired object to manipulate (selection) and then translating, rotating or scaling it [11]. Despite this being a common task, it is challenging because there is no natural mapping between the 2D inputs commonly used for computer interaction, and 3D movement [17].

There are four general methods for mapping 2D input into 3D movement [23]. First, on-screen widgets are used to map each axis of movement onto separate controls; the user can then break 3D movement into combinations of 1D or 2D movements. Second, objects can be moved relative to the viewing plane; either parallel or orthogonal. Third, objects can be moved relative to structures in the scene, for example, Oh and Stuerzlinger [17] developed an algorithm that allows objects to be dragged along the surface closest to the viewer but occluded by the object being dragged. Fourth, using heuristics based on the direction of the input device movement. These projects have used mouse input, predating the commercialization of multi-touch input.

Traditionally manipulation of 3D spaces was via a mouse [8]. A mouse provides single point interaction, with all its obvious drawbacks for 3D manipulation. This has been extended to pen interaction, e.g. McCord et al. [13], however, this is also single point interaction. The advent of multi-touch input displays has seen an extension of this work to provide touch interaction to the 3D space, see Jankowski and Hachet [11], for a recent literature survey. However, there is still not a generally accepted set of touch gestures for the many and varied tasks that can be undertaken in virtual 3D spaces. Various other input methods have been proposed including immersive environments, brain-computer interaction, and puppetry using depth-sensing cameras. Tangibles on a tabletop match our real-world experience more closely than these alternatives

Tangible User Interfaces (TUIs) are an alternative to mouse, pen or touch. TUIs provide a real physical object for the user to manipulate a virtual object with [25], which gives users a more direct method of interacting with a computer [24].

A number of TUI systems have been created for constructing geometric LEGO-like models on a computer using physical blocks, e.g. [26]; Aish et al. [1]. Altering the construction of the physical blocks updates the model displayed on the screen. Cuendet et al. [4] had participants manipulate a 3D world and then select particular block edges. Other work has explored how tangibles can be stacked [2] or sensed in mid-air Held et al. [9]. However, both approaches have unsolved problems. Rearranging stacking tangibles is fiddly and constrains the user to the real world physics of the tangibles. Held et al. [9] used tangibles and a depth sensing camera to move objects in a scene, they report that users could, with some practice, create a simple story, but there are numerous limitations to the current implementation. Working on the tabletop provides a more natural working space with higher accuracy input.

To the best of our knowledge, there is no work that uses tangibles to directly manipulate generic objects in a 3D scene or work that combines multi-touch and tangibles. The closest work is from Bock et al. [3], who developed a set of tangible widgets for playing 2D multiplayer tablet games. They compared the usability of the tangibles to multi-touch interaction and found that users preferred the tangibles over multi-touch. A possible reason for this is that the tangibles allowed the users to focus on the other player rather than on manipulating their own character. This suggests that TUIs may also have benefits for 3D object manipulation.

2.3 Our Approach

In this project we investigate how multi-touch and tangibles together can be used to manipulate objects in a 3D tabletop environment. Our specific research questions are;

- RQ1. How can multi-touch and tangibles be used to manipulate 3D virtual objects in a collaborative tabletop environment?
- RQ2. Which interaction method is more suitable for each of the sub-tasks involved in 3D object manipulation?
- RQ3. Is tangible, multi-touch interaction on a tabletop suitable for collaborative-competitive games?

To provide a context for this inquiry we adopted the block-stacking game Jenga [19]. To be successful playing the game very accurate manipulation of the blocks is required. While Jenga uses regular blocks, their movement and docking is not constrained. Therefore, we posit that interactions that are successful in Jenga will translate well to other 3D object manipulation tasks.

In order to do this we: designed a set of multi-touch gestures and tangibles suited to the task (Sect. 2.4); implemented an appropriate environment for the multi-touch and tangible interaction and verify its usability (Sect. 2.5); carry out a realistic user evaluation of the environment (Sect. 2.6).

2.4 Observational Study

A Wizard of Oz [5, 6] observational study was conducted to understand what gestures and tangibles people find intuitive when manipulating 3D objects through a 2D interface. There were seven participants in the study, four male, and three female, their ages ranged from 22 to 35, all were experienced touch device users.

2.4.1 Method

The study used a real stack of Jenga [19] blocks placed underneath a transparent acrylic sheet (Fig. 2.2). The acrylic sheet acted as a 2D screen, on top of which users could make gestures with their fingers, hands and tangibles. The users explained what they expected to happen as they performed actions, the real Jenga blocks were manipulated by the facilitator to match the participant's verbal instructions. A number of tangibles were provided, including a stack of Jenga blocks, a single Jenga block, and two unspecified objects that the users could assign meaning and actions to—for example a user could say 'this is a magnet and a block sticks to it when I place it on the block'.

Data was collected with pre and post-task questionnaires, a second facilitator observed each participant and multiple cameras were used to record participants' gestures.

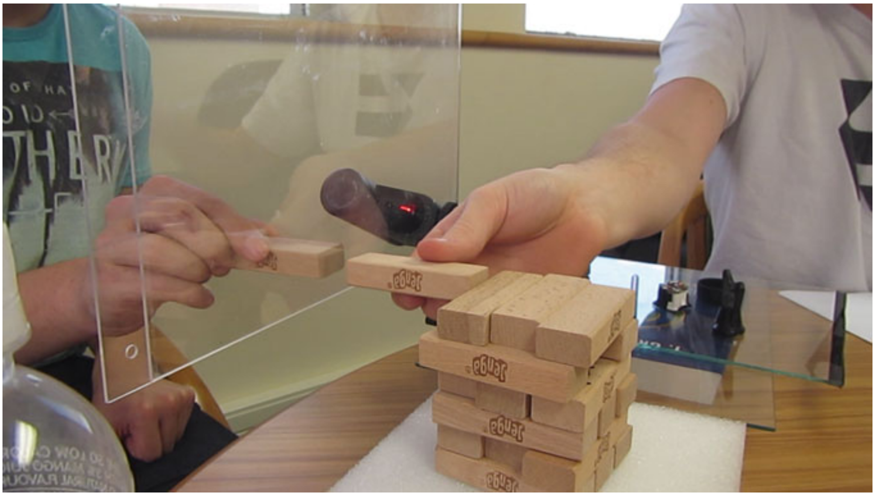


Fig. 2.2 Wizard of Oz interaction example with a user on the *left* simulating a tangible interaction and a facilitator on the right following the user's instruction to move a block

2.4.2 Tasks

Participants performed three sets of tasks. To familiarize the participants with the objects the first set of tasks was completed on the physical blocks. Task sets two and three, touch gestures and tangibles respectively, were performed on the acrylic sheet. Task sets two and three were given to participants in different orders to reduce order effect bias.

Each set of tasks consisted of sub-tasks split into sections:

- Change the camera view of the stack (4 variations of top and side view) (task sets 2 and 3 only)
- Single block manipulation
  - Rotate block −4 variants of direction and degrees x2 (top and side view).
  - Move block on the horizontal and vertical plane −8 variations x2 (top and side view).
  - Remove block from the stack −3 variations, side, and center blocks x2 (top and side view).
- Complete task
  - Move an edge block to the top of the stack (x2 starting from top or side view)
  - Move a center block to the top of the stack

In total, each participant completed 104 tasks.

2.4.3 Results

The results showed common themes for a number of block manipulation tasks. To translate blocks left/right/up/down participants almost always used a single finger drag, and to rotate blocks participants, by and large, used the two-finger rotate gesture. The most commonly used tangible for these tasks was the block tangible. It was placed on the virtual block and moved and rotated to perform these actions.

The most difficult task in real Jenga is to remove a block from the center of a row because of the accuracy required not to move the surrounding blocks. Working through the acrylic sheet, it was clear that the only logical way to remove a center block from the stack is to move a block orthogonally to the screen. However, there was little consensus among participants for moving blocks orthogonally; many different solutions were given, participants often used previously used gestures or didn’t know how they would perform this action. Some suggested, so as to not overload gestures, the pinch touch gesture be used for this action. For the tangible equivalent, a ‘screwdriver’ tangible that is rotated to move the selected block orthogonally suggested by one of the participants seemed to have the most promise.

Our study design constrained movement of the camera view to either top or side positioning of the acrylic sheet. As we will discuss below much more flexible camera positions are easily achieved in the virtual environment. With the constraints we imposed, the most common touch gesture suggestion was the addition of slider-bar widgets. For the tangible manipulation, users suggested placing the tangible Jenga stack provided at the orientation required.

2.5 Virtual Jenga

To realize the results of the observational study, we implemented a multiplayer Jenga [19] block stacking game for a large tabletop display. To compare multi-touch interactions with tangible interactions, we designed and implemented two interaction schemes in the Jenga game: multi-touch based interaction and tangible based interaction. Each interaction scheme has a method for manipulating the camera view, selecting and deselecting blocks, moving and rotating blocks, and translating blocks orthogonally to the camera view.

2.5.1 Implementation

The Jenga game was implemented on a Microsoft PixelSense table, specifically the Samsung SUR40 [21]. XNA Game Studio 4.0 [14] was used to develop the game, as it integrates easily with the Microsoft Surface 2.0 SDK [15]. The Henge3D [10] physics engine was used with XNA Game Studio to provide the physics capabilities of the game.

The touch and tangibles were recognized with the Microsoft Surface 2.0 SDK core layer [16] which processes and recognizes three types of objects in contact with the screen: fingers, blobs, and byte tags (Fig. 2.3).

To track the tangibles we initially used byte tags, these worked fine when the position and orientation weren't needed (e.g. for the stack tangible), however, they gave unstable readings for the position and orientation of objects. To identify and track the position and orientation of the tangibles more accurately, we used blob pairs [27], which consist of a small blob and a large blob. By drawing a line between the centers of these two blobs, the orientation of the tangible is able to be

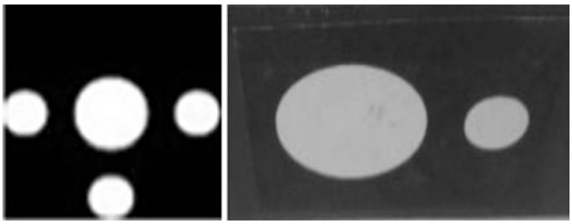


Fig. 2.3 Byte tag, blob pair used to track a tangible's position and orientation

determined. Blob pairs are uniquely identified by the width of the two blobs and the distance between the blobs.

User's interactions with the system are logged into a CSV file for later analysis. Each row contains the player currently interacting with the system (specified with a button on screen), the type of interaction and the time that this occurred. This data together with video recordings allow us to analyze the users' interactions.

2.5.2 Touch and Tangible Interaction

The touch and tangible gestures used in our virtual Jenga game were developed through a process of iterative refinement. This began by using the results of the Wizard of OZ observational study to create the first prototype. As we developed the system more alternatives were investigated and informally tested. The prototype was then evaluated with a usability test; six participants undertook this study individually. We then refined the interaction based on the results of the usability test before evaluating the final prototype with a Jenga tournament that is reported in Sect. 2.6. This section describes the design decisions behind the interaction schemes for touch and tangible interaction and how they evolved during development and usability testing.

2.5.2.1 Camera View Manipulation

Touch

Our observational study camera view was constrained to two views, top and side. However, most 3D development kits include infinitely flexible camera manipulation. Initially, we explored using touch interaction to manipulate a free-flying camera, this allowed users to move forwards and backward as well as move the camera up, down, left and right. We decided against this design because having to manipulate too many camera axes, distracted users.

The next iteration constrained the camera by fixing the focal point to the Jenga tower. Using slider-bars, as suggested in the observational study, the camera orbits around the Jenga tower. From the user's perspective, the Jenga Tower appeared to rotate as the sliders were manipulated. Informal testing suggested that the slider bars were a step in the right direction but still caused a disconnect between the users and the game. Users would look away from the stack, to use the slider bars and then look back to the stack again. This tended to take the user out of the game mentally. In addition, the slider bars introduced unnecessary widgets thus increasing interface complexity.

The final iteration combined features of the previous two interactions. Focus is fixed on the Jenga stack and the camera is manipulated with a finger drag (Fig. 2.4a): dragging your finger left or right rotates the camera around the Jenga stack, whilst dragging your finger up or down rotates the cameras vertically around

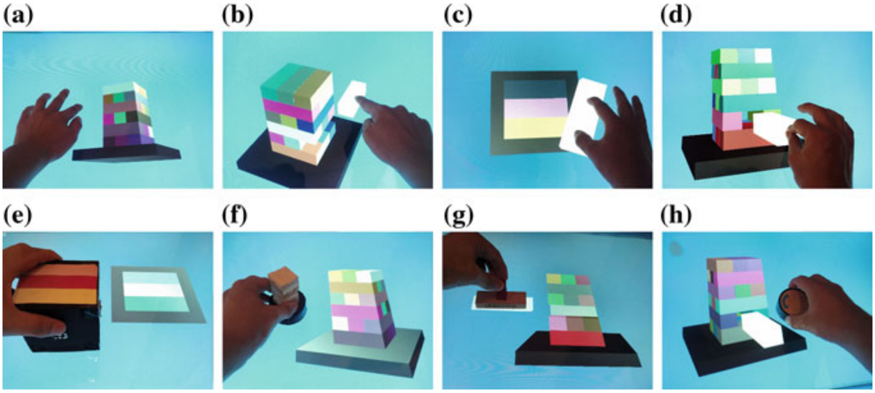


Fig. 2.4 Touch: **a** manipulate camera, **b** move block, **c** rotate block, **d** move block orthogonal. Tangibles: **e** stack—snap camera views, **f** Jenga tower—fine grain camera control, **g** Jenga block—move and rotate virtual block, **h** corkscrew—move orthogonally

the stack. There are no limits when rotating the camera around the stack, however, the vertical rotation is restricted to 90° (directly above to horizontal), so that the table does not obscure the Jenga stack. A gesture can move the camera in both dimensions at the same time. Camera manipulation can be done anywhere on the screen aside from on a selected block and with any number of fingers. This approach to camera movement proved successful in both the usability study and Jenga tournament.

Tangibles
Two tangibles for movement of the camera were implemented. The first is the stack tangible (Fig. 2.4e). This tangible was used in the user study by many of the participants. Placing the stack tangible on the table in a particular orientation snaps the view to one of the five faces (not the top as this would turn the view upside down). This is the only tangible which used the SDK byte tags for recognition. As mentioned earlier byte tags are useful in the case where position and orientation are not needed. For the stack tangible, all the information needed is which face of the tangible is on the table. A different byte tag was placed on each face of the tangible.

The second camera tangible is the fine grain camera stack (Fig. 2.4f), which gives the user fine grain control of the camera. Spinning the fine grain tangible rotates the view around the stack. In addition, moving the tangible up, down, left or right moves the camera view, in the same way a finger swipe would with the same gesture. For example, moving the tangible to the left rotates the camera in an anti-clockwise direction.

2.5.2.2 Block Manipulation

Blocks need to be selected, translated (moved) and rotated. Translation is required in the 2D plane of view and orthogonally to the view.

Touch
Initially translation on the 2D plane was implemented with a touch and drag. However, we found that this would result in users unintentionally moving a block. Therefore double-tap to select was implemented as suggested by a number of participants in the observational study. A single-tap was explored, however because of the number of false positive tap events that can be produced by the table, the double-tap was more reliable. Once a block has been selected, it is able to be moved around using single finger drag (Fig. 2.4b), the movement is restricted to the plane that is parallel to the current view. A block remains selected until released with a double-tap. While selected it is not subject to the physics engine and can be left hanging in midair. We found this was necessary for users to alternate between block and camera manipulation. However, the movement of all the other blocks in the stack is still governed by the physics engine. A double-tap releases a block at which time the physics engine is applied to it and it will drop onto whatever is below it, the stack, table or floor.

Rotation of the blocks is based off well-established two-finger rotation gestures that are typically used on touch devices (Fig. 2.4c). This was popular during the user studies and found to be most intuitive given its familiarity.

The gesture for when a block was to be moved orthogonally, towards or away from the participant, caused the most problems for the participants in the observation study. Yet, this gesture is essential for pulling a block out of a row. We implemented a two-finger pinch to zoom style gesture (Fig. 2.4d). The gesture was initially designed to mimic the grabbing of a block between two fingers and pulling it towards you. However, during the usability study we found that users were confused, the pinch/stretch gesture was reversed for consistency with mobile phones; a pinch is used to zoom out and a stretch to zoom in.

Tangibles
Two main tangibles were developed for manipulating blocks. The first tangible is the Jenga block (Fig. 2.4g). Placing the physical block on a digital block will select it and then movements of the physical block are mirrored onto the digital block.

In order to ensure a close mapping between the tangible and the virtual block, we had to decide how they would snap together. If the virtual block snaps directly to the middle of the physical block, it causes problems when attempting to slowly remove a block from the stack. For example, if a user places the physical block with the midpoint of the physical block on the edge of a virtual block, it causes the virtual block to snap half the length of the block to the center. The result of this is that the virtual block leaps to that particular position often causing the tower to topple. An alternative solution is that no snapping occurs. However, the problem with this is that the virtual Jenga block doesn't necessarily align with the physical Jenga block. Our final solution was that the virtual block slowly tracks to the correct position over a few seconds. In this way, it is almost undetectable by the user but by the time they remove a block from the stack the virtual block aligns perfectly with the physical block.

The block tangible automatically selects the underlying virtual block when it is placed on it, but does not release it automatically for two reasons. First, users are often uncertain of whether they are ready to release a block. Second, the blob tracking can sometimes fail for a moment, if the virtual block is not released this failure does not affect the user's actions. After the usability testing, we discovered that users needed a method consistent with the touch interaction for deselecting a block with a tangible. To do this we settled on a single-tap with the Jenga block; a double-tap would have been more consistent to the touch double-tap, but a single-tap was more reliable.

The second tangible for block manipulation was the corkscrew tangible (Fig. 2.4h). This enables the orthogonal movement of the block towards or away from the screen. Spinning the corkscrew tangible to the right moves the block into the screen and to the left moves the block out of the screen. The movement of the block is orthogonal to the camera view of the selected block. The corkscrew does not select a tangible but operates on the currently selected block. It does not need to be located on the block, it can be anywhere on the display.

2.5.3 Usability Study

The aim of the usability study was to verify the touch gestures and tangibles were easily understood and executable. We refined some of the interactions as reported above. However, we also observed which interaction method was used in given situations. For camera manipulation, almost all participants used a combination of the touch control and the fine grain camera manipulation tangible. The view snapping (stack) tangible was used rarely. The manipulation of the blocks in simple tasks was done primarily with the use of touch. But, interestingly, for the more difficult tasks the users tended towards using the tangible controls. Particularly, the users preferred using the corkscrew tangible for orthogonal movement of a block. The participants expressed that the tangibles were beneficial to their manipulation of the system in some manner. They said that the real-world objects gave them a greater sense of control and allowed for more deliberate actions.

2.6 User Evaluation

To evaluate how well the touch and tangible interaction methods work for manipulating 3D objects in a more realistic environment the user evaluation was in the form of a Jenga knockout tournament. Before any competitive play there was a general training session where all the touch and tangible gestures were demonstrated to the participants and each tried each of the gestures. The tournament was completed in one session that lasted 3 h. Once participants were beaten, they were out of the tournament and were free to either leave or remain and watch. All but two remained until the end. Participants were free to choose how they interacted with the virtual Jenga game: they could use any combination of touch or tangibles. The participants were trained so that they had experience with both touch and tangibles and tried each interaction method before playing in the tournament. There were a number of prizes to motivate participants to choose the interaction method that they felt would best help them win: first prize \$50, second prize \$30 and third and fourth prizes \$20 each.

2.6.1 Participants

There were 9 participants, 2 females, and 7 males, aged from 21–42 (median 28) all were right-handed. All of the participants had previous experience with touch interaction and touch interaction on large touch devices. The participants collectively had much less experience manipulating 3D objects on a normal computer, with touch interaction or with tangibles. An overview is given in Fig. 2.5.

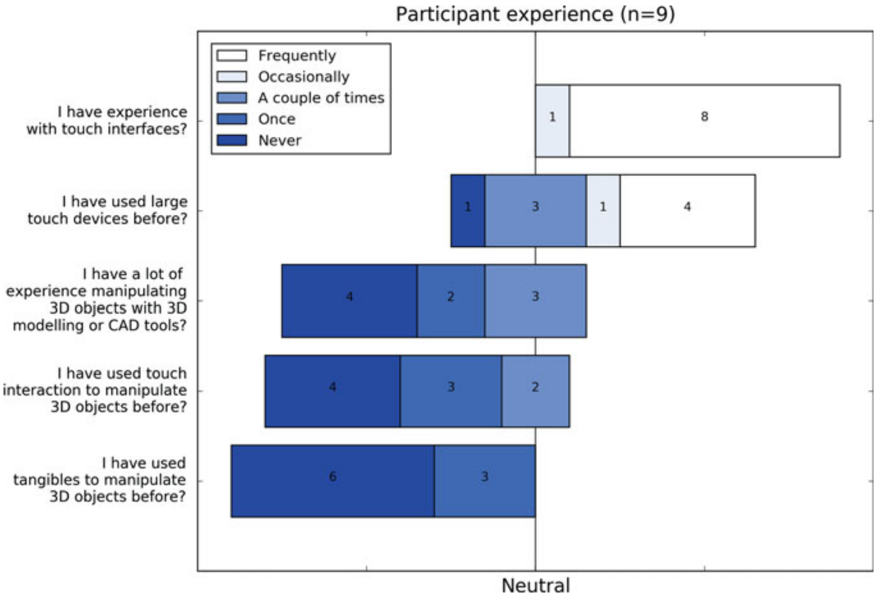


Fig. 2.5 Participants' experience

2.6.2 Tasks

The tournament was structured into three rounds. The first round had each participant compete against another participant. The 9th participant played one of the winners of the first round and the winner of this game progressed to the next round. The four winners of the first round then competed against each other in the second round and the two remaining winners competed against each other in the final round. The first round started with a practice game so that participants could get used to the interaction methods and rules.

2.6.3 Data Collection

A pre and post-task questionnaire was administered to participants. The interactions with the virtual Jenga game were logged providing detailed statistics of what touch gestures and tangibles were used, the time period each was used, and the number of transitions between touch and tangibles. The tournament was videoed. One facilitator ran the tournament and managed the participants while a second facilitator observed and took notes of how participants interacted with the virtual Jenga game.

2.6.4 Data Analysis

The Likert scale responses from the questionnaires and the interaction data logs were analyzed using Jupyter Notebook Pérez and Granger [18]. The freeform questionnaire responses were analyzed by grouping responses with similar themes together, similar to the open and axial coding techniques from Grounded Theory Glaser and Strauss [7].

2.6.5 Results

A number of themes emerged from the data. At a high level, participants found both interaction methods to be generally usable, illustrated by the similar positive Likert scale responses for both touch (Fig. 2.6) and tangibles (Fig. 2.7). The questions related to ease of use, learnability, responsiveness, accuracy and whether participants would use the particular method again.

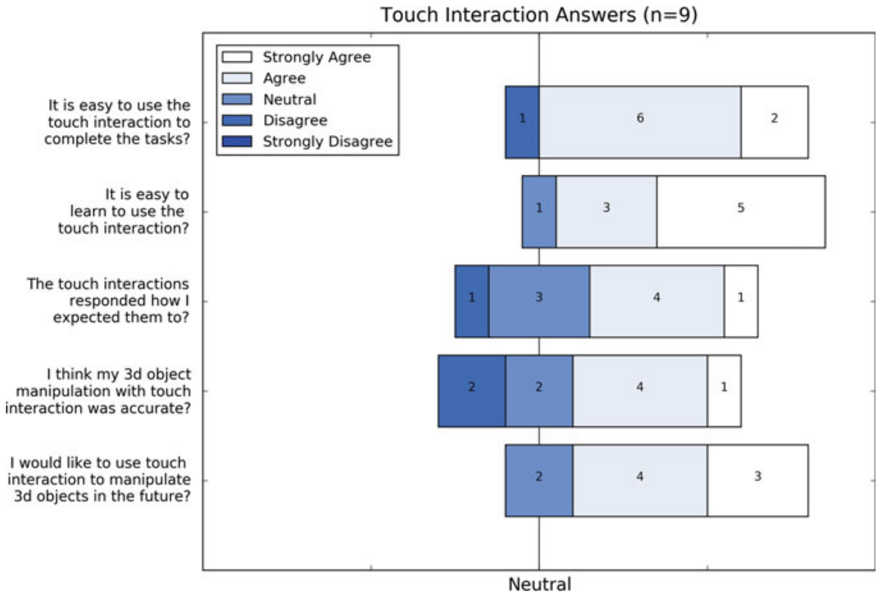


Fig. 2.6 Touch interaction Likert scale results

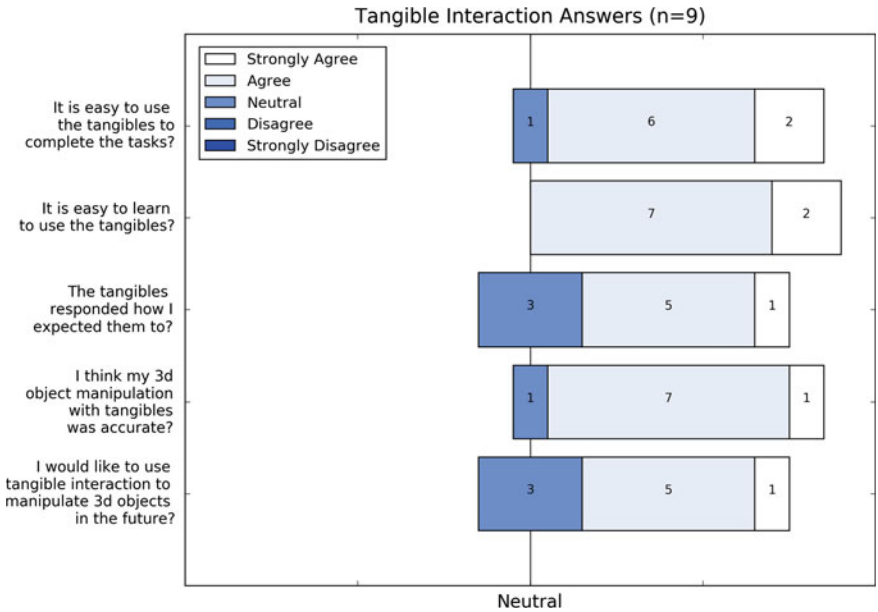


Fig. 2.7 Tangible interaction Likert scale results

2.6.5.1 Touch Interactions

Positive themes regarding touch interaction included, it is easy to use (P1, P6, P7), specifically selecting objects (P9) and moving the camera (P8); it is easy to learn because the gestures are similar to those used on mobile phones (P2, P3, P5); and it is more efficient than tangible interaction as you only need your fingers (P8).

Negative themes regarding touch included poor accuracy (P2, P3, P5, P6, P8), specifically, when rotating blocks (P5, P8) and when moving blocks orthogonal to the screen (P9). To address these areas of interaction, participants thought that using a capacitive screen rather than the SUR40 [21] infrared tabletop would increase accuracy and be able to use the pinch to zoom gesture on any part of the screen would make moving blocks orthogonal to the screen easier (P5).

2.6.5.2 Tangible Interactions

Positive themes regarding tangibles include they are easy to use (P1, P9), specifically to move objects (P1) and change views (P9); they are accurate (P2, P3, P8), presumably, these participants were referring to the corkscrew tangible as it is the only tangible that was used more than touch gestures (Fig. 2.10); lastly, some users appreciated the physical nature of the tangibles, specifically having something to hold onto (P5) and helping them to understand the 3D space (P4).

Negative themes regarding tangible interaction include poor sensing accuracy (P4, P5, P6), especially the stack tangible (P5); a higher learning curve (P2, P5); and there being too many tangibles, which is inefficient (P8) and it makes it hard to remember what they do (P3). To address these issues participants thought that objects could be labeled better (P4) and that the number of tangibles should be reduced (P4, P8).

2.6.5.3 Comparisons Between Touch and Tangibles

The participants were asked to compare the touch and tangible interaction methods (Figs. 2.8 and 2.9), we also logged the time users spent using each touch gesture and each tangible (Fig. 2.10). Two key themes emerged from this data: most users preferred touch interaction; however, the corkscrew tangible was preferred over its touch counterpart.

The Likert scale data slightly favored touch interaction in terms of overall rating (Fig. 2.8) and whether touch interaction was easier to use than tangible interaction (Fig. 2.9, Q1). The participants favored tangible interaction when asked to rate the accuracy of the interaction methods (Fig. 2.9, Q2); however, the only tangible used more than touch interaction was the corkscrew tangible (Fig. 2.10), so this could be in reference to this one tangible.

When participants were asked what interaction method they would prefer if they had to pick one; 7 said they would prefer to use touch interaction over tangible

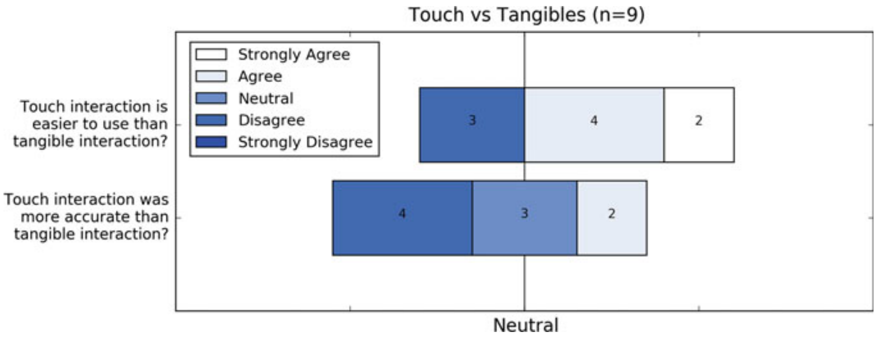


Fig. 2.8 Touch versus tangibles: overall ratings

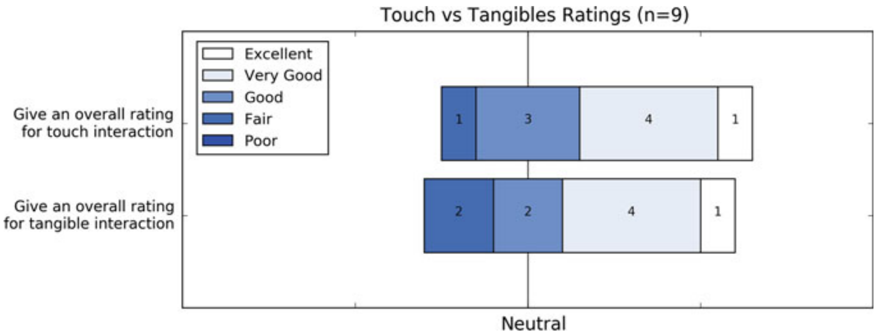


Fig. 2.9 Touch versus tangibles: ease of use and accuracy

interaction, whilst 2 would choose tangible interaction over touch. This reinforces the theme that touch is preferred over tangibles. The data logging also supports this theme, showing that participants used touch interaction much more than tangible interaction in almost all categories, including for moving the camera, translating and rotating blocks and selecting and deselecting blocks. The one area where participants used tangible interaction more than touch interaction was translating a block orthogonal to the screen with the corkscrew tangible.

The last theme that emerged is that 3D control can still be difficult regardless of which interaction scheme is used (P2, P4); specifically, it can be difficult to understand 3D space with the application (P4) and movement relative to the camera is confusing—movement relative to the world may be better (P2).

One observation that surprised us was how quickly some users adapted to two-handed interaction, moving fluidly between touch and tangibles. Figure 2.11 shows the number of transitions between touch and tangibles by each participant. P1, P4, P8, and P9 used a tangible in one hand and touch gestured with the other to rapidly transition between moving the camera and the block; an example is illustrated in Fig. 2.12.

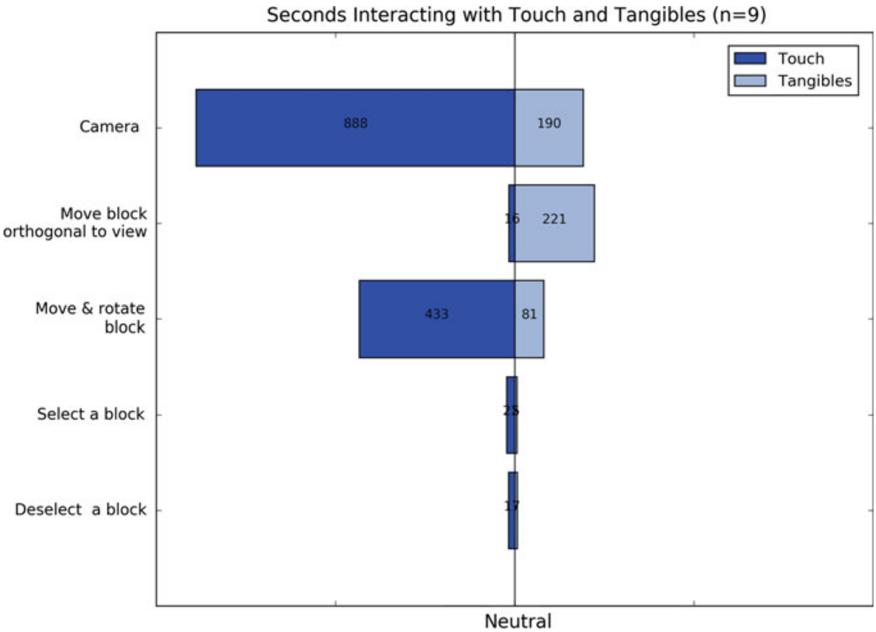


Fig. 2.10 Touch versus tangibles: time breakdown

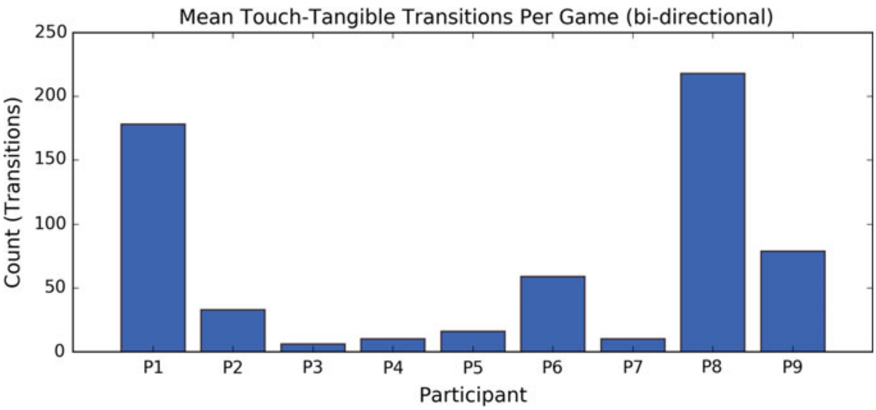


Fig. 2.11 Mean touch-tangible transitions per game (bi-directional)

We noted both collaborative and competitive behavior during the tournament. During a game, the two players would stand close to the table while other people in the tournament watched on from a little further away. The players would swap in and out of the prime interaction position (at the side of the table) as they took turns. The current player would sometimes ask for advice with comments like 'how am I going to do that?' and at other times set out a challenge 'if I take this block out it

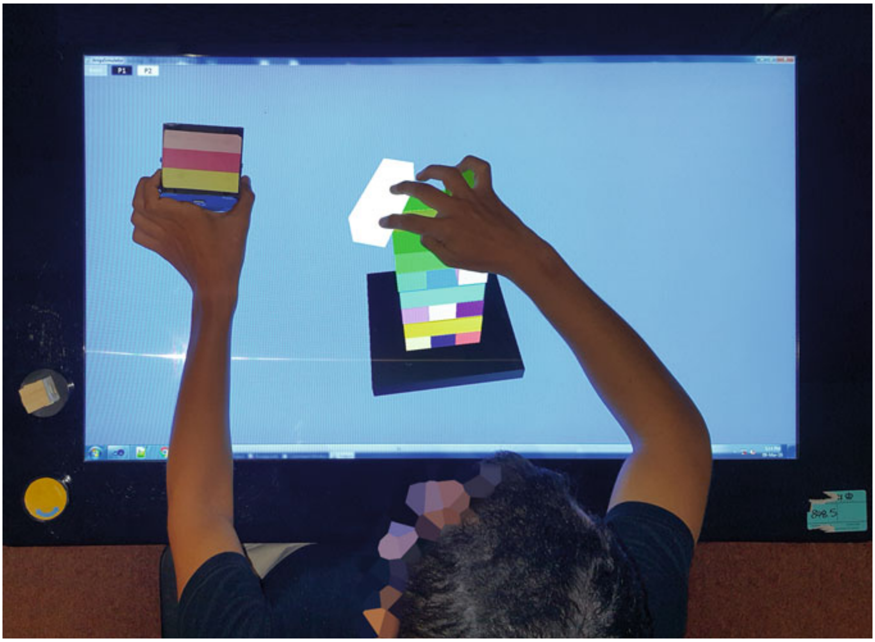


Fig. 2.12 Two-handed interaction example

will make it difficult for you'. Their opponent and other players observing both offered advice and narrated. They made suggestions about how and which block to move next and where to place it. They also commented on the state of play with comments such as 'you can win now'.

2.7 Discussion

The goal of this project was to explore how tangibles together with multi-touch could be used to manipulate objects in a 3D scene. Touch was more familiar to our participants as they were all experienced touch device users. However, they also enjoyed the tangibles and found the interaction to be generally on a par with touch. When asked to choose between the two, most chose touch, this is likely because of familiarity.

We note that users preferred to use touch for most tasks. The exception being orthogonal movements of the blocks where the corkscrew tangible was a strong preference. It could be that a different touch gesture to the pinch to zoom would score better. However, our observational study and explorations during implementation did not uncover a better alternative. Another alternative, suggested by a participant, is to give users the ability to use the pinch to zoom gesture on any part of the screen, rather than just on the block, while we did not trial this, it seems counter intuitive.

Users experienced sensing problems for both touch and tangibles and this negatively affected the results for both in different ways. The stack tangible is the only tangible using the SDK byte tags for recognition. As it was rarely used in the usability study we did not realize how poor its recognition rate is compared to the blob tags. It often did not register, so while users in the tournament tried to use it, they stopped in frustration. Had it worked better, we think that it would have been used much more frequently for aligning the stack to the front view as this is easier and more accurate than touch gesture alignment. The main touch sensing problems occurred caused blocks to jump around and vibrate when moving and rotating them (P5). Use of blob tags, better hardware, and gesture recognizers would solve these sensing problems.

This is the first project to explore using both multi-touch and tangibles, and the first to explore tangibles for 3D object manipulation. It is likely that providing two or more options for the users to complete any task confused some of the user study participants with some claiming there were too many gestures and too many tangibles.

We were surprised at how quickly some participants moved to 2-handed interaction with one hand holding a tangible and the other used for touch. We think that the different affordances of touch versus tangible were quite helpful in this respect. If, as P2 did,[1] the left hand is holding the stack tangible and the right hand used for touch gestures, the affordance of the tangible could be helping balance the cognitive load and reducing the cognitive load of remembering interactions. This is an interesting outcome of the current study that requires further research.

Given the results of the study and the user feedback, for this particular context, we believe it would be optimal to provide two tangibles and six touch gestures. One tangible would be used for camera positioning, with some redesign the functionality of the two camera tangibles could be combined. The second tangible would be used to move blocks orthogonally to the view. The first touch gesture moves the camera as described above. Another set of touch gestures covers block manipulation: double-tap to select, drag to move in the 2D plane, two touch to rotate and tap to release. This combination of tangibles and gestures provides for two-handed interaction and also clearly separates moving blocks on the two different planes.

In this project, we explored two aspects of 3D object interaction. First, there is the need to manipulate the camera view of the world. In an actual game of Jenga, you typically walk around the Jenga tower in order to see what’s on the other side. However, this isn’t possible on a flat display. On a table, it would be technically possible to track a person moving around the table and alter the view in sync with their position. However, it is more flexible to move the camera and therefore, gesture and tangibles were developed for this purpose. Second we developed

[1]This wasn’t picked up by the data logger as much as it should have because the stack tangible was not always detected.

gestures and tangibles for moving objects. Although the Jenga blocks are regular cuboids we believe that the interaction techniques could be applied to any objects in a 3D environment.

Camera control and object manipulation are just two of the controls need for 3D object interaction [11]. Comprehensive interaction would include many other interactions such as complex world navigation, path drawing, and object modeling. To do this with touch alone would require mode changes and overloading of gestures, both of which are generally detrimental, or a complex set of gestures, which are difficult to remember. By adding tangibles to the interaction mix the affordances of the tangibles may make a comprehensive set of interactions both more memorable and easier to use.

2.8 Conclusion

This project explored using touch and tangibles together to manipulate objects in a tabletop 3D virtual environment. These environments are designed for collaborative and playful tasks so we adopted the block building game Jenga as our context. Jenga has the advantages of requiring precise object manipulation and physics alone determining the state of the stack.

While our initial Wizard of Oz observational study guided the development of the touch gestures and tangibles there was little consistency between participants for the most challenging interactions so iterative exploration and testing were required during development.

Returning to our research questions:

RQ1 We found that both methods of interaction were generally usable in the final prototype and acceptable to users. The multi-touch gestures were preferred however, this is partly due to their familiarity.

RQ2 We observed that touch and tangibles can be seamlessly used together for 3D object interaction. Over half of the participants frequently switched between touch and tangibles—some doing so with two-handed interaction—one hand holding a tangible and the other used for touch gestures. While for most tasks touch was the most used method, to move a block orthogonally the tangible was preferred.

RQ3 The tabletop, and interaction methods combined to provide an excellent environment for the 3D competitive-collaborative play.

Providing comprehensive 3D interaction through 2D interfaces has many facets. The ideas explored here could be extended to address other 3D world functionality. In particular combining touch and tangibles may reduce the need for mode changes.

References

1. Aish R et al (2001) Computational construction kits for geometric modeling and design. In: Panel abstract, I3D symposium
2. Baudisch P, Becker T, Rudeck F (2010) Lumino: tangible blocks for tabletop computers based on glass fiber bundles. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 1165–1174
3. Bock M et al (2015) Tangible widgets for a multiplayer tablet game in comparison to finger touch. In: Proceedings of the 2015 annual symposium on computer-human interaction in play. ACM, pp 755–758
4. Cuendet S, Bumbacher E, Dillenbourg P (2012) Tangible versus virtual representations: when tangibles benefit the training of spatial skills. In: Proceedings of the 7th nordic conference on human-computer interaction: making sense through design. ACM, pp 99–108
5. Dahlbäck N, Jönsson A, Ahrenberg L (1993) Wizard of Oz studies: why and how. In: Proceedings of the 1st international conference on intelligent user interfaces. ACM, pp 193–200
6. Frishberg N (2006) Prototyping with junk. Interactions 13(1):21–23
7. Glaser BG, Strauss AL (1967) The discovery of grounded theory: strategies for qualitative research. Transaction Publishers
8. Hand C (1997) A survey of 3D interaction techniques. In: Computer graphics forum. Wiley Online Library, pp 269–281
9. Held R et al (2012) 3D puppetry: a kinect-based interface for 3D animation. In: UIST. Citeseer, pp 423–434
10. Henge3D, Henge3D Physics Library for XNA—Home. https://henge3d.codeplex.com/
11. Jankowski J, HachetM (2015) Advances in interaction with 3D environments. In: Computer graphics forum. Wiley Online Library, pp 152–190
12. Jordà S et al (2007) The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In: Proceedings of the 1st international conference on Tangible and embedded interaction. ACM, pp 139–146
13. McCord G et al (2008) A pen and paper metaphor for orchid modeling. In: GRAPP. pp 119–124
14. Microsoft, Download Microsoft XNA Game Studio 4.0 from Official Microsoft Download Center. https://www.microsoft.com/en-nz/download/details.aspx?id=23714
15. Microsoft, Microsoft Surface 2.0 SDK. https://msdn.microsoft.com/en-us/library/ff727815.aspx
16. Microsoft, Presentation Layer versus Core Layer Applications. https://msdn.microsoft.com/en-us/library/ff727793.aspx
17. Oh J-Y, Stuerzlinger W (2005). Moving objects with 2D input devices in CAD systems and desktop virtual environments. In: Proceedings of graphics interface 2005. Canadian Human-Computer Communications Society, pp 195–202
18. Pérez F, Granger BE (2007) IPython: a system for interactive scientific computing. Comput Sci Eng **9**(3):21–29. http://ipython.org
19. Pokonobe Associates, Jenga. http://www.jenga.com/
20. Ruckenstein M (1991) Homo ludens: a study of the play element in culture. Leisure and Ethics 237
21. Samsung, Samsung 40 SUR40 SMART Signage—Samsung Business UK. http://www.samsung.com/uk/business/business-products/smart-signage/professional-display/LH40SFWTGC/EN
22. Shaer O, Hornecker E (2010) Tangible user interfaces: past, present, and future directions. Found Trends Hum-Comput Inter 3(1–2):1–137
23. Strauss PS, Issacs P, Shrag J (2002) The design and implementation of direct manipulation in 3D. In: SIGGRAPH Course Notes 12

24. Ullmer B, Ishii H (2000). Emerging frameworks for tangible user interfaces. IBM Syst J 39 (3.4):915–931
25. Ullmer B, Ishii H (1997) The metaDESK: models and prototypes for tangible user interfaces. In: Proceedings of the 10th annual ACM symposium on User interface software and technology. ACM, pp 223–232
26. Whiteley B, Blagojevic R, Plimmer B (2015) Tangible-tango: designing and fabricating tangibles with tangibles. In: 16th australasian user interface conference, AUIC 2015, Sydney, Australia, Jan 2015. pp 31–39. http://crpit.com/abstracts/CRPITV162Whitely.html
27. Zhen SJ, Blagojevic R, Plimmer B (2013). Tangeo: geometric drawing with tangibles on an interactive table-top. In: CHI'13 extended abstracts on human factors in computing systems. ACM, pp 1509–1514

Chapter 3
Spontaneous Gesture Production Patterns on Multi-touch Interactive Surfaces

Yosra Rekik, Radu-Daniel Vatavu and Laurent Grisoni

Abstract Expressivity of hand movements is much greater than what current interaction techniques enable in touch-screen input. Especially for collaboration, hands are used to interact but also to express intentions, point to the physical space in which collaboration takes place, and communicate meaningful actions to collaborators. Various types of interaction are enabled by multi-touch surfaces (singe and both hands, single and multiple fingers, etc.), and standard approaches to tactile interactive systems usually fail in handling such complexity of expresion. The diversity of multi-touch input also makes designing multi-touch gestures a difficult task. We believe that one cause for this design challenge is our limited understanding of variability in multi-touch gesture articulation, which affects users' opportunities to use gestures effectively in current multi-touch interfaces. A better understanding of multi-touch gesture variability can also lead to more robust design to support different users' gesture preferences. In this chapter we present our results on multi-touch gesture variability. We are mainly concerned with understanding variability in multi-touch gestures articulation from a pure *user-centric perspective*. We present a comprehensive investigation on how users vary their gestures in multi-touch gestures even under unconstrained articulation conditions. We conducted two experiments from which we collected 6669 multi-touch gestures from 46 participants. We performed a qualitative analysis of user gesture variability to derive a taxonomy for users' multi-touch gestures that complements other existing taxonomies. We also provide a comprehensive analysis on the strategies employed by users to create different gesture articulation variations for the same gesture type.

Y. Rekik (✉) · L. Grisoni
Mint, University of Lille Science & Technology, CNRS, INRIA, Villeneuve-d'Ascq, France
e-mail: yosra.rekik@inria.fr

L. Grisoni
e-mail: laurent.grisoni@univ-lille1.fr

R.-D. Vatavu
MintViz Lab—MANSiD Research Center,
University Stefan cel Mare of Suceava, Suceava, Romania
e-mail: vatavu@eed.usv.ro

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_3

3.1 Introduction

People exhibit inherent intrinsic variations for their gesture articulations because gestures carry dependency with both the person producing them and the specific context, social or cultural, in which they are being produced. In his psycholinguistic studies on human discourse and relationship between gesture and thought, Mcneill [1] considers that "*gestures are the spontaneous creations of individual speakers, unique and personal*" and that gestures "*reveal the idiosyncratic imagery of thought*" (p. 1). More than cultural dependency, gestures are also deeply intertwined with speech, which makes the lexical and syntactic structures of language also affect the specific forms in which gestures are being produced [2]. The user-dependency aspect of gesture production has been many times reflected by previous work that analyzed users' gesture preferences in conjunction with specific gesture sensing technology, such as interactive tabletop surfaces [3, 4], accelerated movements [5], and freehand gestures [6–8]. These studies, generally referred to as "gesture elicitation studies," have shown that some level of consensus exists between users due to similar conceptual models that users naturally seem to construct when thinking about common interactive tasks. However, these studies also pointed out many variations in users' preferences for gesture commands, with probably the most important finding being that users prefer different gesture commands than those proposed by experienced designers [3].

For the specific case of multi-touch input, there are many degrees of freedom that can be independently controlled during gesture articulation, such as the number of fingers or the finger types touching the surface [9], single-handed or bimanual input [10, 11], variations in the number of strokes forming the gesture [12, 13], and the use of additional modalities accompanying finger touch input leveraged by sensing pressure [14] and various parts of the finger anatomy [15]. In their user-defined surface gestures study, Wobbrock et al. [4] captured the many degrees of freedom aspect when noting that "*surface gestures are versatile and highly varied—almost anything one can do with one's hands could be a potential gesture*" (p. 1083). Indeed, our recent work experimentally confirmed variation in multi-touch gesture articulation, and reported the many ways in which people naturally introduce variation for surface gestures when not being constrained by limitations imposed by the interface or the recognizer's ability to discriminate between gesture types [13]. At the same time, the versatility of multi-touch input makes prototyping multi-touch gesture recognizers a difficult task because, in many cases, "*the programming of these [multi-touch] gestures remains an art*" [16, p. 2875]. We believe that one cause for this recognition challenge is our limited understanding of *variability in multi-touch gesture articulation*, which affects not only recognition performance but also users' expression possibilities in current multi-touch interfaces. For example, a better understanding of variability could benefit interface design beyond achieving high-performant recognition, toward more fluent and expressive interactions able to exploit the explicit signals contained within the variability of the articulation [17] and could lead to more accurate multi-touch gesture recognizers [18]. In addition, a better understanding of variability can improve multi-user design in which users interact simultaneously or collaborate to define a gesture in a specific task.

In this chapter, we advocate for the need of more in-depth user studies to better understand how users using multi-touch gestures. As gestures are versatile, we argue for designing interaction techniques that support many-to-one mappings between gestures and commands. In our research methodology, we think that a good start point is to explore the variability of multi-touch gesture input from a user-centric perspective. We conducted a pair of experiments to investigate and to understand how users produce multi-touch gestures. We employed quantitative and qualitative methods to understand the variability of multi-touch gesture articulation. In our previous work [13], we presented a first study to understand the variability of users' multi-touch gesture articulations and we leverage a taxonomy of multi-touch gestures and introduced the concept of atomic movement. Building on these results, we extend our previous experiment in this work to understand whether our findings are consistent and robust for a larger gesture set and new participants. We first compute the number of variations users can propose when they are not constrained. We then provide an in-depth analysis to characterize in a comprehensive manner the strategies employed by users to produce different articulations for the same gesture type. Our results also include subjective and qualitative feedback informing about users' multi-touch gesture preferences.

3.2 Related Work

Supporting users' wide range of multi-touch gesture articulation behavior has been previously noted as an important design criterion to deliver increased flexibility and a high-quality user experience [19]. This fact has led to a number of orthogonal design implications [4, 20, 21]. For example, principled approaches provide basic building blocks from which gesture commands are derived, such as gesture relaxation and reuse [22], fluidity of interaction techniques [23], and cooperative gestures [24]. Other researchers advocated for assisting users in the process of learning multi-touch gestures during actual interaction by proposing gesture visualizations [20], dynamic guides [25], and multi-touch menus [9, 26]. At the same time, other user-centric approaches advocate for enrolling users right from the early stages of gesture set design [4, 5, 27]. Such participatory studies revealed interesting findings on users' behaviors in articulating gestures as well as on users' conceptual models of gesture interaction. This previous work also recommends flexible design of gesture commands to accommodate variations in how users articulate gestures. Oh and Findlater [21] went further and investigated the feasibility of user-customizable gesture commands, with findings showing users focusing on familiar gestures and being influenced by misconceptions about the performance of gesture recognizers. Rekik et al. [28] examined user's perceived difficulty of articulation multi-touch gestures.

To deal with users' variations in multi-touch input, in a previous work [13], we presented the first investigation toward understanding multi-touch gesture variability. We described a general taxonomy to understand users' gestures, and to derive implications of users' variability of gesture articulation. Our taxonomy was the result of a user-centric study giving new insights into the different possible articulations of unconstrained multi-touch gestures. In that study, we considered a small set of 8 gesture types and we explicitly fixed the number of variations that users were asked to produce for every gesture. While this was sufficiently sound to elicit users'articulation, it also gave rise to further questions, which we address in this work. More specifically, we are interested in the following related questions: (1) the number of variations a user would be able to propose; and (2) what strategies users adopt to articulate the same gesture type within unconstrained multi-touch input.

3.3 Spontaneous Gestures

We define *spontaneous gestures* that are produced by users under unconstrained articulation conditions, i.e., users have the total freedom in creating such gestures without any instructions. This concept and definition allows us to capture the versatility of multi-touch gestures in a faithful manner with respect to users' actual intentions. We believe that spontaneous gestures are important for multi-touch interfaces since they deliver a more pleasurable experience by not constrained users to conform to and follow specific articulation patterns. Spontaneous gestures enable us to understand how users are actually transforming a geometric gesture shape into a motor articution plam. Such a fundamental understanding allows to abstract away from existing gestures and to leverage existing multi-touch input by incorporating more general concepts related to users' gesture articulation behavior thus ending up with more flexible and powerful interaction techniques.

In the following, we present the results of two experiments to understand spontaneous gestures. Our analysis was conducted in light of our previous work [13] in order to illustrate and to confirm its predictive power for new people and new gesture types. We first recall the open-ended experiment in which we introduced the concept of atomic movement and established our taxonomy [13]. Then, we present our goal-oriented experiment from which we report strategies employed by users to create spontaneous gestures and we discuss users'gesture preferences.

3.4 Open-Ended Spontaneous Gestures

We report in this section the results of the first task of the experiment conducted in [13]. Our goal in that experiment was to observe and analyze users' unconstrained multi-touch gestures. We asked 30 participants to produce as many gestures as possible that came to their mind such as gestures that had a meaningful sense to them or gestures that they would use to interact with applications. In addition, participants were asked to describe the gesture they performed using the think-aloud protocol.

3.4.1 The Concept of Atomic Movements for Gestures Production

A recurrent observation regarding participants gesture input behavior was that participants grouped their fingers into unitary blocks that moved in a consistent manner. We found that the number of contact fingers did not impact their movements, as long as fingers were close together. One interesting observation was that the notion of finger proximity is relative to the gesture type and also to user-proper referential and seems to be hard to define in absolute and universal manner from a system point-of-view. Users referential can in fact be substantially scaled up or down from the performance of one gesture to another one. However, the referential tends to stay constant and consistent over time and through continuous movements composing the same gesture. For example, one participant used two hands simultaneously with multiple fingers in contact with the surface to draw a circle such as each hand was drawing half of the circle. The same participant used both hands simultaneously moving from the top to the bottom of the surface to denote that he was translating all images that were in-between his hands. For these two examples, the relative distance between fingers composing the same movement is different: in the first example, it represents the distance between the fingers of the same hand, but in the second example, it represents the distance between the two hands, which can cover all the surface width.

To explain these behaviours, we introduce the notion of "*atomic movement*" which reflects users' perceptions of the undividable role that a group of fingers is playing when performing a gesture. From our observations, atomic movements are mostly in reference with the imaginary trail of a group of fingers. An atomic movement has an internal state that can change depending on hands shape, fingers arity, velocity, direction, etc. However, state changes do not alter the role an atomic movement is playing in users' minds and their primary intentions. Atomic movements are often mapped to global strokes in symbolic gestures, but they also capture more abstract movements implied globally by a whole set of fingers. In the particular case of users performing a symbolic gesture, users do not mind about the trail of each individual finger; instead they seem to view the atomic movement produced by a group of fingers as a single stroke without consideration to the actual individual strokes produced by each finger. For more abstract multi-touch gestures, fingers' atomic movements express a global meaning that users convey. In all cases, the stroke or the trace of individual fingers considered separately are not an important issue from the user's atomic movement perspective, which contrasts with the system perspective when processing and interpreting multi-touch input. From our observations, we distinguish between two classes of movements depending on whether (i) the multi-touch path corresponding to fingers is stationary or (ii) the multi touch path implies an embodied motion. As practical examples, variable number of fingers, from one or both hands, moving together following the same path or being held stationary to delimit or point a region on the surface, are among the most frequently observed atomic movements.

3.4.2 An Embodied Taxonomy of Multi-touch Gesture

To capture the space in which our participants produced gestures, we propose the multi-level layered taxonomy summarized in Table 3.1. The multiple levels of our taxonomy do not model separable attributes to be characterized individually. Instead, they represent the different aspects of a single unified dynamic mechanism employed by users in the production of a multi-touch gesture.

At the highest level of our taxonomy, we model the fact that a multi-touch gesture emerges the users' understanding of the gesture path before touching the surface. From this perspective, an external observer can only try to guess the *semantic concept* hidden in the user's gesture, since it might be the case that the gesture itself is not sufficient to fully reveal user's intention—an observation in accordance with previous studies [4, 29, 30]. From a neurological perspective, hands and fingers are controlled and coordinated by the human motor system to achieve a desired task. The *physicality* level thus captures the motor control allowing users to project the semantic level onto the interactive surface. Finally, the *movement* level is the practicalresult of the motor goal expressed by hands and fingers motions in order to infer unitary blocks composing the gesture.

The movement level is at the core of our model since it constitutes the interface between the user and the interactive surface. We propose to structure this level according to two generic classes built in a recursive manner. At the lowest level of the recursion, we find the class of gestures formed by an elementary atomic movement. An elementary atomic movement can be either of type stationary (Ref) or Motion as discussed previously. The *Compound* class refers to the recursive composition of a set of atomic movements. It is expanded in two classes according to the lifetime and the synchronicity of composing atomic movements. The *Parallel* class refers to users making two or more different but synchronous parallel atomic movements. This class engages relative finger motions as well as two-handed symmetric

Table 3.1 A taxonomy of multi-touch gesture

<table>
<tr><th colspan="3">SEMANTIC-CONCEPT</th></tr>
<tr><td colspan="3">Mental model, Users' understanding</td></tr>
<tr><td colspan="3">⇕</td></tr>
<tr><th colspan="3">PHYSICALITIES</th></tr>
<tr><td colspan="3">Enabling Motor Skills (e.g., fingers, hands)</td></tr>
<tr><td colspan="3">Posture, Arity (e.g., single, multiple, mixed)</td></tr>
<tr><td colspan="3">⇕</td></tr>
<tr><th colspan="3">MOVEMENT STRUCTURE</th></tr>
<tr><td colspan="3">A set of Atomic Movements</td></tr>
<tr><td rowspan="2">Elementary (E)</td><td colspan="2">Ref (R)</td></tr>
<tr><td colspan="2">Motion (M)</td></tr>
<tr><td rowspan="2">Compound (C)</td><td>Parallel (P)</td><td>$\text{P} := \text{P}_1 * \text{P}_2;\ P_1, P_2 \in \{E, P\}$</td></tr>
<tr><td>Sequential (S)</td><td>$\text{S} := S_1 - S_2;\ S_1, S_2 \in \{E, C\}$</td></tr>
</table>

and asymmetric interaction. The *Sequential* class refers to users performing a set of atomic movements, either parallel or elementary, holding and releasing hands or fingers, on and from the surface, in a discrete iterative manner.

3.5 Goal-Oriented Spontaneous Gestures

We conducted a second experiment to understand how users explain variability for the gestures they produce. We have two main goals: (1) we are interested in how many variations a user would be able to propose and (2) we are interested in observing how people express variability for the same gesture type and what strategies they employed to create different articulation patterns for the same gesture. We asked 16 new participants to create as many different articulation variations as they were able to for 22 gesture types (see Fig. 3.1), given the requirement that executions were realistic for practical scenarios, i.e., easy to produce and reproduce later.

3.5.1 Gesture Variations

Participants were instructed to propose as many articulation variations as possible for each gesture type. We collected 5,155 (=1031 × 5) total samples for our set of 22 gesture types. In (Fig. 3.2), we summarize the number of gesture variations produced for each gesture type. on average, our participants proposed 2.92 variations per gesture type (SD = 0.45), a result which we found to be in agreement with the findings of Oh and Findlater [21] for action gestures (mean 3.1, SD = 0.8). A Friedman test revealed a significant effect of gesture type on the number of variations ($\chi^2(21) = 84.41$, $p < 0.001$). The “star” and “spiral” gestures presented the lowest number of variations (1.68 and 2.19 variations on average). The gesture with the

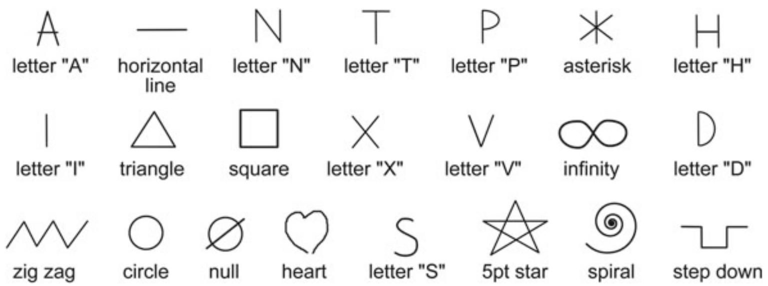


Fig. 3.1 The gesture dataset for the second experiment contains 22 gesture types: letters, geometric shapes (*triangle*, *square*, *horizontal line*, *circle*), and symbols (*five-point star*, *spiral*, *heart*, *zig-zag*), and algebra symbols (*step-down*, *asterisk*, *null*, *infinite*)

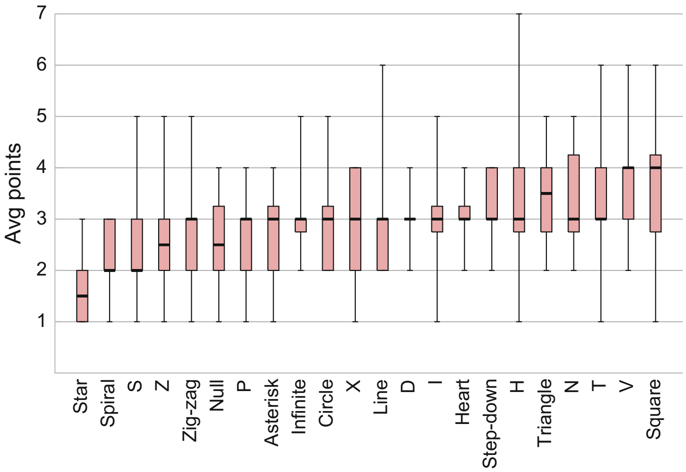


Fig. 3.2 Number of gesture variations produced for each gesture type. *Note boxes* show min, max, median, and first and third quartiles computed with data from all participants

largest maximum number of variations was "square" (3.56 on average) for which our participants managed to easily decompose it into individual strokes that were afterward combined in many ways in time and space. These first results suggest that the specific geometry of the gesture enables users with different affordances of how to articulate that shape. Likely, the mental representation of a gesture variation implies a particular type of articulation which is tightly related to the gesture shape. We can also remark that for all gesture types, except "star" and "spiral" the maximum number of variations was between 4 and 7 variations. This observation suggests that our choice of 4 variations for each gesture type in our first experiment can be even larger for some gestures types and some users. The minimum number of variations was between 1 and 2 variations. This result suggests that for some users and for some gesture types, the number of gesture articulation variations can be limited which can be explained by the previous practice but also by geometrical shape of the gesture.

3.5.2 Strategies for Creating Different Gesture Articulation Variations

To better understand how participants produced different gesture articulation variations for the same gesture type, we report in this section the different strategies elaborated by our participants. Based on our observations and also participants' comments, we arrived at the following strategies:

1. **Vary the number of atomic movements**. As highlighted in our taxonomy, a gesture can be composed of a variable number of atomic movements. To define

a new gesture articulation for the same gesture type, some participants vary the number of atomic movements composing the gesture. Most of them associated the maximum possible number of atomic movements to the number of direction changes in the gesture type (e.g., for "square" gesture, there are four direction changes). Other participants proposed different gesture articulations by varying the way the set of the atomic movement were produced. Two strategies were used: (1) changing the direction of some atomic movements composing the gesture articulation. For instance, an atomic movement representing an horizontal line may be created by moving the fingers from left to right or from right to left; and (2) changing the order of execution of the set of atomic movements composing the gesture articulation. For instance, the same gesture can be articulated using many atomic movements, and for the same atomic movements users may produce different orderings, e.g., there are 442 possible ways to draw a "square" using only sequential movement [31] (p. 273).

2. **Vary the synchronization of atomic movements**. As we showed in our taxonomy, a gesture is composed by a set of atomic movements which can be entered in sequence (i.e., one atomic movement after the other, such as in drawing the "plus" sign with one finger) or in parallel (i.e., multiple atomic movements are articulated at the same time, e.g., using two fingers to draw two sides of a "heart" shape at the same time). To create a new gesture articulation for the same gesture type, participants varied the synchronization of the atomic movements composing their gestures. However, not all gestures can be produced with hand movements in parallel. In fact, only gestures containing a symmetry can be performed with parallel atomic movements. Interestingly, wherever a presented, participants produced synchronous parallel atomic movements to create that part of the gesture (i.e., some atomic movements of the gesture were articulated with one atomic movement at the same time and others were articulated in parallel. e.g., using two fingers at the same time to draw the two diagonal symmetric lines of a "triangle" shape and then one finger to draw the horizontal line).
3. **Vary the number of hands**. As highlighted in our taxonomy, a gesture can be performed by using one hand or both hands. Interestingly, all participants varied the number of hands to articulated gestures. Most participants used one hand only when there was a single atomic movement to produce and used both hands when there were two atomic movements that could be entered in parallel. In addition, when using one-handed gestures, two additional strategies were observed: (1) changing the hand from the dominant to the non-dominant, and (2) alternating hands to enter the sequence of atomics movements. However, that these two strategies were rarely used by our participants.
4. **Vary the number of fingers**. For the same gesture articulation (i.e., the same number of atomic movements and hands with the same synchronization), we rarely observed participants varying the number of fingers to propose a different gesture articulation. This observation confirms that users rarely care about the number of fingers they use to produce multi-touch gestures [4, 13].

3.6 Discussion

In this section, we present user preferences and qualitative data that capture users' mental models as they articulate spontaneous gestures.

3.6.1 Users' Preferences

The primary goal of our user study was to understand users' unconstrained multi-touch gesture articulation behaviors and to analyze the features and degrees of freedom that users will consider to propose different variations for the same gesture type within multi-touch input. This was planned before running our experiment in the form of a questionary that users filled in after completing the task. In fact, we preferred to ask participants about their preferences at the end of the experiment in order to not influence them during the experiment.

After completing the set of gestures, participants were asked to rate their satisfaction regarding their multi-touch performance on a 7-point Likert scale (1 strongly disagree, 7 strongly agree). Results showed that participants were satisfied with the set of gestures they proposed (median 6, stdev = 0.83). Three participants were extremely satisfied and only one participant gave a score of 4 miming that he could propose other gestures by varying the number of fingers.

We then asked participants to rate their preferences regarding the number of fingers, number of strokes synchronization and one hand and bimanual input in gesture articulation; see Fig. 3.3. Interestingly, although bimanual parallel articulations were more represented in the second gesture performed by users rather than in their first gestures, our participants preferred bimanual to one-handed sequential gestures. This observation suggests that people could develop different preferences with practice for articulating gestures in terms of strokes synchronization.

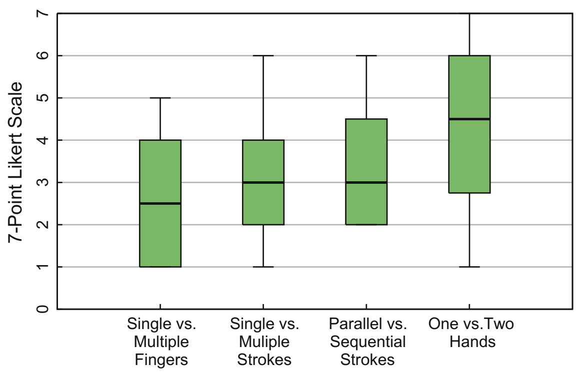


Fig. 3.3 Users' preferences for articulating multi-touch gestures

3.6.2 Users' Mental Models During Spontaneous Gesture Production

Along all our experiments, we observed carefully the variations in how users articulate multi-touch gestures, and we recorded users' qualitative feedback. We highlight in this section such findings.

1. **Preference for multi-finger input**. 13 out of 16 participants used more than one finger per stroke over all. Some participants were enthusiastic to touch the surface with many fingers at once, and witnessed they "*feel more free and comfortable when using many fingers*", while one participant said he was "*more comfortable with multiple fingers, since I feel like their movement is better controlled by my arm*". Although multiple fingers were preferred, participants did not really care about the exact number of fingers touching the surface. One participant witnessed "*one or multiple fingers is the same and has no effect on the stroke nor on gesture expressiveness... I try to see how can I decompose the gesture into multiple strokes and use both hands simultaneously for different strokes*". Also, it was often the case for some fingers to disconnect from the surface for a short period of time during gesture articulation (e.g., start drawing with three fingers, continue with two, finish with three fingers again). For such cases, an appropriate visual feedback might prove useful to show users what unintentionally happened during articulation.
2. **One finger is for precise input**. When participants employed one finger only, they explained that they did so to be more accurate. For example, one participant witnessed that "*when the symbol is complicated, such as a five-point star or spiral, I prefer using one finger to be accurate*". Three participants regularly used one finger to enter gestures. Two witnessed they conceptualized strokes simultaneously articulated by multiple fingers as being different, even though the movement was the same. Participants also made connections between single-finger gestures and pen input in many cases, e.g., "*I use my finger like a pen*". This finding may have implications for future finger gesture designs, as we already know that finger and pen gestures are similar but also different in many aspects [32].
3. **More fingers means more magnitude**. Three participants felt they were drawing thicker strokes when employing more fingers. This finding may have implications on designing interaction techniques that exploit the number of fingers touching the surface beyond finger count menus [9].
4. **Symbol type influences multi-touch input**. Two participants said they articulated letters just like they would write them with the pen, one stroke after another. However, they felt more creative for the other symbols. One participant commented that for the "null" gesture, she would like to draw it just like she have tought at school: first the circle and then the line. Another participant was enthusiastic to touch the surface with both hands at once "*I wish we had been taught to use both hands simultaneously to write letters! It is faster, more precise, and easier*". Most participants considered that the number of strokes and their coordination in time depends on gesture type. One participant said that "*if the symbol*

can be drawn with only one stroke, I prefer to perform it with one stroke only"; two other participants "*whenever there is a symmetry in the symbol, I prefer multiple simultaneous strokes*"; and another participant "*whenever I can decompose the symbol on multiple stokes where I can use my both hands to perform strokes simultaneously, I will do it*".

5. **Gesture position, rotation, size and speed can be a source of variation**. One participant said that the position of the gesture on the surface, gesture size; rotation and velocity represent sources of variation that he could used to propose more gesture articulation variations. However, he did not recur to then for two reasons: (1) varying the number of hands and movement synchronicity over time are more specific and "intuitive" for multi-touch surface, (2) the velocity may be difficult to distinguish without any feedback from the surface.

3.7 Conclusion

The results presented in this chapter contribute toward a better understanding of spontaneous gestures. We now have a more precise idea how users produce unconstrained multi-touch gestures. We also identified how many variations users are able to produce in general, by examining experimental results for a set of representation gesture types. Our findings are important in the context of proposing new interaction techniques that make use of the variability of user gestures. Further work will investigate more aspects of users' multi-touch gesture production behavior in the attempt to reach a systematic understanding of multi-touch interaction with spontaneous gesture production patterns.

Acknowledgements R.-D. Vatavu acknowledges support from the project PN-II-RUTE-2014-4-1187 financed by UEFISCDI, Romania.

References

1. McNeill D (1992) Hand and mind: What gestures reveal about thought. University of Chicago Press, Psychology/cognitive science
2. Kita S (2009) Cross-cultural variation of speech-accompanying gesture: a review. In: Language and cognitive processes, pp 145–167
3. Morris MR, Wobbrock JO, Wilson AD (2010) Understanding users' preferences for surface gestures. Proceedings of GI 2010. ON, Canada, Toronto, pp 261–268
4. Wobbrock JO, Morris MR, Wilson AD (2009) User-defined gestures for surface computing. Proceeding of CHI 2009. ACM, New York, NY, USA, pp 1083–1092
5. Ruiz J, Li Y, Lank E (2011) User-defined motion gestures for mobile interaction. Proceeding of CHI 2011 ACM. NY, USA, New York, pp 197–206
6. Morris MR (2012) Web on the wall: insights from a multimodal interaction elicitation study. Proceedings of ITS 2012. ACM, New York, NY, USA, pp 95–104
7. Vatavu RD (2013) A comparative study of user-defined handheld vs. freehand gestures for home entertainment environments. J Ambient Intell Smart Environ:187–211

8. Vatavu R-D User-defined gestures for free-hand TV control. In: Proceedings of EuroiTV 2012. ACM, New York, NY, USA, pp 45–48
9. Bailly G, Müller J, Lecolinet E (2012) Design and evaluation of finger-count interaction: combining multitouch gestures and menus. Proceedings of IJHCS. Springer, Duluth, MN, USA, pp 673–689
10. Kin K, Agrawala M, DeRose T (2009) Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. Proceedings of GI 2009. ON, Canada, Toronto, pp 119–124
11. Wu M, Balakrishnan R Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In: Proceedings of UIST 2003. ACM, New York, NY, USA, pp 193–202
12. Anthony L, Vatavu R-D, Wobbrock JO (2013) Understanding the consistency of users' pen and finger stroke gesture articulation. Proceedings of GI 2013. ON, Canada, Toronto, pp 87–94
13. Rekik Y, Grisoni L, Roussel N (2013) Towards many gestures to one command: a user study for tabletops. Proceedings of INTERACT. Springer, Berlin, Heidelber, pp 246–263
14. Hennecke F, Berwein F, Butz A (2011) Optical pressure sensing for tangible user interfaces. Proceedings of ITS 2011. ACM, New York, NY, USA, pp 45–48
15. Harrison C, Schwarz, J, Hudson SE TapSense: enhancing finger interaction on touch surfaces. In: Proceedings of UIST 2011. ACM, New York, NY, USA, pp 627–636
16. Lü H, Li Y (2012) Gesture coder: a tool for programming multi-touch gestures by demonstration. Proceedings of CHI 2012. ACM, New York, NY, USA, pp 2875–2884
17. Caramiaux B, Bevilacqua F, Tanaka A (2013) Beyond recognition: using gesture variation for continuous interaction. Proceedings of CHI 2013. ACM, New York, NY, USA, pp 2109–2118
18. Rekik Y, Vatavu R-D, Grisoni L (2014) Match-up & conquer: a two-step technique for recognizing unconstrained bimanual and multi-finger touch input. Proceedings of AVI. ACM, New York, NY, USA, pp 201–208
19. Hinrichs U, Carpendale S (2011) Gestures in the wild: studying multi-touch gesture sequences on interactive tabletop exhibits. Proceedings of CHI 2011. ACM, New York, NY, USA, pp 3023–3032
20. Freeman D, Benko H, Morris MR, Wigdor D (2009) ShadowGuides: visualizations for in-situ learning of multi-touch and whole-hand gestures. Proceedings of ITS 2009. ACM, New York, NY, USA, pp 165–172
21. Oh U, Findlater L (2013) The challenges and potential of end-user gesture customization. Proceedings of CHI 2013. ACM, New York, NY, USA, pp 1129–1138
22. Wu M, Shen C, Ryall K, Forlines C, Balakrishnan R (2006) Gesture registration, relaxation, and reuse for multi-point direct-touch surfaces. Proceedings of TABLETOP 2006. IEEE, Washington, DC, USA, pp 185–192
23. Ringel M, Ryall K, Shen C, Forlines C, Vernier F (2004) Release, relocate, reorient, resize: fluid techniques for document sharing on multi-user interactive tables. Proceedings of CHI EA 2004. ACM, New York, NY, USA, pp 1441–1444
24. Morris MR, Huang A, Paepcke A, Winograd T (2006) Cooperative gestures: multi-user gestural interactions for co-located groupware. Proceedings of CHI 2006. ACM, New York, NY, USA, pp 1201–1210
25. Bau O, Mackay WE (2008) OctoPocus: a dynamic guide for learning gesture-based command sets. Proceedings of UIST 2008. ACM, New York, NY, USA, pp 37–46
26. Kin K, Hartmann B, Agrawala M (2011) Two-handed marking menus for multitouch devices. Proceedings of TOCHI 2011. ACM, New York, NY, USA, pp 1073–0516
27. Kray C, Nesbitt D, Dawson J, Rohs M (2010) User-defined gestures for connecting mobile phones, public displays, and tabletops. Proceedings of mobile HCI 2010. ACM, New York, NY, USA, pp 239–248
28. Rekik Y, Vatavu R-D, Grisoni L (2014) Perceived difficulty understanding users' of multi-touch gesture articulation. In: Proceedings of ICMI. ACM, New York, NY, USA
29. Huang TS, Pavlovic VI (1995) Hand gesture modeling, analysis, and synthesis. In: Workshop on automatic face and gesture recognition. IEEE, pp 73–79
30. Thieffry S (1981) Hand gestures. In: Tubiana R (ed) The hand. University of Chicago Press

31. Vatavu R-D, Anthony L, Wobbrock JO (2012) Gestures as point clouds: a $P recognizer for user interface prototypes. Proceedings of ICMI 2012. ACM, New York, NY, USA, pp 273–280
32. Tu H, Ren X, Zhai S (2012) A comparative evaluation of finger and pen stroke gestures. Proceedings of CHI 2012. ACM, New York, NY, USA, pp 1287–1296

Chapter 4
Remote Proxemics

Maurício Sousa, Daniel Mendes, Daniel Medeiros, Alfredo Ferreira, João Madeiras Pereira and Joaquim Jorge

Abstract Virtual meetings have become increasingly common with modern video-conference and collaborative software. While they allow obvious savings in time and resources, current technologies add unproductive layers of protocol to the flow of communication between participants, rendering the interactions far from seamless. In this work we describe in detail Remote Proxemics, an extension of proxemics aimed at bringing the syntax of co-located proximal interactions to virtual meetings. We also describe the role of Eery Space as a shared virtual locus that results from merging multiple remote areas, where meeting participants' are located side-by-side as if they shared the same physical location. Thus rendering Remote Proxemics possible. Results from user evaluation on the proposed presence awareness techniques suggest that our approach is effective at enhancing mutual awareness between participants and sufficient to initiate proximal exchanges regardless of their geolocation, while promoting smooth interactions between local and remote people alike.

M. Sousa (✉) · D. Mendes · D. Medeiros · A. Ferreira · J.M. Pereira · J. Jorge
INESC-ID/Tcnico Lisboa/University of Lisbon, Lisbon, Portugal
e-mail: antonio.sousa@ist.utl.pt

D. Mendes
e-mail: danielmendes@ist.utl.pt

D. Medeiros
e-mail: daniel.medeiros@tecnico.ulisboa.pt

A. Ferreira
e-mail: alfredo.ferreira@ist.utl.pt

J.M. Pereira
e-mail: jap@inesc-id.pt

J. Jorge
e-mail: jaj@inesc-id.pt

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_4

4.1 Introduction

When people get together to discuss, they communicate in several manners, besides verbally. Hall [14] observed that space and distance between people (proxemics) impact interpersonal communication. While this has been explored to leverage collaborative digital content creation [22], nowadays it is increasingly common for work teams to be geographically separated around the globe. Tight travel budgets and constrained schedules require team members to rely on virtual meetings. These conveniently bring together people from multiple and different locations. Indeed, through appropriate technology, it is possible to see others as well as to hear them, making it easier to communicate verbally and even non-verbally at a distance.

The newest videoconferencing and telepresence solutions support both common desktop environments and the latest mobile technologies, such as smartphones and tablet devices. However, despite considerable technological advances, remote users in such environments often feel neglected due to their limited presence [24]. Moreover, although verbal and visual communication occur naturally in virtual meetings, other modes of engagement, namely proximal interactions, have yet to be explored. This is unfortunate, since proxemics can enable many natural interactions obviating the need for cumbersome technology-induced protocol, which is a plague of remote meetings.

In this work, we deepen the concept of *Eery Space*, introduced in Sousa et al. [30], as a virtual construct to bring remote people together and mediate natural proxemic interactions between everyone as if they were in the same physical place, a mechanism which we call *Remote Proxemics*. To this end, Eery Space allow us to merge different rooms into one virtual shared locus where people can meet, share resources and engage in collaborative tasks. Building on the notion that people do not need hyper-realistic awareness devices, such as virtual avatars, to infer the presence of others and engage in natural social behaviour, Eery Space employs an iconic representation for remote people. Also, to facilitate virtual meetings, we propose novel techniques for person-to-person and person-to-device interactions. We adopt a multiple interactive surfaces environment, which comprises an ecosystem of handheld devices, wall-sized displays and projected floors.

In the remainder of the document, we start by reviewing the related work that motivated our research, describe the Eery Space fundamentals, report the evaluations required to arrived at our proposed awareness techniques and detail our prototype implementation. Also we present both results and findings from our evaluation steps and, finally, we draw the most significant conclusions and point out future research directions.

4.2 Background

Our work builds on related research involving virtual meetings and proxemics applied to ubiquitous computing (ubicomp) environments. In virtual meetings, technology plays a decisive role in providing the necessary means for people to communicate and collaborate while not sharing the same space. Wolff et al. [34] argued that systems that enable virtual meetings can be categorised into *audioconferencing*, *groupware*, *videoconferencing*, *telepresence* and *collaborative mixed reality systems*. Regarding videoconferencing and telepresence, there is research addressing the interpersonal space of the people involved in virtual meetings, by providing a broadcast of a single person to the group. Buxton et al. [8] proposed a system where remote people were represented by individual video and audio terminals, called *HIDRA*. Morikawa et al. [23] followed the concept of the shared space and introduced *HyperMirror*, a system that displays local and remote people on the same screen. Although this approach enables communications, its focus on the interpersonal space renders the user experience not appropriate to jointly create content. People need to meet in a shared space to perform collaborative work [7]. Thereby, Buxton [6] argued that virtual shared workspaces, enabled by technology, are required to establish a proper sense of shared presence, or telepresence.

Using a shared virtual workspace, people can meet and share the same resources, allowing for collaboration and creation of shared content. Following this concept, Tanner et al. [31] proposed a side-by-side approach that exploits multiple screens. One was used for content creation and another to display a side view of the remote user. Side-by-side interactions allow people to communicate and transfer their focus naturally between watching and interacting with others and the collaborative work [31]. In addition, efforts to integrate the interpersonal space with the shared workspace, resulted in improved work flow, enabling seamless integration of live communication with joint collaboration. Ishii et al. [16] introduced *Clearboard*, a videoconferencing electronic board that connects remote rooms to support informal face-to-face communication, while allowing users to draw on a shared virtual surface. Differently, Kunz et al. [20], with *CollaBoard*, employed a life-sized video representation of remote participants on top of the shared workspace.

Shared immersive virtual environments [25] provide a different experience from "talking heads" in that people can explore a panoramic vision of the remote location. In the Office of the future, a vision proposed by Raskar et al. [25], participants in a virtual meeting can collaboratively manipulate 3D objects while seeing each other as if they were in the same place, by projecting video on the walls of each room, thereby virtually joining all remote places into one shared workspace. Following similar principles, Benko et al. [5] introduced *MirageTable*, a system that brings people together as if they were working in the same physical space. By projecting a 3D mesh of the remote user, captured by depth cameras, onto a table curved upwards, a local person can interact with the virtual representation of a remote user to perform collaborative tasks.

Beck et al. [4] presented an immersive telepresence system that allows distributed groups of users to meet in a shared virtual 3D world. Participants are able to meet front-to-front and explore a large 3D model. Following a different metaphor, Cohen et al. [9] described a video-conferencing setup with a shared visual scene to promote co-operative play with children. The authors showed that the mirror metaphor can improve the sense of proximity between users, making it possible for participants to interact with each other using their personal space to mediate interactions similarly to everyday situations.

The theory of Proxemics describes what interpersonal relationships are mediated by distance [14]. Furthermore, people adjust their spatial relationships with other accordingly to the activity they are engaged on, be it simple conversation or collaborative tasks. Greenberg et al. [11] argued that proxemics can help mediate interactions in ubicomp environments. Furthermore, they proposed that natural social behaviour carried out by people can be transposed to ubicomp environments to deal with interactions between people and devices, and even, by devices talking to each other.

When ubicomp systems are able to measure and track interpersonal distances, digital devices can mediate interactions according to the theory of Proxemics. Effectively, Kortuem et al. [19] demonstrated how mobile devices can establish a peer-to-peer connection and support interactions between them by measuring their spatial relationship. Proximity can also be used to exchange information between co-located devices either automatically or by using gestures [15]. This is illustrated by the GroupTogether system, where Marquardt et al. [22] explored the combination of proxemics with devices to support co-located interactions.

Vogel and Balakrishnan [32] developed design principles for interactive public displays to support the transition from implicit to explicit interaction with both public and personal information. By segmenting the space in front of the display, its content can change from public to private for distinct users or the same user at distinct occasions, and different interactions become available. Similarly, Ju et al. [17] applied implicit interactions using proxemic distances to augmented multi-user smartboards, where users in close personal proximity can interact using a stylus, while users at a distance are presented with ambient content. More recently, Marquardt et al. [21] addressed connecting and transferring information between personal and shared digital surfaces using interactions driven by proxemics. In this environment, digital devices are aware of the user's situation and adapt by reacting to different interactions according to context. Ballendat et al. [2] introduced a home media player that exploits the proxemic knowledge of nearby people and digital devices, including their position, identity, movement and orientation, to mediate interactions and trigger actions.

Based on social space considerations, Edward Hall [14] encapsulated everyday interactions in a social model that can inform the design of ubiquitous computing systems to infer people's actions and their desire to engage in communication and collaboration. Indeed, recent research uses proxemics theory, not only to infer the intentions when people want to start interacting with others, but also to mediate interactions with physical digital objects [17, 22, 27]. Despite the advances in both ubicomp and cooperative work, no attempts to extend proximity-aware interactions to remote people in virtual meeting environments, have been made so far. We strongly believe that remote collaborative environments have much to gain by applying proxemics to mediate interactions between remote people. By transposing the way people work in a co-located settings to telepresence environments, the constraints imposed by current technologies for computer supported collaborative work can be alleviated and the sense of presence by remote participants enhanced.

4.3 Joining People Together

Our work focuses on mediating collaborative work in virtual meetings between geographically dispersed people, in order to tackle the issue of the remote people's lack of physical presence. By doing that we can say that proxemic interactions may improve people engagement on virtual meetings. The proposed solution succeeds at bringing geographically distant people together at the same space, and provides the necessary devices and feedback for participants in the virtual meeting to be able to proximally interact with each other at the distance. Therefore, we propose a solution that exploits Remote Proxemics and allows local people to improve the level of awareness of the remote participants' presence.

By bringing proxemic interactions to a remote setting, local and remote participants in virtual meetings can naturally relate to the presence of one another and engage in collaborative tasks as if they were co-located. Providing that they are aware of one another's situation and actions (Awareness). Subsequently, we present presence awareness techniques to render Remote Proxemics possible.

4.3.1 Eery Space

We propose an approach to bring geographically distant people together into a common space. We call the ability to provide feedback on virtual meetings participants' devices for proximal interaction, Eery Space. Given that people are distributed across similar rooms in different locations, it attempts to consolidate these in a common virtual locus, while providing new opportunities for interaction and communication between participants. In this way, people equipped with personal handheld devices can meet, collaborate and share resources regardless of where they are.

Instead of placing users in front of each other, as it is typical of commercial applications and other research works [4, 5], we place both remote and local people side-by-side, similarly to Cohen et al. [9]. Unlike the common interactions with remote people using the mirror metaphor, Eery Space provides remote participants with a sense of being around local ones in a shared space. This creates and reinforces the model of a shared meeting area where proxemic interactions can take place. Moreover, each person has a definite position and a personal space within Eery Space, as

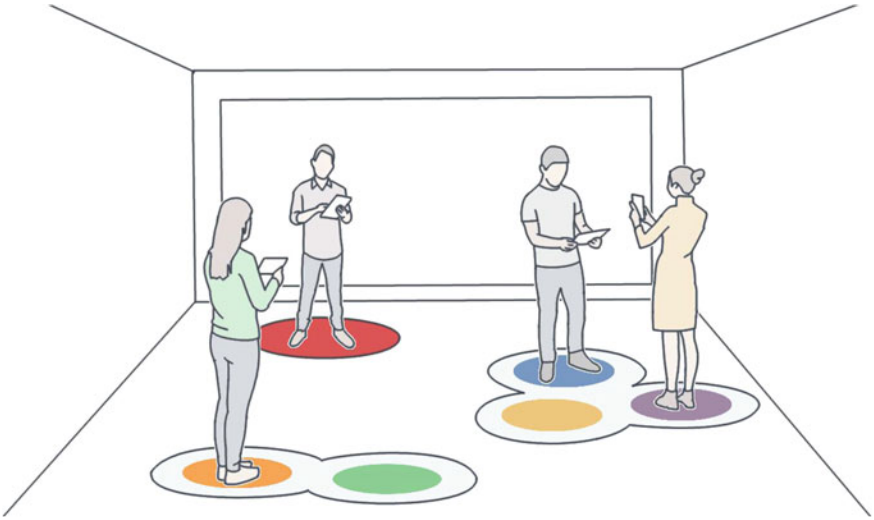


Fig. 4.1 Eery Space

depicted in Fig. 4.1. Allowing both local and remote people to collaborate by relating to their personal spaces strengthens the notion that everyone is treated similarly as if they were all physically local.

Furthermore, Eery Space makes it possible to accommodate differently-sized rooms. Its overall size and shape reflect the dimensions of the meeting rooms in use, as depicted in Fig. 4.2. Eery Space's goal is to create an area that preserves the dimensions and proportions of the people's position and motion, thus avoiding unrealistic scaled movements by the meeting participants. Nevertheless, when a room is substantially smaller than the other, people can be located out of reach. This requires additional techniques to gather their attention in order to collaborate, which is described later.

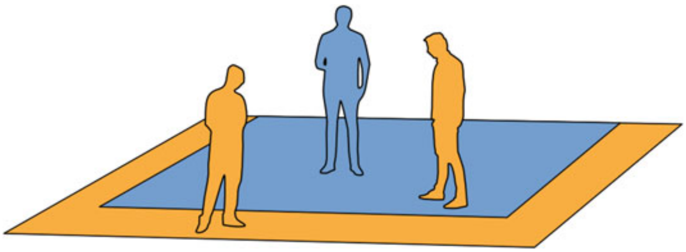


Fig. 4.2 Eery Space, merging two different sized rooms into the same virtual space. The different colours match people to their corresponding physical room

4.3.2 Social Bubbles

Hall's [14] model for proxemic distances dictates that when people are close to each other they can interact in specific ways. Within a proxemic social space, people do interact in a formal way, typical of a professional relationship. In contrast, the personal space is reserved for family and friends and people can communicate quietly and comfortably. Yet, as described by Hall [14], these distances are dynamic. Friendship, social custom and professional acquaintanceship decreases interpersonal distances [28]. We adapted these concepts to Eery Space, using a device we call *Social bubbles*.

Inside Eery Space, interactions are initiated by analysing the distribution of people within the shared virtual space. People having a closed conversation or involved in the same task usually get closer, and, therefore, we create social bubbles resorting to distance. People naturally create a bubble, by coming sufficiently close, and the destruction of bubbles is analogous to its creation—social bubbles cease to exist when participants move apart.

The intention of the people to perform a collaborative task is implicitly detected when they create a social bubble around them. Since we are in a working environment, people do not need to enter the personal space of each other, because they can be neither family nor friends. Instead, a social bubble is created through the intersection of the personal spaces, as depicted in Fig. 4.3. This formulation of social bubbles' model allows people motivated by the collaborative work to easily create proximal interactions using a distance well inside their social space, without needing to violate each others' personal space. In our work, we considered the personal space to be a circle with a 0.6 m radius, thus two people should be at maximum distance of 1.2 m to create a social bubble.

To summarise, we define the concept of social bubbles as the virtual space that results from the intersection of the personal space of two or more people, where people can meet, share resources and engage in closed conversation.

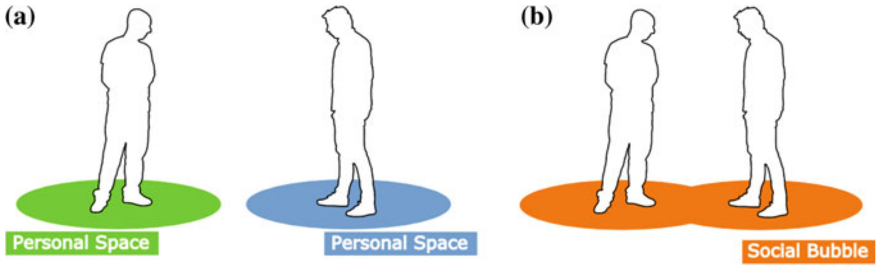


Fig. 4.3 Social Bubbles: **a** While distant from one another, **b** A social bubble is formed when people's personal spaces intersect

4.3.3 Remote Proxemics

Remote Proxemics captures the natural interactions between co-located people and makes them available to all meeting's participants, even when not physically present.

Thus, all interactions, within Eery Space, have the ability to work similarly for local and remote people. The success of our approach is to ensure that the local and remote people are always present and side by side so participants can create social bubbles in the same way, regardless of their physical presence in the same room. These social bubbles can encompass two or more users, either local or remote. When located in the same bubble, users can engage in collaborative activities.

Since Eery Space implements an environment with multiple people and devices, we have grouped these interactions into two groups: person-to-person, for interactions involving people and their own handheld devices; and person-to-device, for interactions between people and shared devices, such as wall displays or tabletops.

4.3.3.1 Person-to-Person Interactions

When people come together and create a social bubble, a set of tools are made available to support collaborative tasks in the form of person-to-person interactions. These interactions not only include the participants but also their personal handheld devices, as depicted in Fig. 4.4a. Since verbal communication is a key element for the success of virtual meetings, participants can talk and hear the other people in their bubble. When people establish a social bubble, their handheld devices automatically open a channel of communication between local and remote participants. This channel of communication is then closed immediately when the bubble is destroyed. Similarly and simultaneously, in case of existing a shared visualization device, such as a wall display, the handheld devices of participants in the same social bubble can be synchronised with the common visualisation. At this stage, participants can engage in a collaborative session around said visualization, either discussing or collaboratively creating content.

4.3.3.2 Person-to-Device Interactions

The Eery Space may feature shared devices, that are well suited for shared visualization and collaborative settings, such as wall displays or tabletops. In our work, we explored more specifically the latter kind, as shown in Fig. 4.4b. Due to their large dimensions, such displays have the characteristic of providing a visualisation surface capable of serving multiple people at the same time. Also, they do not restrict people to a single position. Users of wall displays can freely move alongside the surface to better reach the displayed information, move forward to glimpse more details or move further back to get a general view. This kind of displays serve the purpose of relaying the information under analysis to all meeting's participants. Naturally, these shared devices must be located at the same position across all remote areas that make

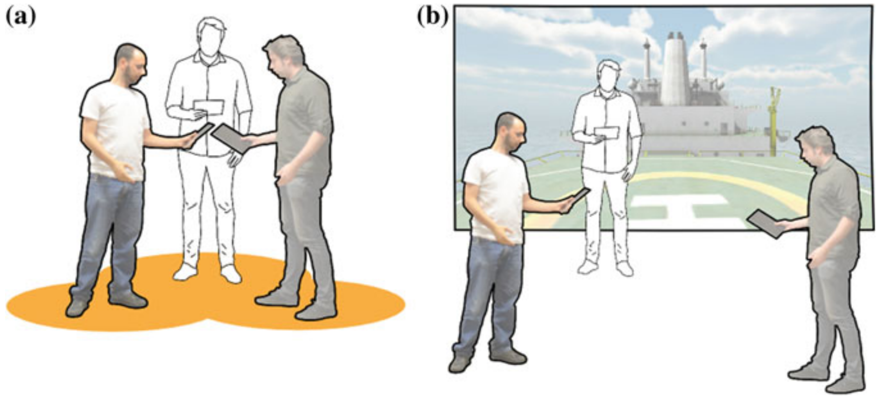


Fig. 4.4 Remote Proxemics: **a** Two local people and one remote (*white*) compose a social bubble and are engaged in collaborative work. **b** The remote participant, closest to the wall display, acts as the moderator, controlling what it is shown in he shared visualization

up the Eery Space, while displaying the same to ensure a consistent source the information to all.

When a participant establishes a proximity relationship with the display, he/she acquires the role of moderator. In Eery Space, the moderator has a special authority that allows him/her to take control of the common visualisation on all shared displays, local or remote, by mirroring actions made on the handheld device. This authority is granted to whom gets closest to the display, inside the moderator space. We define the moderator space as the area contained up to a distance of 1.5 m away from the wall display, analogously to the place normally occupied by a person giving a talk to an audience. Furthermore, the role of moderator can only be handed over when the current person assuming this role abandons the moderator space, leaving it vacant for the next person to take over. Also, when a meeting participant becomes the moderator, a channel for speech communication is opened so that they can address all meeting's attendees. Moderator's speech is relayed through his/her handheld device to participants that are remote in relation to him/her, since local participants already can hear him. The current moderator relinquishes his role when leaving the moderator space. If this happens and another person is standing in that space, then they become the new moderator. Otherwise, the moderator role will be open for anyone to take.

4.4 Tech Overview

Our solution aims to join various rooms in a single virtual space, therefore each component of the environment needs to be aware of its physical location. The prototype is comprised of an ecosystem of multiple running modules communicating to each other using local networks and the Internet and a user tracker to determine the location of all participants. Since Eery Space thrives in merging together multiple physical rooms into one unique virtual space, there must be a similar setup in each physical room. We built our prototype using a multiple Microsoft Kinect-based user tracker, which is able to track six users in a room, dealing with occlusions and resolving each users' position. We used Unity3D to develop a distributed system for multi-peer 3D virtual environment exploration, with support for display-walls, tablets and smartphone clients (iOS and Android). In general, there is a software client for each device in Eery Space and follows the client-server network model for communication and synchronization. Each client has its own version of the data model, a representation of what happens in Eery Space, which is synchronized using messages that pass through the root server. The communication between the various components of the architecture and the server is made by invoking *Remote Procedure Calls* (RPC) over UDP, as depicted in Fig. 4.5. Therefore, our prototype is distributed system responsible for the virtual meeting and manage participants, flow of communication and individual devices present in this environment.

For the purpose of this work, we developed a Graphical User Interface (GUI) approach to bind people with their handheld devices. This approach consists in displaying a map with the position of currently tracked people in the selected room, highlighting ones without association. Thereby, when entering the Eery Space, participants are required to select their representative icon on the handheld device screen before initiating any interactions. Also, we consider the position of an handheld device the same as the person holding it.

4.5 Providing Presence Awareness

While becoming and staying aware of others is something that we take for granted in everyday life, maintaining this awareness has proven to be difficult in real-time distributed systems [13]. Previous research indicate that people can respond socially and naturally to media elements [26]. Thus, we allow remote users to interact through appropriate virtual proxies, by making both the shared space and actions mutually visible. When trying to keep people conscious of other people's presence, an important design issue is how to provide such information in a non-obtrusive, yet effective manner.

4.5.1 First Iteration: Handheld Device-Centered Approach

The initial awareness techniques studied exploited the fact that all participants must hold an handheld device to participate in the meeting to perform collaborative tasks, while remaining in audio contact with remote participants. Following the collaborative guidelines proposed by Erickson and Kellog [10], we used the techniques

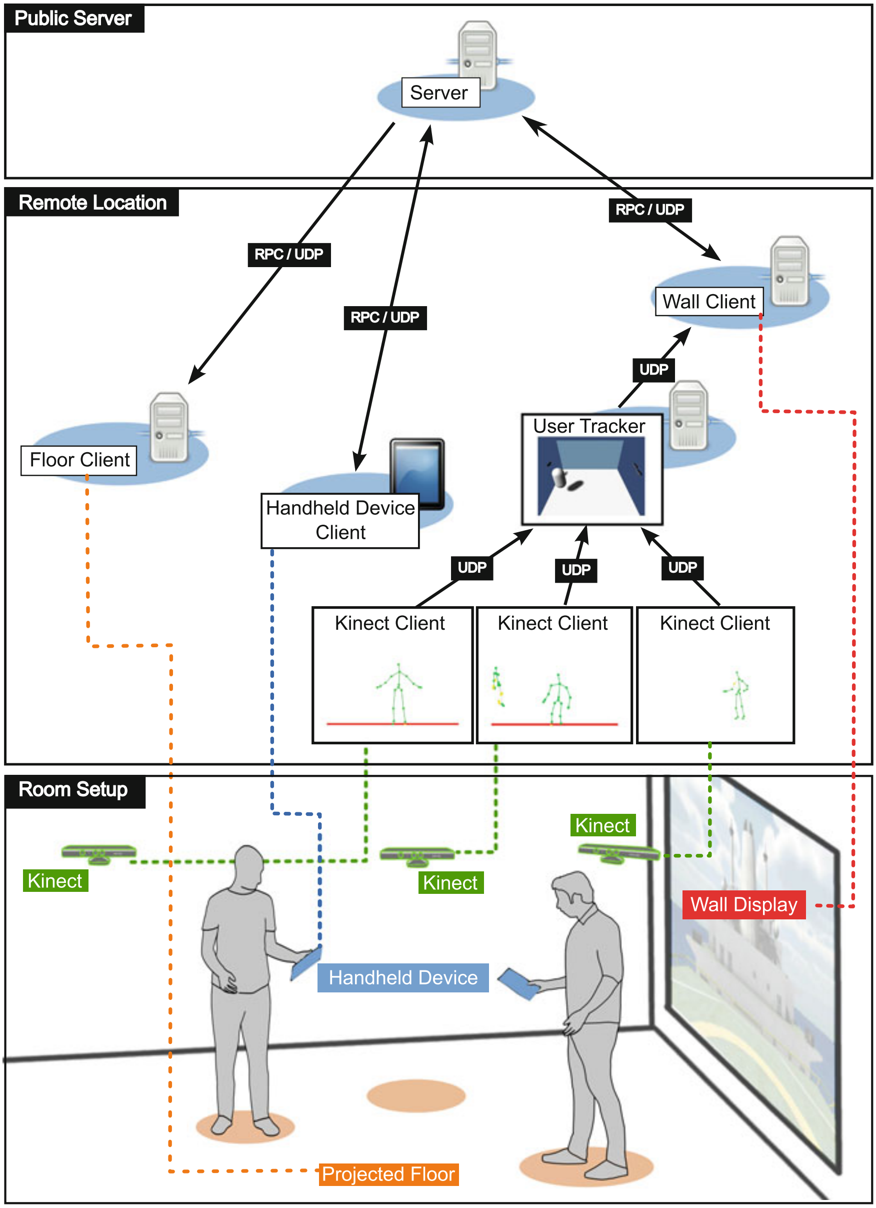


Fig. 4.5 Overall architecture

described below to increase visibility and awareness of other users, namely for remote participants, spanning through two separate iterative phases.

Bubble Map
Whenever someone tilts their handheld device to an horizontal position, a partial top view of the Eery Space is displayed, as depicted in Fig. 4.6a. In the center of the screen, its owner is represented by a large white circle. Other users who are close enough to lie in the same Interactive Bubble as the device owner's are also portrayed as large circles, painted with the colour of each user. Users outside the bubble are considered off-screen. Resorting to an approach similar to Gustafson et al. [12], we place these circles (smaller than users in the same bubble) on the screen edge, indicating their direction according to the device owner's position.

Virtual Windows
Virtual Windows provides a more direct representation of other users' position and orientation. These depict a view into the virtual world, in a similar manner to the work of Basu et al. [3]. Using the combined information of participants' position and the orientation of their handheld device, we calculate the user's own perspective, allowing them to point the device wherever they desire. The virtual window shows both local and remote users (Fig. 4.6b), represented by avatars within the virtual environment. For the purpose of this dissertation, we used a 3D model of a generic clothed human.

Wall Shadows
Every person has a representative virtual shadow on the wall display, distinguished by a name and a unique colour, as shown in Fig. 4.6c similarly to the work of Apperley et al. [1]. The location of the shadow reflects distance from the person to the wall to give a sense of the spacial relationship between the people and the interactive surface. Wall shadows takes in consideration an imaginary directional light source at the infinity and towards the wall display. Thus, the nearest person to the wall display, will have the shadow with more coverage area than the others. A much larger shadow also makes it clear who is the moderator. Furthermore, each user has a coloured aura around their shadow. When two or more people share the same aura colour, this means they are in the same social bubble and can initiate collaborative tasks.

4.5.1.1 Initial Evaluation

This evaluation session aimed on testing the proximity-aware interactions within the Eery Space, both with a local and with a remote person. It also served to evaluate the role of the moderator by grabbing control of the wall display using the person-to-surface interaction method. For this experiment, individual test users were placed in with one local and one remote person, scattered through the two separate rooms. With this, the user was encouraged to move freely in the space in front of the wall display and look for others to establish social bubbles in order to be able to do the collaborative tasks required.

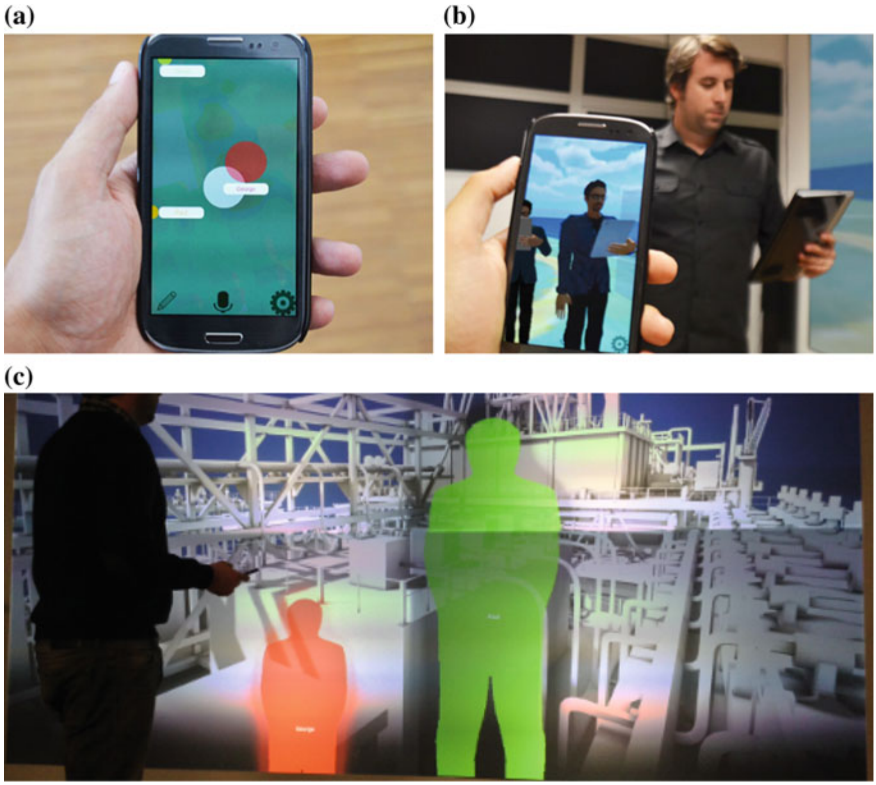


Fig. 4.6 Awareness techniques: **a** Bubble map; **b** Virtual window; **c** Wall shadows

Methodology

In this section, we describe the methodology performed to evaluate the feasibility of proximity-aware interactions in a remote virtual meeting setting. The expected duration for each session with users was about 30 min and was divided into three stages (Table 4.1):

1. **Introduction**
   First of all, an explanation of the objectives and the context of the virtual meeting, were given to the test subject, followed by a demonstration of the prototype's features with a description of all available awareness techniques.
2. **Evaluation Tasks**
   The local participant, acting as the evaluation session moderator, gave out commands to the test user, while engaging in conversation. Approximately, 5 min into this stage, the remote participant was signaled to enter the Eery Space environment.

Table 4.1 Remote Proxemics' preliminary evaluation stages

#	Stage	Time (min)
1	Introduction	10
2	Evaluation tasks	15
3	Filling in the questionnaire	5

3. **Filling in the questionnaire**
   At the end, the user was then asked to fill out the questionnaire. The questionnaire was comprised of a user's profile section and another section with nine questions in a *Likert* scale form of 6 values.

Performed Tasks
Despite having been done informally, the set of evaluation tasks followed a strict order. Next, we present a description of the tasks performed during this stage.

1. **Associate user with mobile device**
   At first, participants were presented with a detailed description in how to become associated with the handheld device provided. This explanation was accompanied with a description of the user tracker and the depth cameras in the setup environment. Then, the they were asked to log into the system.
2. **Change role to moderator**
   This task started with a demonstration of the navigational technique on the handheld device and a demonstration of grabbing the control of the wall's common visualisation. While exploring the virtual environment on the handheld device, participants were encouraged to assume the role of moderator and share the point of view of the his device's virtual camera.
3. **Collaborative task with local participant**
   At this stage, the participants were asked to establish a social bubble with the local participant and start a collaborative sketch annotation. During this task, participants were made aware of the visual changes provided by the awareness techniques.
4. **Collaborative task with remote participant**
   Finally, participants were encouraged to find the remote participant and start a collaborative sketch annotation.

Participants
This initial evaluation session was attended by eleven users, two of them females. With ages ranging from 25 to 60 years, the large majority below 35 years. Also, all had at least a bachelor's degree and revealed different backgrounds, mainly in Engineering and Architecture.

Results and Discussion
The main objective of this initial evaluation was to study the feasibility of remote proxemics by exposing people to the concept. Therefore, the evaluation tasks were

Table 4.2 Questionnaire's results of the initial evaluation (median and interquartile range)

Question	Median (IQR)
1. It was easy to see who is present at the meeting	5 (1)
2. It was easy to see where each participant is	5 (0.5)
3. It was easy to see how I become moderator	6 (0)
4. It was easy to see who is the current moderator	6 (0.5)
5. It was easy to see who is in my social bubble	6 (0.5)
6. It was easy to join the social bubble of another local participant	6 (0.5)
7. It was easy to join the social bubble of another remote participant	5 (1)
8. The task of making a sketch with a local participant was simple to perform	6 (1)
9. The task of making a sketch with a remote participant was simple to perform	5 (1)

designed to expose the test users to local and remote proximal interactions, in order to correlate these two user experiences.

In the Table 4.2, are listed the answers from questionnaire in the form of median and the *interquartile range*, the measure of statistical dispersion of the data values. Since the beginning, the main driver of this work was to present a solution to mediate interactions between local and remote people seamlessly. Since the values obtained from the tasks are two related samples and come from the same population in an ordinal scale, we applied the *Wilcoxon Signed Ranks test* to highlight possible statistically significant differences between local and remote interactions. Accordingly to the *Wilcoxon Signed Ranks test* there are no significant differences between the making a sketch with local and remote participant. Nevertheless, establishing a social bubble, using the same test, shows a statistically significant difference between local and remote ($Z = -2.000$, $p = 0.046$), evincing a degree of difficulty while engaging in remote collaborative tasks, which brings us to the conclusion that the awareness techniques employed were insufficient and not adequate. In fact, from observation, test users were reluctant to utilise the virtual window and the bubble map awareness techniques and restricted their attention to the more expressive information from the wall shadows.

4.5.2 Second Iteration: Augmented Surfaces Approach

Since the results of the first evaluation show that the virtual window and the bubble map were inefficient in grabbing the attention of the participants, we choose to remove those awareness techniques and tryout an augmented floor approach. By introducing an iconic representation all participants on the floor, while preserving the participants intimate space in the Eery Space.

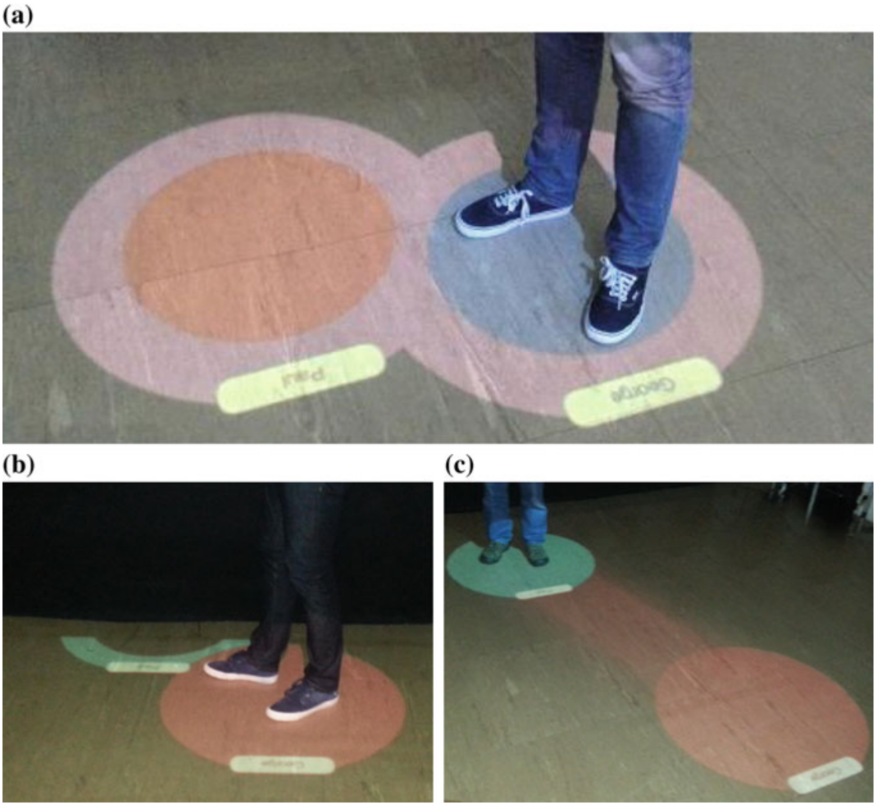


Fig. 4.7 Test user interacting with the local participant during the an evaluation session

Floor Circles

In Eery Space, every local and remote participant has a representative projected circle on the room's floor, as depicted in Fig. 4.7. All floor circles are unique to its corresponding person and are distinguished from each other by a name (the participant identity) and the user's unique colour, analogous to the wall shadows. In addition, these circles move in accordance with the person's position within the Eery Space to visually define the participant's personal space and makes people aware of each other. The floor circles provides the necessary spacial information for participants to initiate proximity interactions. When people come together to start a social bubble, the circles on the floor inform on the status of their bubble, by displaying a coloured aura around the bubble's members. The projected aura gets its colour by the computation of the social bubble's members colour difference. This makes that new colour unique and unmistakably different from the other shadows on the floor. Also, the projected circles depicts the user's proxemic zones. The inner circle, with a radius of 0.3 m, matches the participant's intimate space and the outer ring, the personal space, which in turn occupies the space until 0.6 m.

Since Eery Space merges physical rooms of different dimensions into one shared virtual space, it may occur for some participants to be out of reach of others positioned in smaller rooms. To address this, we implemented a technique to gather the attention of participants out of reach. These participants appear on the edges of the smaller room's floor with only a portion of their circles showing their direction in the virtual space. To differentiate these participants, their intimate space is left blank (Fig. 4.7b). When a participant tries to interact with another that is out of reach, a glowing path appears in the floor of the latter's room (Fig. 4.7c), indicating that someone is trying to interact with him/her, but is unable to. He/she can then approach the circle of said person and normally initiate a proximal interaction.

Intimate Space

We designed Eery Space keeping each person's personal locus in mind. Every user has their own space assured, even if they are not in the same physical room as the others. To prevent users from invading another user's intimate space, we provide haptic feedback by vibrating their handheld device, when this happens. In this way, participants can quietly adjust their positions without interrupting the main meeting, since this technique does not use audio or visual cues.

4.5.3 Solution Evaluation

Since with the preliminary evaluation we can conclude that it is possible to interact with remote people using proxemics, this test session focused on the study of the overall developed solution. In general, this phase of evaluation is comprised of a proxemics interactions and awareness overview, as well as, a comparison of the awareness techniques employed in the final solution. Once the conclusion of the first preliminary study, the prototype had been altered. Accordingly to the noted low usage of the virtual window and bubble map awareness techniques, these features were discarded at this stage. Nevertheless, the floor shadows technique was developed and added to the final prototype, providing the precise information of the location of each remote participant. Similarly to the preliminary evaluation, a single user, for each session, was invited into the main setup environment and asked to interact with a local Fig. 4.8 and another remote participant. Every test user had no previous experience with work in question.

4.5.3.1 Methodology

The solution evaluation phase maintained a similar methodology to the previous phase to guarantee the consistency of the data obtained. Table 4.3 demonstrates the three stages of each session with users. In total, each session lasted approximately 35 min. Below, we describe each stage of this session:

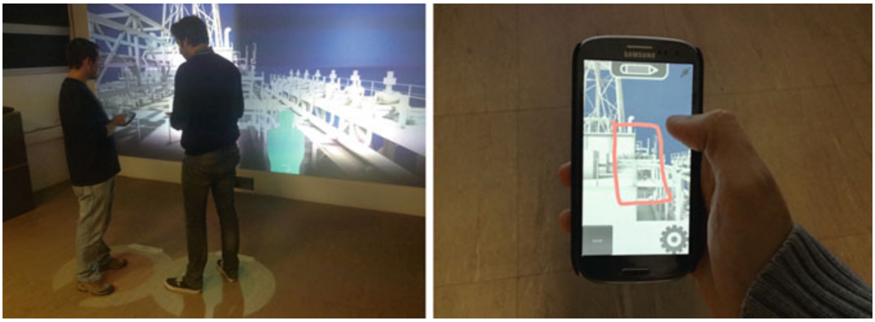


Fig. 4.8 Test user interacting with the local participant during the an evaluation session

1. **Introduction**
   At the start, the new user was greeted with an explanation of the objective for this evaluation session, and with general consideration regarding the prototype. Firstly, was a description of the motivating aspects of our project, the design and review of 3D CAD models in virtual meetings and social interactions with remote people. The user was made aware that they would be interacting with another remote person. Secondly, the user was introduced in the basic features of the prototype, how to be associated with the handheld device and log on into the system. Thirdly, a brief description was made about the nature of the Eery Space and the awareness techniques. Fourthly, the concept of proxemics were discussed, accompanied with a demonstration on how to become the moderator and how to perform a collaborative task by forming a social bubble, emphasising the concept of intimate and personal space. This was followed with a demonstration of the haptic feedback by stepping on the user's intimate space. Finally, the user was encouraged to explore these concepts for a few minutes.
2. **Evaluation Tasks**
   The user was accompanied by both the local and the remote experienced participant. Thus, the user was receiving verbal commands for the tasks it should perform. Since this evaluation requires an honest reaction from the test users, any posed questions were responded with another formulation of the commands, avoiding influences on the user's behaviour.

Table 4.3 Solution's evaluation stages

#	Stage	Time (min)
1	Introduction	10
2	Evaluation tasks	20
3	Filling in the questionnaire	5

3. **Filling in the questionnaire**
   Upon completion of the tasks, the user was asked to fill out a questionnaire, not only to define the profile of the user, but also to gain an appreciation of the various components of our solution.

4.5.3.2 Performed Tasks

The set of evaluation tasks was designed with the intention to check if there is any significant difference between local and remote interactions. Therefore, users were asked to perform a collaborative task, firstly with the local participant, following with the remote participant. Also, to verify if people do react to the presence of other remote people, even only being aware of the representation of their presence. Below, we describe the set of tasks performed by the test users (Fig. 4.9):

1. **Interaction with Wall Screen Display**
   Since navigation in the virtual environment is beyond the context of this evaluation, which focuses on awareness and proxemic interactions, a button was placed on the prototype that automatically redirects the virtual camera to a specific point of interest in the model. This point of interest, common to all users, corresponds to an engineering detail in the virtual environment, and highlighted in red. Thus, the test user were encouraged to press the button and then displaying it on the wall display, by willingly assuming the role of moderator.
2. **Interaction with the local participant**
   To perform this task, user was asked to jointly create a collaborative sketch. For this, he had to physically move to establish a social bubble with the local participant and wait for instructions. Then, the local participant promptly drew a square around the point of interest in the virtual environment, and instructed the test user to draw a circle inside.
3. **Interaction with the remote participant**
   This task is essentially the same as that described with the local participant. The particularity of this task is that the test user was asked to create a collaborative task with the remote participant. To this end, the user had to move to establish a social bubble with the remote participant and wait for instructions. Similarly to

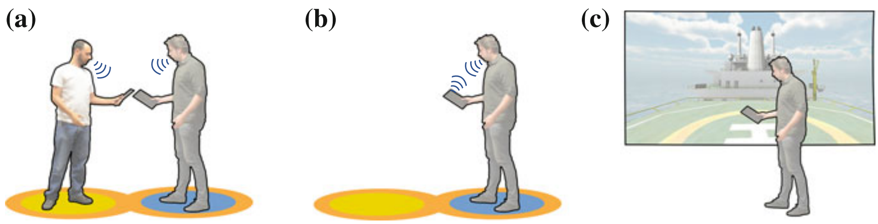


Fig. 4.9 Evaluation tasks

the interaction with the local user and with the remote proxemics-enabled communication in the handheld device, the remote participant, then, instructs the test user to draw a cross inside his circle. At the same time, the remote participant intentionally steps in the test user's intimate space so as to arouse a reaction. Ideally, an adjustment in the position by the test user, demonstrating that he acknowledged the importance of preserving remote people's space in the meeting and also realising that he was interacting with another person.

4. **Intimate space invasion**
   For this task, the test user was instructed to watch the action performed by the remote user as moderator. The remote user moved into position and started to control the navigation on the wall display. At this stage, a computer generated remote user starts pursuing the test user in attempt to invade his intimate space. Again, this task is intended to observe the reaction of the user against an intrusion of their intimate space.
5. **Pathway between remote participants**
   This task was designed to realise if the test user understood the concepts exposed in this evaluation. The user was instructed to move to a target location while considering the presence of four computer generated remote participants. At no point in this evaluation, the user was informed that the computer generated remote users were in fact artificial.
6. **Stress test with multiple participants**
   The final task served as thanks to the test users for their participation in the evaluation session. This task was designed purely as a game and was not considered in the final analysis of the results. Therefore, the final task consisted of six computer generated remote users pursuing the test user in the attempts to invade his intimate space.

4.5.3.3 Participants

The participants in this trial were invited randomly and were mainly students attending our educational institution. Thereby, the set of test users was comprised of 12 participants, one of which was female, and all with a college degree. In regard to their age, the majority of the test users were between 18 years old and 24, remaining one of them between 35 and 55 years old. Nearly all reported having experience using smartphones in a daily base, except one of them that did not own a smartphone.

4.5.3.4 Results and Discussion

In this section, we present an analysis of the data obtained from the evaluation of the overall solution. The data gathered of the user's preferences were obtained from the *Likert* scale of 6 values, presented previously. Also, data from the observation of the performed tasks is considered in this analysis.

The main objective of this evaluation was to demonstrate the feasibility of remote proxemics by maintaining an adequate level of awareness of the people that are remote, since the main premise of this work was that remote proxemics is possible provided that local participants have the awareness of the location and status of the remote ones. Furthermore, this evaluation provides a study of each awareness technique present on the final solution. Therefore, the analysis of the results is divided into a *Proxemics Overview*, an *Awareness Overview* and a comparison between the awareness techniques employed. A discussion of the final results is also provided along this section.

Proxemics Overview
The user's preferences regarding proxemics interactions are related to their easiness to perform proximal interactions with both local and remote people, and also the ability to interact with the wall display. The latter, poses a conscious decision to become the moderator of the virtual meeting. In Table 4.4 are depicted the responses obtained from the questionnaire regarding those interactions, in the form of median and interquartile range. The presented data suggests that it was easy to assume the role of moderator. According to the *Wilcoxon Signed Ranks* test between the first and second questions ($Z = -1.890$, $p = 0.059$), there are no statistically significant differences between starting a interaction with the other participants, despite their local or remote statuses. What leads us to conclude that interacting with remote people is no different than local interactions. This result is encouraging insofar as it can prove that remote proxemics are in fact possible and do not add obstacles in the course of virtual meetings. In the trials, users did not demonstrate any difficulty in repositioning themselves to establish social bubbles in the collaborative tasks, although three users took up to five seconds to remember how to become the moderator while in the first task.

Awareness Overview
Regarding awareness, Table 4.5 summarises the data obtained from the test trials. In general, the presented data shows that people in the virtual meeting can relate to the presence of remote participants. User's preferences suggests that, despite exhibiting a slight dispersed data (Table 4.5, question 2), the absolute location of remote people is always visible. We can safely deduce that participants in the virtual meeting are always aware of the people involved. One of the requirements of our approach is the preservation of the intimate space of remote people. This design principle is required to impose their presence, while fostering remote interactions by establish-

Table 4.4 Questionnaire's results of the proxemics overview (median and interquartile range)

Question	Median (IQR)
1. It was easy to control what is shown on the wall display	6 (1.25)
2. It was easy to start an interaction with a local participant	6 (0)
3. It was easy to start an interaction with a remote participant	6 (1)

Table 4.5 Questionnaire's results on the awareness overview (median and interquartile range)

Question	Median (IQR)
1. It was easy to see who is present at the meeting	6 (0)
2. It was easy to see where each participant is	6 (1.25)
3. It was easy to see who is controlling the wall display	6 (0.25)
4. It was easy to see that I'm interacting with other people	6 (0.25)
5. It was easy to see which participant I'm interaction with	6 (1)
6. It was easy to see that I'm in the intimate space of another local participant	6 (0)
7. It was easy to see that I'm in the intimate space of another remote participant	6 (0)

ing social bubbles. The *Wilcoxon Signed Ranks test* applied to the questions 6 and 7 shows no statistically significant difference between local and remote people, suggesting that test users were aware when their intimate space intercepted the others. Curiously, while performing the collaborative task, three test users made a point of informing the remote participant of his infringement on their personal space during the smartphone-enabled conversation before readjusting their position. While during the intimate space invasion task every one of them changed their positions, responding to the haptic feedback from the handheld device. Despite that, four users complained that the remote participant was invading their intimate space, and then proceeded to readjust their positions. In the final task, the pathway between remote participants, only one user did not take into account the presence of the remote participants and walked right through them, suggesting that almost every one accepted the presence of remote people and walked accordingly by dodging the floor shadows while walking to the destination. It is then safe to say, that in general, were aware of the presence of the remote participant and reacted accordingly. Nevertheless, one of the test subjects expressed the need to be aware of the others orientation in the meeting.

Awareness Techniques Comparison

To provide a model for future remote proxemic interactions, we compared the awareness techniques employed in the final solution to understand their decisive role in providing awareness of the presence of remote people. Not just presence in general, but also location and status. In out final approach, the awareness techniques encompass multiple devices and surfaces. Wall shadows provide the whereabouts of all users, in the meeting, on the wall screen display. The projected floor surface displays representative shadows of the absolute location of each person in the Eery Space. And handheld devices provide haptic feedback when the participants intersect their intimate spaces. In the Table 4.6 we show the results from the questionnaire regarding the awareness techniques. First of all, by design, the handheld device client do not provide awareness of the meeting's participants or their location. Also, the handheld device is not able to show who is the moderator. For this, we applied the *Wilcoxon Signed Tanks test* in every question between all awareness techniques. There was no

Table 4.6 Questionnaire's results on the awareness techniques comparison (median and interquartile range for techniques available on the wall display, floor and handheld device)

	Median (IQR)		
Question	Wall	Floor	Handheld
1. Helped to realise who is present at the meeting	5 (1)	6 (1.25)	–
2. Helped to realise where each participant is[a]	5 (0.25)	6 (1)	–
3. Helped to realise who is controlling the wall display[a]	6 (0)	4 (3)	–
4. Helped to realise that I'm interacting with other participants[a]	4 (1.25)	6 (0.25)	6 (1.25)
5. Helped to realize with whom I'm interacting with[a]	4 (2.25)	6 (0.25)	3.5 (1.25)
6. Helped to realize that I'm in the intimate space of a remote participant[a]	2.5 (1.25)	6 (1)	6 (0.25)

[a]There are statistically significant differences

statistically significant difference in providing the information of who was present in the meeting, between the floor and the wall shadows. This was the only question that the *Wilcoxon Signed Tanks test* did not demonstrated a statistically significant difference. Although the floor shadows has scored higher (median) in the questionnaire, we believe that this is due to the remote participant entering the Eery Space from the side and not from the rear, which could invert these results. Regarding the location of each participant, the floor shadows proved to be better at this task ($Z = -2.456$, $p = 0.014$). In return, the wall shadows proved to be more efficient in showing who is controlling the wall display ($Z = -2.966$, $p = 0.003$). In fact the representation of the moderator changes on the wall display, while the floor shadows only shows him at close proximity. Wall shadows proved insufficient to provide awareness of the formation of social bubbles while interacting with other people, against the floor shadows ($Z = -2.595$, $p = 0.009$) and the handheld device ($Z = -2.122$, $p = 0.034$). We believe that the auras on the wall shadows are not that expressive and the information on the floor and on the handheld device is more prone to hold onto the participants attention. In a similar note, the floor shadows ($Z = -2.956$, $p = 0.003$) and the handheld device ($z = -2.958$, $p = 0.003$) both proved to be effective in providing the awareness while stepping in the intimate space of another participant. And finally, according to the *Wilcoxon Signed Tanks test*, the projected floor depicts a better representation of the people involved in a collaborative task, against the wall shadows ($Z = -2.461$, $p = 0.014$) and the handheld device ($Z = -2.958$, $p = 0.003$). Thereby, it is safe to conclude that the wall screen display is necessary for the tasks of design and review of 3D models, but proved to be somewhat irrelevant to the functioning of the social interactions in the Eery Space.

The final results clearly show an improvement in the awareness of the remote people's presence, in the way that local and remote interactions are virtually the same. We would not have gotten this result if we had not done the preliminary evaluation. The first evaluation trials, suggested the need for a more expressive awareness technique to depict the exact location and status of remote participants. With that, the projected floor has filled this requirement and presents a favourable acceptance by the test users.

4.6 Challenges and Opportunities

In Eery Space [29] people can see each others representation and quickly grasp their location and status on the virtual meeting through a large scale display, the projected floor and personal handheld devices. As a matter of fact, these awareness techniques render Remote Proxemics possible since they effectively highlight the presence of remote people. The current state of the art regarding remote proxemics employs the usage of fixed proxemic dimensions [14] to engage people in collaborative tasks. Also, Eery Space counts on fixed distances to start and terminate interactions, which, despite being near to normal social interactions are still somewhat far from the way people locally interact, meaning that transitions between interactions are still abrupt. We believe that remote proxemic interactions can be improved by allowing a more fluid model like gradual engagement [21] and f-formations [18, 22].

In general terms, Eery Space can be enriched with new workflow dynamics. Gradual engagement can improve interactions in various tasks including data visualization, communication and content creation, instead of the previous way of interaction. Figure 4.10 depicts a future model for remote interactions, by dividing the distance between two people in proxemic distances. Another important aspect is the orientation of one person among the others which also is an crucial factor to define the Eery Space's interaction stages. Following the previous generalization of the interaction model, we can describe the an enrich Eery Space interaction stages:

Ambient Interaction
In this interaction stage, people present in the Eery Space can easily take a glance to the state of the meeting by looking to the projected floor. They can only get an overall glimpse of who is present and what are they doing. Despite that, people physically closer, or in any of the other stages, if their orientation do not match a close encounter [22], the system deal with them as if they were in the Ambient stage.

Peripheral Interaction
People passing by this stage are able to access peripheral information the collaborative tasks performed by the others.

Engaged Interaction
People inside this stage implicitly makes notice what is being discussed and can communicate with the contents owners.

Personal Interaction
On this stage users can interact directly with each other and edit content, if the content owner gives access to the other users.

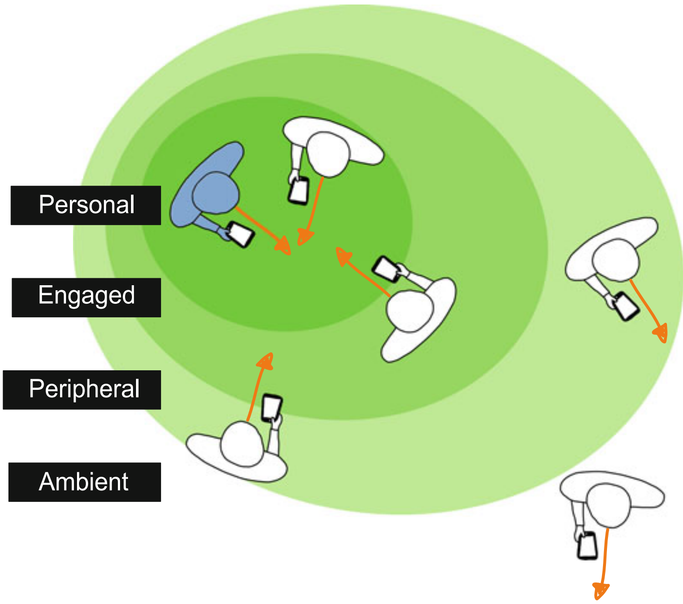


Fig. 4.10 Gradual engagement model for remote proxemics

Since people, in real life, rely on close encounters and orientation to communicate. The same approach can be used to establish group interactions and broadcast speech to other participants in remote proxemic environments.

4.7 Conclusions and Future Work

Virtual meetings play an important role in bringing geographically separated people together and are broadly used in business and engineering settings where experts around the world engage in collaborative tasks. While current videoconference and telepresence solutions do enable verbal and visual communication between all participants, other forms of non verbal communication, namely social interactions through proxemics, have not been explored to their full potential.

We have introduced Remote Proxemics, which brings social interactions to virtual meetings, and explores interactions between local and remote people as if they were in the same space. To this end, we created Eery Space, a shared virtual locus where people can meet and interact with each other using Remote Proxemics. We developed a prototype to explore both people-to-people and people-to-device interactions and study techniques for providing the appropriate awareness. Indeed, these awareness techniques render Remote Proxemics possible, since they highlight the presence of remote people. Using a projected floor and personal handheld devices, people can see others' representations and quickly realise their location and status on the virtual meeting. Results from our evaluation show the promise of Remote Proxemics, since we were able to achieve seamless interactions between local and remote people. We believe that the work here described extends proxemic interactions to augment the presence of remote users in virtual collaborative settings to address commonly-raised concerns. Furthermore, our results apply even in the absence of commonly explored devices such as avatars and eye contact.

We consider it is both possible and interesting to apply our innovative approach to additional purposes and scenarios, ranging from engineering to architectural projects. To bind people with their personal handheld devices in a more flexible manner, we intend to explore automatic approaches, for example using computer vision, as suggested by Wilson et al. [33]. Also, we consider that it would be interesting to assess whether adding support for f-formations [22] will also enrich remote interactions in the Eery Space.

Acknowledgements This work was partially supported by the Portuguese Foundation for Science and Technology (FCT) through the projects TECTON-3D (PTDC/EEI-SII/3154/2012), CEDAR (PTDC/EIA-EIA/116070/2009), and by national funds through FCT with reference UID/CEC/50021/2013. Daniel Medeiros would like to thank CAPES Foundation, Ministry of Education of Brazil for the scholarship grant (reference 9040/13-7).

References

1. Apperley M, McLeod L, Masoodian M, Paine L, Phillips M, Rogers B, Thomson K (2003) Use of video shadow for small group interaction awareness on a large interactive display surface. In: Proceedings of AUIC '03
2. Ballendat T, Marquardt N, Greenberg S (2010) Proxemic interaction: designing for a proximity and orientation-aware environment. In: ACM ITS. ACM, pp 121–130
3. Basu A, Raij A, Johnsen K (2012) Ubiquitous collaborative activity virtual environments. In: Proceedings of CSCW'12
4. Beck S, Kunert A, Kulik A, Froehlich B (2013) Immersive group-to-group telepresence. IEEE Trans Vis Comput Graph
5. Benko H, Jota R, Wilson A (2012) Miragetable: freehand interaction on a projected augmented reality tabletop. In: Proceedings of CHI '12
6. Buxton W (1992) Telepresence: Integrating shared task and person spaces. Proc Graph Interface 92:123–129
7. Buxton W (1997) Living in augmented reality: ubiquitous media and reactive environments. Video Mediated Communication. pp 363–384
8. Buxton WA, Sellen AJ, Sheasby MC (1997) Interfaces for multiparty videoconferences. Video-mediated Communication, pp 385–400
9. Cohen M, Dillman KR, MacLeod H, Hunter S, Tang A (2014) Onespace: shared visual scenes for active freeplay. In: Proceedings of CHI '14
10. Erickson T, Kellogg WA (2000) Social translucence: an approach to designing systems that support social processes. ACM TOCHI
11. Greenberg S, Marquardt N, Ballendat T, Diaz-Marino R, Wang M (2011) Proxemic interactions: The new ubicomp? Interactions 18(1):42–50
12. Gustafson S, Baudisch P, Gutwin C, Irani P (2008) Wedge: clutter-free visualization of off-screen locations. In: Proceedings of CHI '08

13. Gutwin C, Greenberg S (2002) A descriptive framework of workspace awareness for real-time groupware. CSCW
14. Hall ET (1966) The hidden dimension. Doubleday
15. Hinckle K (2003) Synchronous gestures for multiple persons and computers. In: ACM UIST. ACM, pp 149–158
16. Ishii H, Kobayashi M (1992) Clearboard: a seamless medium for shared drawing and conversation with eye contact. In: CHI. ACM, pp 525–532
17. Ju W, Lee BA, Klemmer SR (2008) Range: exploring implicit interaction through electronic whiteboard design. In: Proceedings of the 2008 ACM conference on computer supported cooperative work. ACM, pp 17–26
18. Kendon A (1990) Conducting interaction: patterns of behavior in focused encounters, vol 7. CUP Archive
19. Kortuem G, Kray C, Gellersen H (2005) Sensing and visualizing spatial relations of mobile devices. In: UIST. ACM, pp 93–102
20. Kunz A, Nescher T, Kuchler M (2010) Collaboard: a novel interactive electronic whiteboard for remote collaboration with people on content. In: 2010 International Conference on Cyberworlds (CW). IEEE, pp 430–437
21. Marquardt N, Ballendat N, Boring S, Greenberg S, Hinckley K (2012) Gradual engagement: facilitating information exchange between digital devices as a function of proximity. In: Proceedings of ITS '12
22. Marquardt N, Hinckley K, Greenberg S (2012) Cross-device interaction via micro-mobility and f-formations. In: Proceedings of UIST '12
23. Morikawa O, Maesako T (1998) Hypermirror: toward pleasant-to-use video mediated communication system. In: CSCW. ACM, pp 149–158
24. Neyfakh L (2014) My day as a robot, May 2014. http://www.bostonglobe.com/ideas/2014/05/10/day-robot/6UAMgmUFn0mZhoMS8vy0GK/story.html
25. Raskar R, Welch G, Cutts M, Lake A, Stesin L, Fuchs H (1998). The office of the future: a unified approach to image-based modeling and spatially immersive displays. In: Proceedings of SIGGRAPH '98
26. Reeves B, Nass C (1996) The media equation: how people treat computers, television, and new media like real people and places. Cambridge Univ Press
27. Roussel N, Evans H, Hansen H (2004) Proximity as an interface for video communication. MultiMedia, IEEE 11(3):12–16
28. Sommer R (2002) Personal space in a digital age. In: Handbook of environmental psychology, pp 647–660
29. Sousa M, Mendes D, Ferreira A, Pereira JA, Jorge J (2015) Eery space: facilitating virtual meetings through remote proxemics. In: Human-Computer Interaction–INTERACT 2015. Springer
30. Sousa M, Mendes D, Ferreira A, Pereira JM, Jorge J (2015) Human-Computer Interaction—INTERACT 2015: 15th IFIP TC 13 international conference, Bamberg, Germany, 14–18 Sept 2015, Proceedings, Part III, chapter eery space: facilitating virtual meetings through remote proxemics. Springer International Publishing, Cham, pp 622–629
31. Tanner P, Shah V (2010) Improving remote collaboration through side-by-side telepresence. In: CHI'10 extended abstracts on human factors in computing systems. ACM, pp 3493–3498
32. Vogel D, Balakrishnan R (2004) Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In: UIST. ACM, pp 137–146
33. Wilson AD, Benko H (2014) Crossmotion: fusing device and image motion for user identification, tracking and device association. In: ICMI, ICMI '14, New York, NY, USA. ACM, pp 216–223
34. Wolff R, Roberts DJ, Steed A, Otto O (2007) A review of telecollaboration technologies with respect to closely coupled collaboration. Int J Comput Appl Technol 29(1):11–26

Chapter 5
Content Sharing Between Spatially-Aware Mobile Phones and Large Vertical Displays Supporting Collaborative Work

Ricardo Langner, Ulrich von Zadow, Tom Horak, Annett Mitschick and Raimund Dachselt

Abstract Large vertical displays are increasingly widespread, and content sharing between them and personal mobile devices is central to many collaborative usage scenarios. In this chapter we present *FlowTransfer*, bidirectional transfer techniques which make use of the mobile phone's position and orientation. We focus on three main aspects: multi-item transfer and layout, the dichotomy of casual versus precise interaction, and support for physical navigation. Our five techniques explore these aspects in addition to being contributions in their own right. They leverage physical navigation, allowing seamless transitions between different distances to the display, while also supporting arranging content and copying entire layouts within the transfer process. This is enabled by a novel distance-dependent pointing cursor that supports coarse pointing from distance as well as precise positioning at close range. We fully implemented all techniques and conducted a qualitative study documenting their benefits. Finally, based on a literature review and our holistic approach in designing the techniques, we also contribute an analysis of the underlying design space.

R. Langner (✉) · U. von Zadow · T. Horak · A. Mitschick · R. Dachselt
Interactive Media Lab, Technische Universität Dresden, Dresden, Germany
e-mail: langner@acm.org

U. von Zadow
e-mail: uzadow@acm.org

T. Horak
e-mail: horakt@acm.org

A. Mitschick
e-mail: annett.mitschick@tu-dresden.de

R. Dachselt
e-mail: dachselt@acm.org

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_5

5.1 Introduction

Hardware advances are making very large vertical displays more common in a variety of scenarios. Thanks to their size, they support collaborative work [25]. At the same time, personal devices such as mobile phones have become ubiquitous over the last decade, as they allow people to conveniently manage their digital identities and content. In combination, the two device classes provide the advantages of both settings: among others, personalized interaction, on-demand data sharing, and collaboration. In this context, there is a need to be able to effectively copy and share digital content between mobile phones and large displays. Thus, it is not surprising that data exchange across such devices has been explored before (e.g., [15, 34, 38]). Still, important issues have not been sufficiently addressed, including seamless support for interaction at varying distances [4, 21], casual versus focused and precise interaction [31], the distinct support of multi-item transfer, and working with layouts on large displays.

This becomes evident when we consider that the context of a transfer operation directly influences the vocabulary of interactions. The following scenarios illustrate this: Single-item distant transfer using pointing [37] might be suitable when sitting in a meeting and close-range transfer by touching [35] is adequate for precise interactions. At the same time, a presenter at a software design review might prefer transferring a multitude of items in a predefined layout (Fig. 5.1b), while people casually showing holiday photos might find the option attractive to 'spray' them on the large display in quick succession. Alternatively, they might like to select the images to show based on a map of photo locations (Fig. 5.1c). In still other contexts (e.g., after brainstorming sessions), participants might want to transfer complete layouts to their mobile devices to preserve the spatial relationships between the different items. Finally, in the case of a public display showing product information (Fig. 5.1d), one could imagine quickly transferring interesting groups of items using coarse pointing with the mobile phone.

Using a holistic approach, we explored the aforementioned challenges and scenarios to develop *FlowTransfer*, a set of five novel interaction techniques. Our exploration focuses on three main aspects: multi-item transfer, interactions at varying

Fig. 5.1 *FlowTransfer*, a set of bidirectional transfer techniques using the spatial display configuration. **a** Study setup. **b** Item layout *(Layout and SnapshotTransfer)*. **c** Metadata-based transfer *(RevealingTransfer)*. **d** Fast multi-item transfer *(JetTransfer)*

distances, and casual as well as precise interactions. In addition, the contributed multiway transfer techniques integrate item layout and require a minimum of gaze switches [33]. Furthermore, they do not require a touch-sensitive display wall, since touch is not available or appropriate in numerous situations. The techniques all exploit spatially-aware mobile devices: by assuming devices that have knowledge of their location and orientation, we can exploit phone-based pointing and mid-air gestures, among other features. Accordingly, one of our contributions is a novel distance-dependent pointing cursor that supports physical navigation by allowing coarse pointing from distance as well as precise positioning at close range.

Besides the *FlowTransfer* techniques themselves, we contribute a fully functional prototype implementation as well as a qualitative user study. Finally, based on a careful analysis of prior work as well as our own experiences and studies, we present a comprehensive design space for content sharing techniques between large displays and mobile devices, which can inform the design and development of future systems. We conclude with a discussion of the implications of our design decisions.

5.2 Related Work

Interaction with large displays and personal mobile devices (e.g., content creation, content sharing, object manipulation) is an active research field. In the following, we discuss local and distant transfer techniques, which have been explored in prior work as well as research on distant pointing.

Interaction with Large Displays. For a general introduction to interaction with wall-sized displays, we refer to overviews by Müller et al. on public displays [25] and Andrews et al. on data visualization [2]. Additionally, Marquardt et al.'s work on Gradual Engagement [24] provides a design framework for integrating the relative positions of the devices involved in cross-device interaction. A related notion is Greenberg et al.'s Proxemic Interaction (e.g., [14]), in which interactions are based on spatial relationships between people and devices. Ball et al. [4] examined Physical Navigation—moving the body for interaction—and found that locomotion significantly improves performance when interacting with large displays. Rädle et al.'s work on Navigation Performance [32] finds that the effects are most pronounced when the task exercises spatial memory, e.g., in navigation tasks not involving zooming. At the same time, Jakobsen and Hornbæk [18] found no advantages for locomotion; their task did not involve spatial memory.

Data Transfer. Much of the work on cross-device data transfer considers single-item transfer in close proximity. Rekimoto's Pick-and-Drop [34] is early work on cross-device data transfer using a pen as interaction device. More recently, Schmidt et al.'s PhoneTouch associates touches on a large display with a mobile phone by correlating the phone's motion sensor signals, covering both the technology [35] and numerous interaction techniques [36]. In SleeD [41], von Zadow et al. use an arm-worn device; transfer involves touching the large display with the hand the device is strapped on. With WatchConnect, Houben and Marquardt [17] provide a toolkit for developing cross-device applications with smartwatches. Alt et al. [1] compare content creation for and exchange with public displays using multiple modalities, while Seifert et al. [38] introduce a number of interaction techniques that allow privately selecting the data to share before performing the actual transfer.

With regard to data transfer operations at a distance, several researchers investigated the use of the mobile device's camera [3, 5, 8, 9]. In Shoot and Copy [8], image recognition on the camera image is used to extract semantic data, while Touch Projector [9] and Virtual Projection [5] explore remote interaction through a live video image on a mobile device. Distant transfer using device gestures has also been investigated several times [12, 16]. Dachselt and Buchholz's Throw and Tilt [12] utilizes expressive gestures for data transfer, while Hassan et al.'s Chucking [16] is interesting because it also supports positioning of items on the large screen. Finally, CodeSpace [10] integrates distant transfer in an application case, using a depth-sensing camera to support positioning, and Jokela et al. [19] compare transfer methods. However, none of the above approaches sufficiently address layouts for transferred items or focus on transfer at varying distances and locations. To our knowledge, neither differences between casual or focused interactions nor gaze switches are focused on in prior work.

Distal Pointing. Distal pointing allows selection and positioning of items and is therefore significant in our context. Hand pointing is investigated in early work by Bolt [7], and, more recently, by Vogel and Balakrishnan [40]. Nancel et al. investigated distant pointing using handhelds; the authors contribute several techniques for precise selection from a distance [27, 28]. In PointerPhone [37], Seifert et al. investigate the interactions possible when remote pointing is combined with interactions on the phone. Myers et al. [26] found that in distant pointing, precision suffers because of hand jitter. Most techniques that support positioning either live with this restriction (e.g., [7, 37]) or introduce a second step for improved precision (e.g., [9, 26, 27]). Particularly, Myers et al. [26] use the mobile phone's camera to copy an area of interest to the small display, allowing touch selection on the small display. Peck et al.'s work [30] is one of the few that combines pointing and physical navigation. Furthermore, Lehmann and Staadt [23] propose different 2D manipulation techniques that use varying interaction precision based on the user-display distance. To our knowledge, however, a cursor that maps distance to pointing precision has not been presented in prior work.

5.3 Development Process and Design Goals

As part of an iterative design process, we developed the concepts and improved our implementation supported by a preliminary user study. In this preliminary user study (12 participants, laboratory setting, sessions lasted around 60 min), participants tested an early prototype that supported two single and two multi-item transfer techniques and provided initial user feedback. Participants explored transfer techniques, performed technique-specific tasks such as transfer all images with specific features to the mobile device, and completed a questionnaire. Based on the results of this study and prior work, we specified formal design goals, refined our concept and developed an improved, fully functional prototype. Finally, a qualitative user study was conducted to evaluate the final implementation and verify its usefulness.

In the following, we present six design goals, referred throughout the paper as D1–D6. These goals informed our design; we examine the implications and results in our qualitative study and the discussion that followed. Of our six design goals four (D1, D2, D3, and D4) correspond directly to the three aspects (casual/focused interaction, interactions at varying distances, as well as multi-item transfer and layout) we focused on from the outset. The last two design goals (D5 and D6) are grounded in the preliminary study as well as our analysis of related work.

(D1) *Adapt to user's level of engagement*: The scenarios presented in the introduction show that users have different requirements and priorities in different situations (e.g., speed vs. precision). Therefore, we would like users to be able to "control the level to which they engage" [31]: our techniques should enable effortless and casual (e.g., coarse positioning) as well as focused (e.g., precise, exact selection on a crowded display) interaction.

(D2) *Support interaction at varying distances*: Related work shows the benefits of locomotion in terms of performance [4] and spatial memory [32]. Therefore, content sharing techniques should bring out the benefits of working at varying distances; they should work well with both the overview that users have at a distance and the detailed view they have when close to the display, and they should adapt seamlessly.

(D3) *Adapt to number of items transferred*: The scenarios mentioned above as well as transfer operations in conventional desktop systems show that considering both single-item and multi-item transfer is necessary to cover a wide range of use cases. Support for this is largely missing in the literature.

(D4) *Support item arrangement*: As illustrated by some of the scenarios and as we learned from desktop techniques such as drag and drop, item positioning and layout naturally complement data transfer. We assume that the huge workspace provided by large displays will make seamless positioning and layout support even more important. Furthermore, this aspect has not been investigated in the literature. Therefore, we aim to integrate layout functionalities into our techniques.

(D5) *Minimize gaze switches*: Gaze switches are an integral part of working with multi-display setups. However, our preliminary study as well as the literature [33, 39, 41] show that they are disrupting and time-consuming. Therefore, we aim to minimize the number of necessary gaze switches.

(D6) *Map user movements to appropriate parameters*: To develop both comprehensible and easy to remember techniques (and help to bridge the gulf of execution [29]), the devices' movement directions should correspond to changed parameters, thus avoiding a mental break between user movements and system reactions. Examples include mapping precision to the distance from the large display (corresponding with D2) or mapping an upwards flick on the mobile phone to transfer towards the large display.

5.4 FlowTransfer Techniques

To explore the space of cross-device data transfer with large displays, we developed five novel techniques. Besides the goal of developing individual techniques, we were interested in exploring the underlying design space. Therefore, we focused on the design goals described above during development.

The only single-item technique, *FlickTransfer*, is an incremental improvement on prior work and serves as baseline. With *JetTransfer*, *LayoutTransfer* and *SnapshotTransfer*, we present three techniques that work with groups of items and their arrangement (D3, D4). JetTransfer sequentially transfers a multitude of items to the large display, LayoutTransfer adds the possibility to evenly arrange items on the large display along a motion path upon transfer, and SnapshotTransfer preserves the layout of items on the large display when transferred to the mobile device. The final technique, *RevealingTransfer*, illustrates the combination of different techniques and allows selection of transferrable items based on predetermined locations.

All *FlowTransfer* techniques share a number of characteristics including: common feedback mechanisms, a unified approach for minimizing gaze switches (D5), and a new distance-dependent pointing cursor controlled by the mobile phone (representing the focus of interaction on the large display). Visual cues on the large display include an unobtrusive cursor visualization in the form of a shaded circular area (e.g., Fig. 5.1d). To inform users of different application states, this cursor visualization dynamically makes use of different visual properties (e.g., border color for transfer activities, border width for current transfer rate). Furthermore, when transferring to the large display, a preview is shown at the destination (Fig. 5.3a). We propose to blur this preview in order to avoid privacy issues in multi-user scenarios. To also address both privacy and visibility, the strength of this effect can depend on user positions. Items selected for transfer from the large display are highlighted (e.g., border color; Fig. 5.3b). Finally, the prototype delivers vibrotactile feedback whenever the selection is changed and upon transfer.

The techniques are designed to allow users to focus on the large display, minimizing the need for gaze switches (D5). This precludes giving essential visual feedback on the mobile phone during transfer operations. For the same reason, we avoid traditional GUI elements such as buttons on the phone; instead, our techniques use touch gestures that can be performed while focusing on the wall. However, when transferring items to the large display, users will generally select the items to be transferred on the mobile phone for privacy reasons [38]. We expect the details of this to be application-specific. Possibilities include employing classical, widget-based techniques to select contiguous or non-contiguous ranges of items, or using search techniques to select items that satisfy desired criteria. In any case, we generally assume a single gaze switch after the selection.

5.4.1 Distance-Dependent Pointing Cursor

Central to transferring data to a large display is the specification of a target position. Building on previous work in distal pointing (e.g., [22, 23, 40]), we developed a *distance-dependent pointing cursor* that provides a smooth transition between projective and orthogonal pointing (Fig. 5.2). It is designed to bring out the benefits of working at varying distances (D2) and compensate the effects of hand jitter [27]. The pointing cursor works using three distance zones: at overview distance (when they can see the complete display), users can directly point at a target (i.e., projective pointing or ray-casting; similar to PointerPhone [37]). At close distance, the orientation of the mobile is ignored and an orthogonal projection is used to determine the selected position, thus increasing precision of cursor control and reducing jitter. At intermediate distances, we interpolate linearly between the two projection methods (reducing motor space from 6 to 2DoF), thereby ensuring smooth transitions between the aforementioned zones. The goal was to allow users to employ various distances (D2) to implicitly transition between modes of control and thus determine the level of precision they require (D1, D6).

The cursor position is directly used as destination position when transferring items to the large display. In the opposite direction, we use an area cursor [20] to target items: The item closest to the center of the activation radius (e.g., visible in Fig. 5.3a, right) is selected. Initial user feedback showed that this technique allows for coarse, less precise pointing (D1) when compared to simply selecting the image under the cursor, compensating for hand jitter. Finally, to support varying interaction distances (D2), the activation radius (size) of the cursor continuously scales with distance (Fig. 5.2).

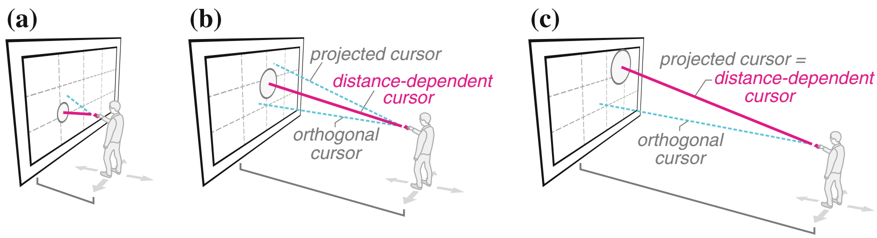


Fig. 5.2 *Distance-dependent pointing cursor* at different user positions: **a** close distance, **b** intermediate distance, **c** overview distance

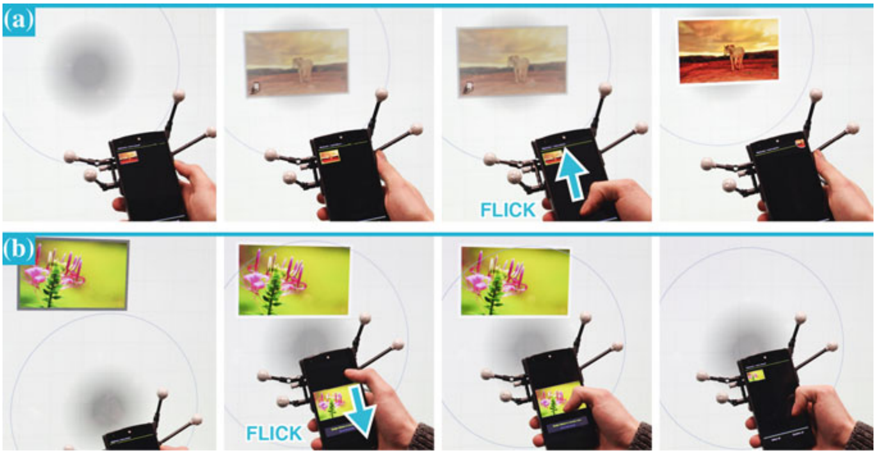


Fig. 5.3 *FlickTransfer* is a state-of-the-art technique for single-item transfer

5.4.2 Individual Transfer Techniques

This section details the proposed five data transfer techniques and discusses design implications. Please also refer to the accompanying video illustrating the dynamics of the interaction techniques.[1]

FlickTransfer is a simple technique for single-item transfer that uses phone pointing and flicks (swipes) on the mobile phone. To transfer an item to the large display, users first select it on the phone, then point the device towards the large screen and flick upwards (i.e., towards the destination) to initiate the transfer (Fig. 5.3a). Conversely, flicking downwards transfers the currently selected item from the large display to the phone (Fig. 5.3b). We considered alternative gestures, but the only significantly simpler solution—tapping the screen—does not allow the user to distinguish between transfer directions. To increase pointing precision, an additional damping mode can be activated by a hold before the flick gesture. In this mode, the pointing cursor movement is decreased greatly.

FlickTransfer is an incremental improvement over techniques presented in prior work (e.g., [10, 11]). The technique extends existing approaches by using our pointing cursor as well as the blurred preview, and providing an additional damping mode. Development of this technique allowed us to focus on and refine the common feedback mechanisms, the operation of the pointing cursor, and the minimization of visual attention switches. In FlickTransfer, precision and thus level of engagement (D1) can be controlled by moving towards or away from the screen (D2). Furthermore, the mapping of flick direction to transfer direction is designed to be direct and easy to understand (D6), since users simply flick in the desired direction of transfer.

[1] see https://imld.de/flowtransfer/.

JetTransfer transfers multiple items in quick succession using 'spray paint' and 'vacuum cleaner' metaphors. Transfer to the large display is initiated by touching the phone, sliding the finger upwards and holding it. While holding, selected items are transferred in quick succession, using the pointing cursor position as drop point (Fig. 5.4a). The transfer rate is specified by the distance between the initial and the current touch position, i.e., the length of the slide motion. By moving the phone, items can be 'sprayed' on the large screen. Conversely, transferring items back to the mobile phone involves a 'vacuum cleaner' mode (similar to [6]) that is activated by touching the phone screen and sliding the finger down. Items in the cursor's active radius are attracted towards the center (shown using moving grayscale previews, Fig. 5.4b); when they reach it, they are transferred. If the touch is released or the cursor is moved so the item is outside of the active radius, attracted items snap back to their original positions.

Again, parameter mapping is designed to be direct and easy to understand (D6): the radius of attraction increases with the distance to the large display (D2), and transfer rate is based on thumb-dragging on the phone screen, specifically the distance between the initial touch point and the current thumb position. This allows users to choose between speed and precision (D1). Both very fast bulk transfer of items (farther away and with fast transfer speed) and slower, more controlled transfer of single items (close to the large display with slow transfer speed) are possible. Furthermore, it supports casual spatial arrangement of items (D4). A typical use case is the transfer of multiple images with different subjects, with the images loosely grouped by subject on the large display.

LayoutTransfer enables users to effectively create an orderly layout for a group of items. It expands upon FlickTransfer and employs a phrased gesture design (see the corresponding state diagram in Fig. 5.5) to transfer items to the large display.

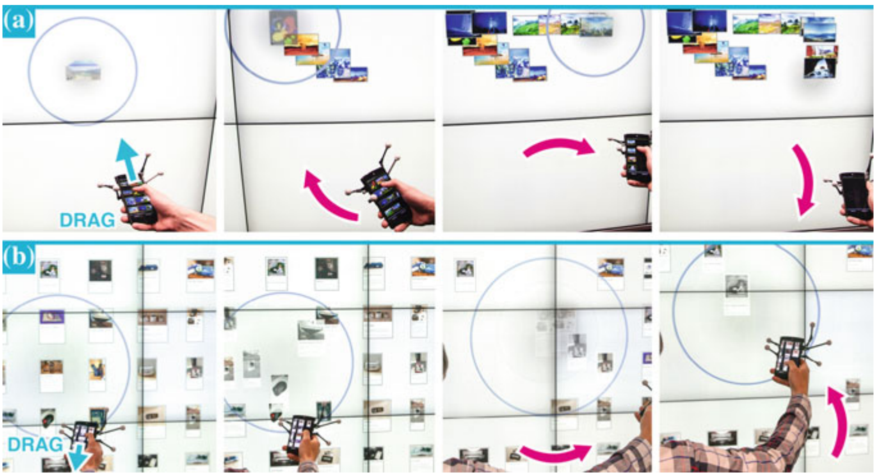


Fig. 5.4 *JetTransfer* uses rough and casual positioning for multi-item transfer

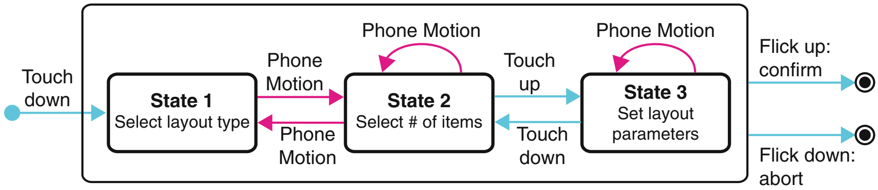


Fig. 5.5 *SpanningTransfer* state diagram: All states can be accessed at all times. The operation can be confirmed with a swipe up or aborted with a swipe down gesture at any time

Interaction begins by touching and holding the mobile phone. Users can determine the number of items to transfer and their arrangement on the large display by moving the pointing cursor. The type of layout is determined by the initial direction of movement (State 1, shown in Fig. 5.6a), and the number of items is determined by the movement distance (State 2). Layout parameters can be adjusted when the finger is released from the phone screen (State 3): Pointing cursor position controls item spacing and phone distance controls item size (Fig. 5.6b). Users can switch between number of items (State 2) and layout parameter (State 3) modes at any time by touching and releasing the mobile phone. Finally, flicking upwards on the phone confirms the transfer, while flicking downwards aborts it.

LayoutTransfer was inspired by some of the NEAT multi-touch gestures [13]. Its phrased gesture design allows users to interleave several interaction sub-tasks seamlessly while allowing abort at any time. This makes it possible to quickly set multiple parameters for spatial arrangements (D4): number of items, layout type, item spacing and size can all be set in a phrased interaction, making it useful when working with organized, sorted groups of data items (D3).

SnapshotTransfer allows users to easily transfer multi-item layouts from the large display to the mobile phone and back. Similar to selection techniques in conventional desktop systems, users create a rectangular selection area by pointing the phone towards one corner of the layout, touching and holding the phone screen, and pointing towards the opposite corner (Fig. 5.7a). Alternative methods for a more

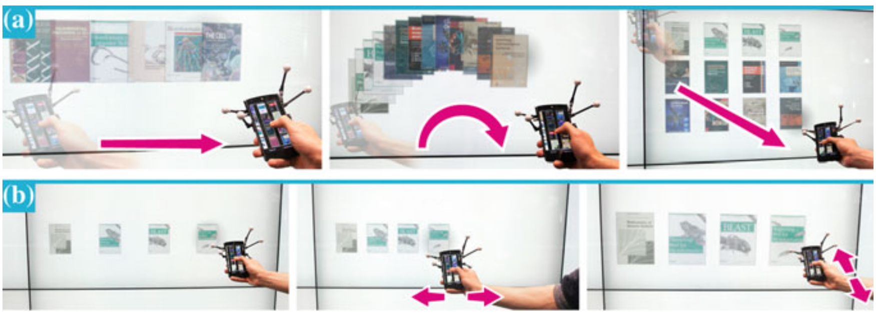


Fig. 5.6 *LayoutTransfer* enables simultaneous transfer and arrangement of multiple items

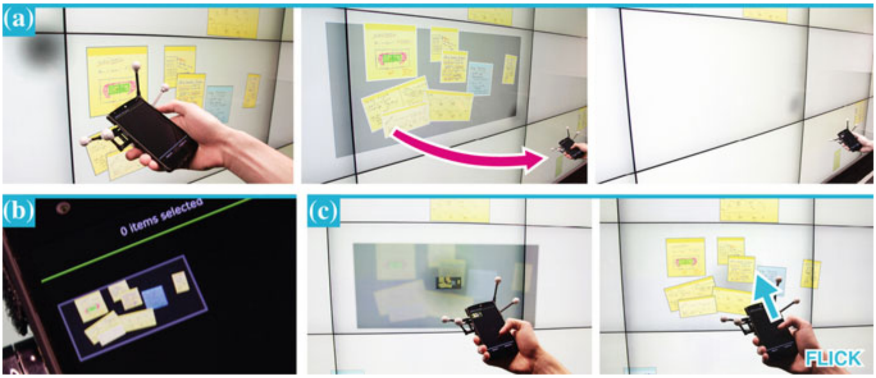


Fig. 5.7 *SnapshotTransfer* copies multiple items and their layout

refined selection include using a lasso metaphor or running the cursor over all items to select them. Releasing the finger transfers the items and their layout as a single entity (Fig. 5.7b). The user can move the cursor back to the initial position and release the touch to abort at any time. The layout can be transferred back to the large display using FlickTransfer. In this case, the complete group of items is shown as preview (Fig. 5.7c).

SnapshotTransfer provides a quick and easy way to preserve item layouts created on the large display (D4). It is also useful if complete layouts need to be moved from one place on the large display to another.

RevealingTransfer supports transfer of items to predetermined locations on the large display based on item metadata (e.g., the transfer of geotagged photos to a map). Transfer proceeds in two phases: metadata of items are automatically transferred first, allowing the large display to show item marks (here: blue circles) at corresponding positions. To preserve privacy, item previews are not revealed before the pointing cursor reaches their location (Fig. 5.8a, b). For the actual transfer, we incorporate elements of FlickTransfer and JetTransfer to support single and multi-item transfer (D3): Tapping the mobile phone transfers the item closest to the pointing cursor (Fig. 5.8c) while flicking upwards transfers all items contained in the cursor

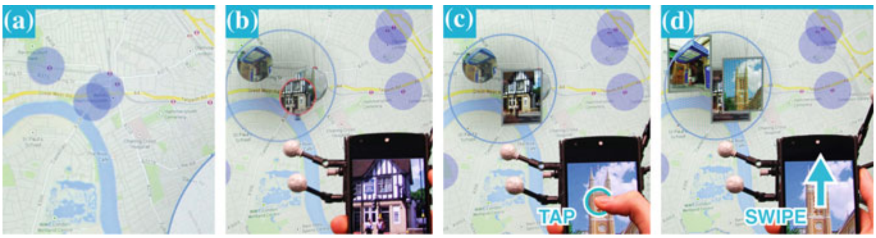


Fig. 5.8 *RevealingTransfer* combines techniques and allows metadata-based transfer

radius (Fig. 5.8d). Additionally, a swipe and hold transfers a sequence of items along the cursor's path (JetTransfer).

RevealingTransfer is designed for situations where the selection and positioning of items is influenced by their metadata. Furthermore, it illustrates the combination of different techniques using simple touch gestures to switch modes.

5.5 Implementation

Our prototype runs on a 5 × 2 m large display wall, consisting of twelve 55" 1080p displays (Fig. 5.1a) driven by a dual-processor Xeon workstation running Ubuntu. As a mobile device, we use an LG Nexus 5 smartphone with a 5" display running Android. We track the phone's absolute position in space using 12 infrared OptiTrack[2] cameras; accordingly, the phone is instrumented with reflective markers. On the software side, we use the Python-based libavg[3] framework for the user interface. The OptiTrack system runs on a separate stand-alone computer, which streams the phone's 3D position using the OSC[4] protocol.

Implementation-specific parameters are as follows. The pointing cursor uses raycasting at a distance of 3.5 m and more; at 1.0 m or closer, it uses orthogonal pointing (Fig. 5.2). Layout parameter adjustment in LayoutTransfer scales the images by a factor of 2 for every 10 cm of distance change. FlickTransfer's damping mode only applies 2.5 % of the current cursor movement, while JetTransfer allows transfer speeds between 4 to 15 images/sec (to the large display). Depending on the distance, the size of its active radius continuously scales from 15 to 60 cm. In the case of SnapshotTransfer, the current prototype only supports rectangular selection. We implemented a minimal item selection interface on the phone: items are arranged in a scrollable grid and selectable by tapping. As data items, we use sets of images appropriate to the individual use cases.[5]

5.6 User Study

To evaluate our techniques and identify practical implications, we conducted a qualitative study in a laboratory setting. Among other things, we wanted to know how well varying distances were supported (D2), what impact eyes-free interaction

[2] http://www.optitrack.com/.

[3] https://www.libavg.de/.

[4] http://opensoundcontrol.org/.

[5] Photos from Flickr by @dhilung (https://www.flickr.com/photos/dhilung), @mualphachi (https://www.flickr.com/photos/mualphachi), and @duncanh1 (https://www.flickr.com/photos/duncanh1), Map by GoogleMaps (https://maps.google.com).

had (D5), and if the parameter mapping was indeed comprehensible and easy to understand (D6). Furthermore, we wanted to ascertain that the distance-dependent pointing cursor worked as intended and the visual feedback given was helpful.

Since our techniques span a variety of application cases and user scenarios, differ in complexity, and useful prior work for comparison was not available in several cases (e.g., multi-item or layout transfer), we opted against a quantitative study comparing techniques for performance. Instead, we focused on the design goals mentioned above, investigated usefulness and acceptance of the techniques, and collected rich feedback for all of them.

Method. Seven students from the local university (1 female, 1 left-handed) volunteered for the study, which took on average 75 min per person. The average age was 25 years ($M = 25.14$, $SD = 2.59$). All participants use smartphones and computers daily. Two use monitor-sized touchscreens daily, and six had already used a wall-sized display.

To evaluate our techniques, we developed a within-subject and repeated measures study design. To ensure a natural progression of techniques and due to interdependencies between them (e.g., RevealingTransfer utilizes elements of FlickTransfer and JetTransfer), we did not counterbalance the order of presented techniques. The procedure was the same for each technique: A short training phase was followed up by technique-specific tasks, a brief phase of free exploration, a discussion, and a short questionnaire. About half of participants' time (approx. 35 min) was spent on the actual tasks. Besides an overall rating of a technique, we asked participants to rate understandability, easiness, control, target compliance, and perceived speed. We further integrated questions on performance, physical demand, and mental demand based on NASA TLX. For all these ratings we used a 5-point likert scale (1–strongly agree, to 5–strongly disagree). We logged phone motion data, recorded the sessions on video and asked participants to think aloud. Each session was accompanied by two researchers, with one exclusively observing behaviors of participants and taking notes.

Tasks were specific to the use cases of the corresponding techniques and were tailored to verify that the techniques' specific capabilities worked as intended. For FlickTransfer, we asked participants to transfer five tagged monochrome images from the mobile phone to corresponding target areas on the large display and subsequently five tagged colored images back to the phone. For JetTransfer, the phone initially contained multiple colored images (20) followed by monochrome images (20). Participants had to transfer colored images to the left and monochrome images to the right half of the large display. Next, the large display showed a widespread mixture of colored and monochrome images; participants had to transfer all monochrome ones to the mobile phone. The task for LayoutTransfer was to create specified matrix (5×3) and row (9) layouts from images stored on the phone. In the case of SnapshotTransfer, participants had to transfer a specified group of notes (6 items) to the phone and back to a different location on the large display. For Revealing-Transfer, the large display showed a map of London with 53 markers of transferrable geotagged images. Participants had to transfer all images taken at specific locations, e.g., 13 images along the River Thames.

General Results. Altogether, we received rich and very encouraging feedback. Fatigue seemed not to be an issue; it was neither mentioned by participants nor noticeable in the videos. Participants realized the effects of physical navigation (D2). This was evidenced in comments implying, e.g., that they could gain an overview by stepping back and that walking around took time. However, we observed—and the motion data confirmed—that moving was generally avoided when possible. Users often moved closer to gain precision only after they had committed errors. In this respect, our results are similar to Jacobsen et al. [18], who also found that users do not prefer physical navigation. On the other hand, it is possible that this would change with longer usage as users learn to anticipate an optimal, task-specific distance, and that a study setup that required more precise interaction would have resulted in more physical navigation. By determining patterns of easily visible gaze switches through observation and video analysis, we could ascertain that the goal of minimizing gaze switches (D5) was achieved. This was also commented upon positively by five participants. Mobile phone usage was almost exclusively one-handed, indicating support for casual interaction (D1). The distance-dependent pointing cursor was commented on favorably by four participants, but most participants did not notice the interpolation as such; instead, the increased precision at close distance was observed. Mapping the user's distance to cursor size was mentioned positively by two participants. However, we observed that touches on the mobile phone had a minor impact on the cursor position, affecting precision to a certain degree. FlickTransfer's damping mode helped here.

Furthermore, it became evident that our techniques provide sufficient feedback. Excepting JetTransfer, users generally had no issues determining what the current application state was or which item would be affected by the next action. The pointing cursor's circular feedback was commented upon favorably by three participants.

Results for Individual Techniques: Our observations and comments showed that participants were intrigued by the techniques and with few exceptions able to use them to achieve the goals. This is confirmed by the results of the questionnaire (selected results in Fig. 5.9), which provide additional interesting insights as well. However, due to the limited number of participants the quantitative results do not allow generalizations.

FlickTransfer was described as simple and easy-to-use by six participants. Additionally, four participants commented favorably on the damping mode.

Participants found *JetTransfer* to be enjoyable, very fast and casual, but also inaccurate. The fun aspect was mentioned five times; it was also very visible in the reactions of several users. However, control suffered: all participants had issues stopping the transfer at intended items and the questionnaire results reflected this as well (avg. ratings: 4.00 for transfer of single items; 3.43 for control over the transfer). Accurate placement was difficult because the exact transfer time was hard to predict, and unclear visual feedback was mentioned twice as drawback. JetTransfer was introduced late in the development cycle, and we believe that controllability can be improved significantly by lowering the minimum transfer rate and improving feedback.

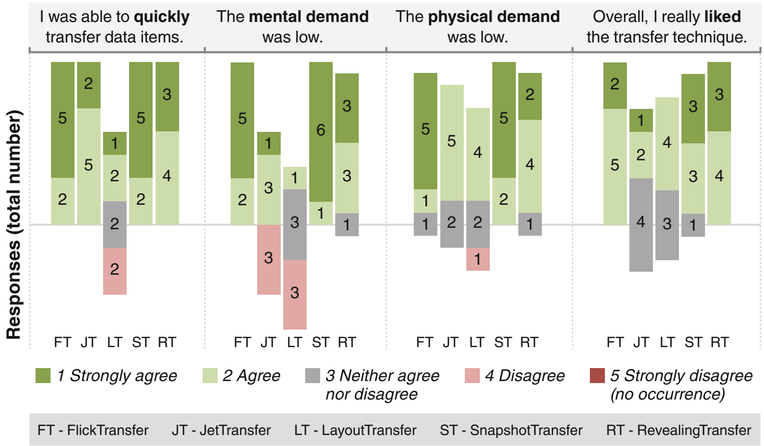


Fig. 5.9 Selected ratings of the techniques

The general response to *LayoutTransfer* and the idea of transferring and specifying a layout in one seamless interaction was positive (five mentions). The technique has a clear learning curve (avg. rating understandability: 2.86) and is thus not suitable for walk-up-and-use situations. After a short time, however, users were able to use all features; three users commented that the technique was 'not too complicated'.

SnapshotTransfer was found to be very easy to use (avg. rating understandability: 1.29). The familiar selection method ("like under windows") was mentioned four times. Additionally, five users mentioned that a lasso mode could be useful, confirming our concept.

Regarding *RevealingTransfer*, it generally took users some time to grasp the concept of pre-located items. Once understood, however, the technique was viewed positively. Six participants commented favorably on the different transfer modes (single and multi-item).

5.7 Design Space

Based on the development of the transfer techniques, prototype implementation, and findings of the user study, we contribute a design space for content sharing techniques between mobile devices and large displays (Fig. 5.10). We identify, refine, and systematically discuss essential design dimensions to abstract the specific interaction techniques into a reusable framework and to allow comparative discussions of existing techniques. Our aim is to support design decisions and allow the identification of gaps for the development of future techniques. In addition, Fig. 5.10 maps content sharing techniques to the design dimensions and makes this assignment a subject of discussion.

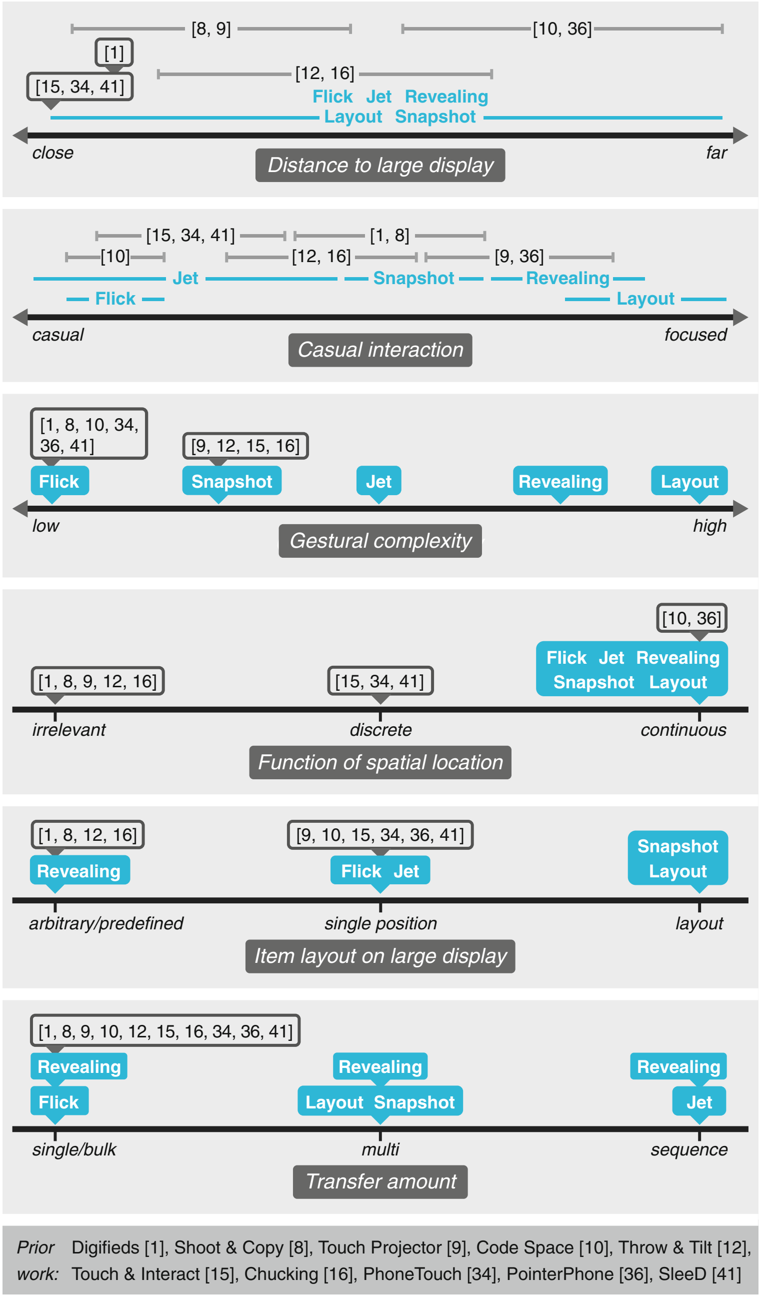


Fig. 5.10 Dimensions of the design space for content sharing techniques involving mobile devices and large displays. Existing techniques (*gray*) and techniques presented in this work (*blue*) are mapped to the dimensions based on their characteristics as well as typical usage

Distance to large display. This describes the usage of various distances during interaction (D2). This continuous dimension ranges from *close* (i.e., users can touch the large display) to *far* (i.e., users see complete wall).

Casual interaction. Introduced by Pohl and Murray-Smith [31], this dimension describes the user's level of engagement for interactions (D1). This continuous dimension ranges from *casual* (i.e., only minimal attention is required) to *focused*.

Gestural complexity. This describes the level of complexity of gestural input needed to initiate and control transfer actions. For instance, the number of involved devices or body parts (cf. [42]) affects gestural complexity. This continuous dimension ranges from *low* (e.g., tap a virtual button) to *high* (e.g., draw a complex shape using touch). While complex input often requires much of the user's attention, it usually provides more interaction capabilities, for example, define and manipulate the arrangement of transferred items (LayoutTransfer, D4).

Function of spatial location. This describes the higher-level function and usage of the spatial location of mobile devices in relation to large displays. We distinguish three values: *irrelevant*, *discrete*, and *continuous*. Spatial location is irrelevant if applications are not aware of or ignore locations of mobile devices, e.g., transfers through a wireless connection or QR codes. Discrete location mapping can be used as on/off switches to control specific application states, modes, tools, or conditions [14, 24]. The mapping is continuous if location controls a cursor [15, 34, 36], pointer [27, 37], or another continuous application-specific parameter.

Item layout on large display. This describes the use and type of item positioning for transfers to large displays. This dimension spans three discrete values: *arbitrary/predefined* if techniques do not allow users to specify an item position, *single position* if an item can be positioned, and *layout* if multi-item arrangements—possibly including layout parameter adjustments (e.g., size, spacing) are supported.

Transfer amount. This describes the number of items that can be transferred in a single interaction step (D3). With respect to discrete items (e.g., images, files), this dimension distinguishes three discrete values: *Single/Bulk* represents techniques that focus on transfer of individual items (e.g., photo, song) or data containers (e.g., folder), *Multi* includes techniques with 'distinct' support for a collection of items while considering or specifying additional item attributes (e.g., spatial relation), and *Sequence* techniques support the successive transfer of multiple items. For data of continuous nature (e.g., movies, music), this would describe whether users can specify a portion of an item to be transferred.

5.8 Discussion

The proposed techniques are influenced by various concepts including physical navigation [4], casual interaction [31], and proxemic interactions [14]. Since the techniques cover a broad range of different characteristics, we believe that they highlight the variety and richness of the underlying design space. At the same time, the design space reveals that our techniques fill out existing gaps (e.g., transfer amount, item layout). In this section, we further discuss the relationship between different design space dimensions, design issues, and valuable lessons learned.

5.8.1 Design Space Parameters and Consequences

Gestural complexity of user interactions seems to correlate strongly to many of our design goals and design dimensions. Both casual and almost eyes-free interaction (D1, D5) can be realized by using simple input gestures, because they require less attention and are easy to learn and remember. Utilizing physical navigation as well as mapping movement to an appropriate application parameter (e.g., input precision, zoom) seems to encourage people to perceive interactions as easy and simple (D2, D6).

For most transfer techniques, Fig. 5.10 shows a correlation of casual interaction and gestural complexity. Furthermore, there is also a correlation between casual interaction and distance to the large display, since distant interaction is very likely more casual, whereas close proximity interaction is often associated with focused input. By utilizing the flexible pointing cursor, our techniques scale and support both casual interaction at a distance and focused interaction at close proximity (D1, D2); this is not directly visible in Fig. 5.10.

5.8.2 Interaction Design

Complex interactions involving multiple parameters and modalities require careful interaction design. Mode switches can clear up a cluttered gesture space and thus allow reuse of input dimensions. As demonstrated by LayoutTransfer, this in turn allows mapping of, e.g., device movement to changing but always appropriate parameters (D6). The manner of mode switching is important: An early prototype utilized the orientation of the mobile phone to switch transfer techniques, but mandatory hand postures (preventing a casual interaction style, D1) were not received well by users. Instead, RevealingTransfer demonstrates that mode switching using different touch gestures is a simple and viable solution. Similar techniques for mode switching could be used to switch between transfer modes copy and move as well.

Similarly, LayoutTransfer maps distance to the large display and pointing cursor position to unrelated parameters (item spacing and item size) in layout mode. This requires fine motor control when the goal is to adjust only one of the parameters without affecting the other. Therefore, we already improved the prototype by locking changes to the predominant movement direction (e.g., ignore distance changes when users move the pointing cursor across the wall and vice versa). An alternative option would be to map parameters to different modalities. This is demonstrated by JetTransfer, where we successfully combined device movement (for positioning) with touch input (for transfer direction and speed).

Our study showed that we were successful in minimizing gaze switches (D5), and from early prototypes as well as prior work [33, 39, 41] we know that this has a positive effect on usability. We believe that our corresponding design principles, e.g., placing visual feedback on the large display and only gestural touch input on the phone (as opposed to GUI-based), were instrumental in achieving this. As presented, our techniques implement item selection on the phone and thus require a single gaze switch after selection. This was done to avoid privacy issues. In a trusted setting, private images could be shown on the large display and the selection performed there, avoiding even this gaze switch. Conversely, in RevealingTransfer, we show blurred preview images on the large display to facilitate selection. This would not be possible in a situation where privacy is very important, and if privacy is not an issue, it might not be necessary to blur previews at all.

5.9 Conclusion and Future Work

In this chapter we presented *FlowTransfer*, a set of five novel interaction techniques that allow users to transfer data between mobile devices and large displays—which is central to many collaborative usage scenarios. Our multiway transfer techniques combine concepts from physical navigation [4] and casual interaction [31]. They address various challenges at once, among them the rapid or slow transfer of both single and multiple items, the creation and transfer of sophisticated layouts, as well as the handling of gaze switches. In addition, the *FlowTransfer* techniques adapt to the user's level of engagement by allowing a smooth transition between casual and more focused interaction.

In the context of multi-user scenarios, our techniques support people collaboratively sharing digital content with a large display. Due to the usage of our distance-dependent pointing cursor, multiple users can transfer objects from dynamic and individual positions, thus addressing occlusion issues and allowing to work in parallel or together. Furthermore, we also consider the separation of private data on the personal device and shared data on the large display, e.g., by selecting items to be transferred on the mobile device and showing only a blurred preview of selected items on the large display.

We described our design goals and iterative design process, presented a fully functional prototype implementation of the proposed techniques, and reported on a qualitative user study. Based on the design process and study results, we contributed a design space for content sharing techniques between mobile devices and large displays. The presented dimensions of this design space, such as distance to large display, casual interaction, item layout on large display, and transfer amount, provide support for the development of future data transfer applications. Furthermore, we look forward to research that extends the space.

Our proposed distance-dependent pointing cursor was successful in the context of the transfer techniques, and allowed users to control precision by varying their distance to the large display. However, there is still room for tuning parameters such as the minimum and maximum distance for interpolation. Therefore, we believe that it deserves further analysis. Furthermore, we want to thoroughly examine the capability of the distance-dependent pointing cursor and compare it to other existing approaches.

Regarding sensing the phone's position, our current setup requires separate tracking equipment. However, we expect upcoming mobile devices to integrate reliable positional and rotational sensors (or inertial location using depth-sensing cameras) that make external sensing unnecessary. For future work, we plan to explore different strategies for a seamless selection of appropriate transfer techniques depending on specific tasks or goals. We already took a first step in this direction by developing the RevealingTransfer technique.

Finally, we believe that our techniques and the proposed design space represent both a solid foundation and inspiration for the development of future user interfaces in the expanding space of applications combining large displays and personal mobile devices.

Acknowledgements We thank Wolfgang Büschel, Ulrike Kister, Fabrice Matulic, and other members of the Interactive Media Lab for their feedback and helpful suggestions. We would also like to thank students of the lab for their support on the prototype. Finally, we thank our anonymous reviewers for their valuable and helpful comments.

References

1. Alt F, Shirazi AS, Kubitza T, Schmidt A (2013) Interaction techniques for creating and exchanging content with public displays. In: Proceedings CHI 2013. ACM, pp. 1709–1718
2. Andrews C, Endert A, Yost B, North C (2011) Information visualization on large, high-resolution displays: issues, challenges, and opportunities. Inf Vis 10(4):341–355
3. Ballagas R, Rohs M, Sheridan JG (2005) Sweep and point and shoot: Phonecam-based interactions for large public displays. In: Proceedings CHI 2005 Extended Abstracts. ACM, pp 1200–1203
4. Ball R, North C, Bowman DA (2007) Move to improve: promoting physical navigation to increase user performance with large displays. In: Proceedings CHI 2007. ACM, pp 191–200
5. Baur D, Boring S, Feiner S (2012) Virtual projection: exploring optical projection as a metaphor for multi-device interaction. In: Proceedings CHI 2012. ACM, pp 1693–1702
6. Bezerianos A, Balakrishnan R (2005) The vacuum: facilitating the manipulation of distant objects. In: Proceedings CHI 2005. ACM, pp 361–370
7. Bolt RA (1980) "Put-that-there": Voice and gesture at the graphics interface. In: Proceedings SIGGRAPH. ACM, pp 262–270
8. Boring, S, Altendorfer M, Broll G, Hilliges O, Butz A (2007) Shoot & copy: Phonecam-based information transfer from public displays onto mobile phones. In: Proceedings Mobility 2007. ACM, pp 24–31
9. Boring S, Baur D, Butz A., Gustafson S, Baudisch P (2010) Touch projector: Mobile interaction through video. In: Proceedings CHI 2010. ACM, pp 2287–2296
10. Bragdon A, DeLine R, Hinckley K, Morris MR (2011) Code space: Combining touch, devices, and skeletal tracking to support developer meetings. In: Proceedings ITS 2011. ACM, pp 212–221
11. Chung H, North C, Self JZ, Chu S, Quek F (2014) Visporter: facilitating information sharing for collaborative sensemaking on multiple displays. Pers Ubiquit Comput 18(5):1169–1186

12. Dachselt R, Buchholz R (2008) Throw and Tilt—seamless interaction across devices using mobile phone gestures. In: Proceedings Workshop MEIS 2008. German Informatics Society (GI), pp 272–278
13. Frisch M, Langner R, Dachselt R (2011) Neat: a set of flexible tools and gestures for layout tasks on interactive displays. In: Proceedings ITS 2011. ACM, pp 1–10
14. Greenberg S, Marquardt N, Ballendat T, Diaz-Marino R, Wang M (2011) Proxemic interactions: the new ubicomp? Interactions 18(1):42–50
15. Hardy R, Rukzio E (2008) Touch & interact: touch-based interaction of mobile phones with displays. In: Proceedings MobileHCI 2008. ACM, pp 245–254
16. Hassan N, Rahman MM, Irani P, Graham P (2009) Chucking: a one-handed document sharing technique. In: Proceedings INTERACT 2009. Springer, Berlin, Heidelberg, pp 264–278
17. Houben S, Marquardt N (2015) Watchconnect: a toolkit for prototyping smartwatch-centric cross-device applications. In: Proceedings CHI 2015. ACM, New York, NY, USA, pp 1247–1256
18. Jakobsen MR, Hornbæk K (2015) Is moving improving?: Some effects of locomotion in wall-display interaction. In: Proceedings CHI 2015. ACM, pp 4169–4178
19. Jokela T, Ojala J, Grassel G, Piippo P, Olsson T (2015) A comparison of methods to move visual objects between personal mobile devices in different contexts of use. In: Proceedings MobileHCI 2015. ACM, pp 172–181
20. Kabbash P, Buxton WAS (1995) The "prince" technique: Fitts' law and selection using area cursors. In: Proceedings CHI 1995. ACM Press/Addison-Wesley Publishing Co., pp 273–279
21. Keefe DF, Gupta A, Feldman D, Carlis JV, Keefe SK, Griffin TJ (2012) Scaling up multi-touch selection and querying: Interfaces and applications for combining mobile multi-touch input with large-scale visualization displays. Int J Hum Comput Stud 70(10):703–713
22. Kopper R, Bowman DA, Silva MG, McMahan RP (2010) A human motor behavior model for distal pointing tasks. Int J Hum Comput Stud 68(10):603–615
23. Lehmann A, Staadt O (2013) Distance-aware bimanual interaction for large high-resolution displays. In: Computer vision, imaging and computer graphics. Theory and application, communications in computer and information science, vol 359. Springer, Berlin, pp 97–111
24. Marquardt N, Ballendat T, Boring S, Greenberg S, Hinckley K (2012) Gradual engagement: facilitating information exchange between digital devices as a function of proximity. In: Proceedings ITS 2012. ACM, pp 31–40
25. Müller J, Alt F, Michelis D, Schmidt A (2010) Requirements and design space for interactive public displays. In: Proceedings MM 2010. ACM, pp 1285–1294
26. Myers BA, Bhatnagar R, Nichols J, Peck CH, Kong D, Miller R, Long AC (2002) Interacting at a distance: measuring the performance of laser pointers and other devices. In: Proceedings CHI 2002. ACM, pp 33–40
27. Nancel M, Chapuis O, Pietriga E, Yang XD, Irani PP, Beaudouin-Lafon M (2013) High-precision pointing on large wall displays using small handheld devices. In: Proceedings CHI 2013. ACM, pp 831–840
28. Nancel M, Pietriga E, Chapuis O, Beaudouin-Lafon M (2015) Mid-air pointing on ultra-walls. ACM Trans Comput Hum Interact 22(5), 21:1–21:62 (2015)
29. Norman DA (2002) The design of everyday things. Basic Books, Inc
30. Peck SM, North C, Bowman D (2009) A multiscale interaction technique for large, high-resolution displays. In: IEEE symposium on 3D user interfaces 2009. IEEE, pp 31–38
31. Pohl H, Murray-Smith R (2013) Focused and casual interactions: allowing users to vary their level of engagement. In: Proceedings CHI 2013. ACM, pp 2223–2232
32. Rädle R, Jetter HC, Butscher S, Reiterer H (2013) The effect of egocentric body movements on users' navigation performance and spatial memory in zoomable user interfaces. In: Proceedings ITS 2013. ACM, pp 23–32
33. Rashid U, Nacenta MA, Quigley A (2012) The cost of display switching: a comparison of mobile, large display and hybrid ui configurations. In: Proceedings AVI 2012. ACM, pp 99–106

34. Rekimoto J (1997) Pick-and-drop: a direct manipulation technique for multiple computer environments. In: Proceedings UIST 1997. ACM, pp 31–39
35. Schmidt D, Chehimi F, Rukzio E, Gellersen H (2010) Phonetouch: a technique for direct phone interaction on surfaces. In: Proceedings UIST 2010. ACM, pp 13–16
36. Schmidt D, Seifert J, Rukzio E, Gellersen H (2012) A cross-device interaction style for mobiles and surfaces. In: Proceedings DIS 2012. ACM, pp 318–327
37. Seifert J, Bayer A, Rukzio E (2013) Pointerphone: using mobile phones for direct pointing interactions with remote displays. In: Kotzé P, Marsden G, Lindgaard G, Wesson J, Winckler M (eds) Human-Computer Interaction—INTERACT 2013, vol 8119. Springer, Berlin, Heidelberg, pp 18–35
38. Seifert J, Dobbelstein D, Schmidt D, Holleis P, Rukzio E (2014) From the private into the public: privacy-respecting mobile interaction techniques for sharing data on surfaces. Pers Ubiquit Comput 18(4):1013–1026
39. Tan DS, Czerwinski M (2003) Effects of visual separation and physical discontinuities when distributing information across multiple displays. Proc INTERACT 2003:252–255
40. Vogel D, Balakrishnan R (2005) Distant freehand pointing and clicking on very large, high resolution displays. In: Proceedings UIST 2005. ACM, pp 33–42
41. von Zadow U, Büschel W, Langner R, Dachselt R (2014) Sleed: using a sleeve display to interact with touch-sensitive display walls. In: Proceedings ITS 2014. ACM, pp 129–138
42. Wagner J, Nancel M, Gustafson SG, Huot S, Mackay WE (2013) Body-centric design space for multi-surface interaction. In: Proceedings CHI 2013. ACM, pp 1299–1308

Chapter 6
Interactive Exploration of Three-Dimensional Scientific Visualizations on Large Display Surfaces

Tobias Isenberg

Abstract The chapter surveys the different approaches investigated to interact with scientific visualizations on large surfaces such as tables and walls. The chapter particularly does not focus on VR-based interaction or tangible input but on those interaction techniques where the input is provided on the surface itself or where it is focused on the surface. In particular, tactile interaction techniques are covered and the challenges of gestural input as well as of combining touch input with stereoscopic rendering are discussed. Where possible, connections to collaborative interaction scenarios are pointed out, even though most publications to date focus on single-user interaction.

6.1 Introduction

Scientific visualization of data which has an implicit mapping to the 3D Euclidean space has traditionally been a domain for which interaction plays an important role. For example, the interactive exploration of 3D medical data, physical or astrophysical simulations, or models from structural biology has always been important as soon as the underlying graphics hardware had become powerful enough to support such interactive exploration. Initially, this interaction typically concentrated on navigation of 3D environments or the manipulation (translation, rotation, scaling) of parts of the visualization. Recently, however, researchers have started to focus on more flexible interaction techniques that facilitate advanced exploration of scientific datasets [51]. A large part of this work has explored surface-based interaction in which (one of) the data display(s) also serves as the main space where input is provided by the interacting person or people—a topic that has received an increasing amount of attention in the recent years for visualization in general [38, 39] and specifically for the exploration of 3D data [52].

T. Isenberg (✉)
Inria, Saclay, France
e-mail: tobias.isenberg@inria.fr

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_6

Naturally, the use of interactive surfaces inherently supports the collaboration of people, in contrast to the single-person input of traditional interaction settings based on mouse, keyboard, or dedicated 3D input devices. While collaborative interaction is the core topic of this book, in 3D data visualization has the additional constraint that the data domain typically needs to use a single, linear mapping to a shared display surface so that a navigation of the 3D data representation, at least, can only be done by one person at a time. In this chapter we thus focus on single-user interaction techniques and point out the few approaches that have explored collaborative scenarios.

In addition, we focus on those interaction techniques that use the 2D input provided on interactive surfaces for interaction, but mention some special approaches that use both 2D input and immersive displays in our discussion. While there have been approaches for other forms of input, for example, through wands, gloves, or 3D tracking for use in immersive VR environments [26, 28, 57] as well as fully tangible interaction (e.g., [58]), our focus on directly capturing the input on an interactive surface has several advantages. First, the provided input is direct as the input location can directly correspond to a displayed data element, in contrast to wands and remote pointing devices as well as passive tangible props. Second, input on interactive surfaces does not require elaborate additional hardware (as for gloves) or 3D tracking setups (as in VR and tangible interaction), making the management and use of such data exploration systems easier and less expensive. Third, the properties and advantages of tactile interaction known from other forms of interaction [17] similarly apply, such as improved performance of tactile input [54], the support of awareness of collaborators [37], somesthetic information and feedback [67], and improved performance on physically large displays [77]. Moreover, recently it has been shown that certain interaction techniques such as based on tactile input on large displays can serve as a communication channel when presenting visualizations to an audience [76].

In the remainder of this chapter we first discuss the issue of the difference between the data space on the one side and input and output spaces on the other side. Next, we review basic interaction techniques for surface-based interaction with 3D data. Then we introduce a number of design studies for data exploration systems from various domains, before we conclude the chapter with a summary.[1]

6.2 Data Space Versus Input and Output Spaces

The restriction to (typically planar) 2D input spaces for the control or exploration of 3D data spaces brings with it an important mapping problem: the need for mapping from the 2D input space to the 3D data space. In addition to this mapping, the

[1] It is difficult to understand interaction techniques by simply reading about them or seeing traditional figures with snapshots. We thus hyperlink to videos of the discussed techniques where possible from the figures in the electronic version of the chapter to better illustrate the techniques.

normal visualization mappings, of course, also take place. In contrast to the mappings needed for the visualization of abstract, non-spatial data [18], for the visualization of spatial 3D data we already have an implicit mapping so that most mappings in the visualization pipeline concern data filtering, the assignment of abstract aspects to visual variables, and 3D projection and rendering (e.g., [79, Section 4.1]). Essential for interactive visualization in general and our specific case of 2D surface-based input, however, is that we also need to take the physical presentation of the generated visualization [47] into account.

6.2.1 2D Projected Viewing of 3D Data Visualizations

Unlike in immersive VR settings, much 3D visualization relies on the rendered visualization being projected to 2D and displayed on a "normal" screen. Typically, this screen is the display on which input is captured. In this case the input is co-located to the visual representation of the output, and the same ***mental mapping*** from visualization space to data space and vice versa can also be applied to the provided input (Fig. 6.1). In that sense the input is as direct as possible as the user never encounters a visual representation that has the same dimensionality as the data. If the input "display" is separate from the display that shows the projected visualization, then we have a situation similar to touch pad interaction or the use of digitizer tablets where, while the dimensionality of the displayed visualization and the provided input is the same, a ***mental mapping*** from input space to output space and from that to the data space is necessary, which makes this indirect interaction more difficult.

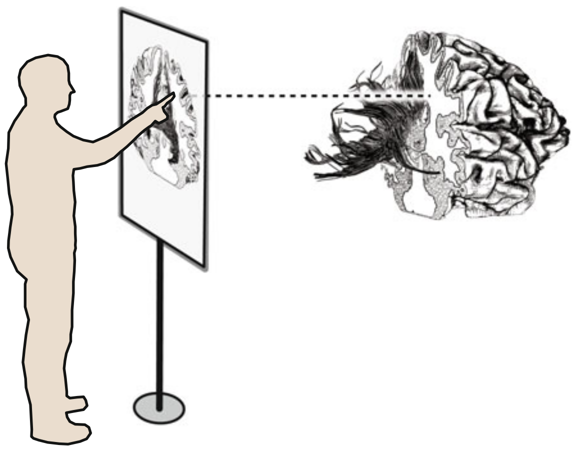


Fig. 6.1 Surface-based 2D interaction with 3D data: a mental mapping from the input and the projected visualization to the 3D data space is required. Image © Tobias Isenberg, used by permission

6.2.2 3D Stereoscopic Viewing of 3D Data Visualizations

In contrast to a projected display of the visualization one may also want to take advantage of the better depth perception and the resulting increased feeling of immersion of a stereoscopic presentation of the visualization. The use of surface-based input in such immersive virtual reality environments, however, presents additional challenges [82]. Here, the dimensionality of the presented visualization is the same as that of the data—the data is perceived at the same location as that of the data (with the exception of 3D manipulations of the visual representations that can also be understood as manipulations that are applied to the data itself).

The input, however, still is provided on a 2D surface to take advantage of the benefits mentioned above. This means that only in rare cases is the input actually performed where the user perceives the data to be manipulated. Moreover, it has been show that such tactile (or pen-based) interaction suffers from the parallax between the two images that are shown for both of the eyes [13, 14, 24, 82, 83]. In addition, touch-through [19, 78, 81] and invisible wall problems make such an interaction setup problematic. Only in situations when the element to be accessed by the surface-based input is at a close distance to input screen do users perceive their input to directly control the manipulated elements [83].

Some solutions exist to address these problems, yet none are ideal. For example, Schmalstieg et al. [69] suggested to use transparent props, yet these are static and would not work well with 3D data space manipulations and time-dependent data. Hachet et al. [31] separated the touch surface from the stereoscopic display in their Toucheo system, but thus significantly restricted the space in which people can interact. Jackson et al. [45] use the touch surface of a table interface as the interaction reference frame on which widgets are placed, and input is provided not only through tactile sensing but also through over-the-surface means supported by 3D tracking (see Sect. 6.3.1 for a more detailed description of this technique). Butkiewicz and Ware [15, 16], finally, used a very specific setup that relies on a tilted setup, shallow-depth data, and a single “natural” interaction surface (see the more detailed description of this setup in Sect. 6.4).

In addition to these hardware solutions, also some software-based interaction designs were proposed to alleviate the parallax problem. For example, Valkov et al. [81] suggested to move objects toward the touch surface as a user reaches out to them. Giesler et al. [30] used “void shadows” that connect the objects behind a touch surface with it and which serve as interaction proxies. These interaction designs, however, may cause problems in a visualization environment as the data display itself should not be obscured and often no dedicated objects exist. As an alternative, people thus also examined hybrid settings that separate the stereoscopic display from the input surface as discussed next.

6.2.3 Hybrid View Settings for 3D Data Visualization

In normal PC-based settings we are quite used to having the surface that displays the data or object with which we interact (i.e., the computer screen) to be different from the surface on which we provide input (i.e., the table on which the keyboard and mouse are located). Humans are able to deal with such separation and are ready to make a mental mapping from one surface to the other if the mapping only contains translations [8, 10, 85] and simple rotations [1, 2, 25], also in immersive environments [84]. The same concept has also been used for the surface-based interaction with stereoscopic displays. For example, Coffey et al. [20–22] use a vertical display to show the stereoscopic content of a visualization, while capturing tactile input on a horizontal surface (see Fig. 6.2). Both surfaces are physically connected perpendicularly, and are visually connected to each through a stereoscopic world-in-miniature (WIM) display of the data as well as shadows that this WIM casts on the tactile input surface. Similar static hybrid setups at interactive 3D visualization have been used by Bogorin et al. [11] and Novotný et al. [64].

In addition to such static setups, modern smart phones and tablet computers also facilitate interaction styles where the input is provided on a mobile surface [72], while the data is still visualized stereoscopically. This scenario, however, poses additional challenges as the mapping from the input provided and data displayed on the mobile surface to the stationary (and typically large) stereoscopic surface

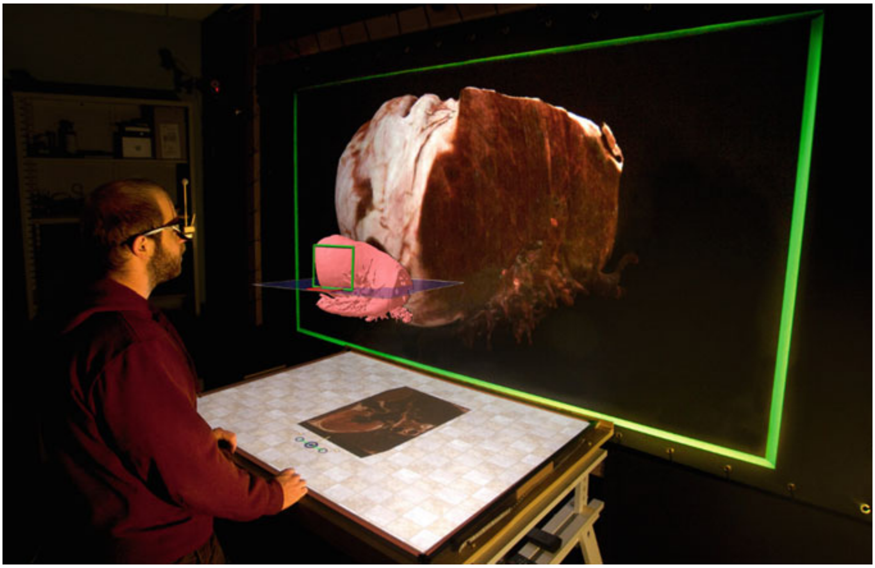


Fig. 6.2 Static mapping between input surface and stereoscopic data display in Coffey et al.'s [21, 22] Slice WIM setup. Image courtesy of and © Daniel F. Keefe, used by permission

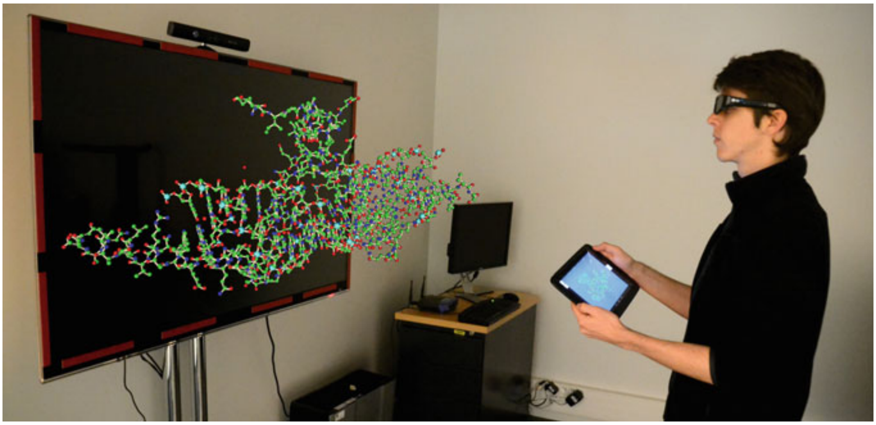


Fig. 6.3 Stereoscopic data exploration with mobile control. Image from [60], © IEEE, used by permission

constantly changes as the interacting person is moving around. López et al. [60] recently analyzed the interaction modes in this situation (e.g., Fig. 6.3). In particular, they described the mapping issues between the different frames of reference that arise when the interacting person moves beyond a small distance around their initial position—due to the issues humans have with such mappings if more than simple translations and a single rotation is involved as described above. Based on this analysis they suggested an data exploration workflow that allows users to move in the physical space and transition between the different interaction modes, synchronizing the two views when the reference frames can no longer be easily mentally integrated.

6.3 Basic Interaction Techniques

Based on these viewing and interaction settings we can now turn to the specific interaction techniques used for surface-based 3D data exploration and analysis. In doing this, we concentrate on those techniques that use planar, monoscopic 2D surfaces as input spaces because most of the special cases mentioned in Sect. 6.2.2 do not play a role in 3D visualization applications so far. In this chapter we first describe basing interaction techniques that are common to many of not most 3D visualization applications, in particular those for 3D navigation and for data selection. Next, in Sect. 6.4, we then discuss a number of systems and design studies that combine several interaction techniques for a more comprehensive interaction with and exploration of data.

6.3.1 Data Navigation

Navigation techniques for 3D environments have been investigated for a long time [12, 36]. Also surface-based 3D interaction techniques are not only a domain of visualization [46]. Here we mention a number of techniques that are, in principle applicable to interactive 3D visualization, even if the techniques were not initially designed with this application in mind. However, in general 3D interaction the focus often is placed on the ***manipulation of individual 3D objects*** within a larger space, such as moving furniture items around in a virtual environment that shows a new interior design. In 3D visualization, however, we rarely manipulate individual objects but rather navigate in the 3D data space to look at specific aspects of the data more closely. Nevertheless, many generic surface-based 3D navigation techniques can be used in visualization by using them to ***affect the "data space"*** of the visualization.[2]

One of these interaction techniques for general 3D shapes is the 3D-RST approach by Reisman et al. [66] that is inspired by the common two-finger pinching interaction. RST stands for rotation, scaling, and translation and, in the 2D case, allows users to perform these transformations for 2D shapes in their native 2D space [35]. This interaction relies on the principle that the interaction points are "sticky" [33, 66], i.e., that they stay connected to the same location on the manipulated object for the entire interaction. Reisman et al.'s [66] 3D-RST[3] extends this general principle to the reorientation and translation of 3D shapes, using the 2D screen-space locations of multiple interaction points to constrain the mapping. Of course, this mapping is easily either under-constrained or over-constrained:

- one to two fingers provide only ≤ 4 DOF, while 6 DOF are needed to specify the location and orientation of a 3D shape, while
- four or more fingers provide ≥ 8 DOF for the same necessary 6 DOF.

Reisman et al. solve this problem in screen-space using an energy minimization approach the find the best possible mapping despite the possibly under- or over-constrained input (Fig. 6.4). They demonstrate how their technique can be applied to many types of surfaces including terrain renderings, and it is not difficult to envision to apply the same technique to other planar elements in visualization applications such as cutting planes.

An alternative interaction mapping was designed by Liu et al. [59] who integrate both the 4DOF x-/y-/z-translation plus z-rotation with the 2DOF x-/y-rotation. Users can seamlessly switch between these two modalities by either moving both fingers at

[2]In interactive 3D visualization there may indeed be some dedicated objects to be moved such as cutting planes and particle sources. Nevertheless, for such objects often dedicated interaction techniques are used as explained in the remainder of the chapter.

[3]3D-RST is a somewhat inappropriate name as Reisman et al.'s [66] technique is constrained to translations and rotations. The scale always remains constant with this technique. In fact, a technique that is entirely based on "sticky" contact control cannot affect both z-distance and object scale at the same time, the two properties are visually ambiguous (see also Hancock et al.'s [33] "Sticky Tools" interaction mapping and its application to Sandtray therapy [34]).

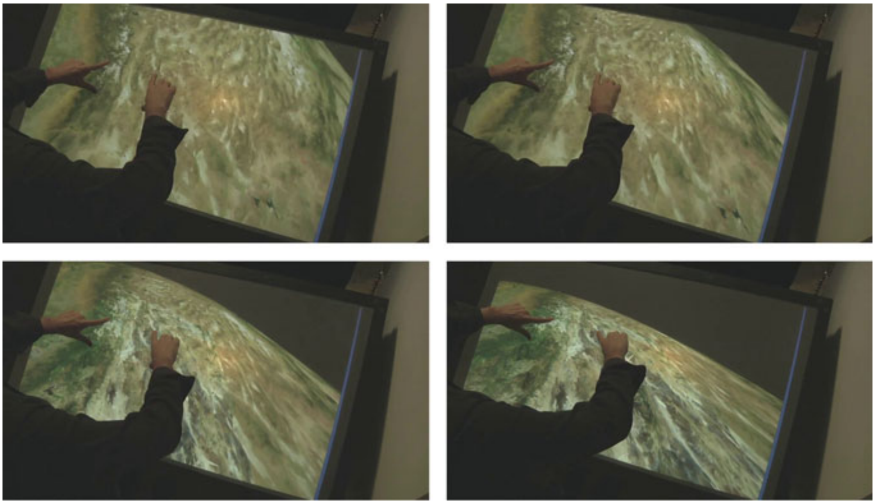


Fig. 6.4 Interaction sequence from the control of the orientation and location of a map orientation in 3D space using 3D-RST [66]. Images courtesy of and © Jason L. Reisman, Philip L. Davidson, and Jefferson Y. Han, used by permission

the same time or by leaving one finger static on the tactile surface. Similar to Reisman et al.'s [66] approach, this technique allows Liu et al. [59] to facilitate flexible and fluid interactions with 3D shapes.

While Reisman et al.'s [66] 3D-RST and Liu et al.'s [59] two-finger technique facilitate a flexible and fluid type of interaction, it always affects all 6 DOF (for Reisman et al.'s [66] technique) or 2 DOF resp. 4 DOF (for Liu et al.'s [59] approach) of the output simultaneously. In visualization applications, however, it is often necessary to single out specific DOF for the interaction to be able to constrain which aspects of a visualization are affected.

To address this problem, Au et al. [6] describe a set of gestures to single out specific DOFs to control individually. One problem with such a gestural approach is that the gesture set has to be learned and is not easily discoverable. Cohe et al. [23] describe a similar constrained interaction with their tBox technique (Fig. 6.5) for up to 9 DOF control. This widget-based approach shows a box-shaped interaction widget overlaid on the rendering, whose orientation is tied to that of the shown 3D scene or object. Manipulations can now been applied based on where on the widget input is provided. For example, single inputs on the cube's sides provide single-axis rotations, while single inputs on a cube edge start translations along the axis parallel to the edge. Uniform scaling is possible using pinching on a cube side, non-uniform scaling by pinching on two opposite cube edges. These interactions allow users to constrain their manipulations to only single-DOF control, and studies [60] indicate that the tBox provides people with an increased feeling of precision for the 3D interaction.

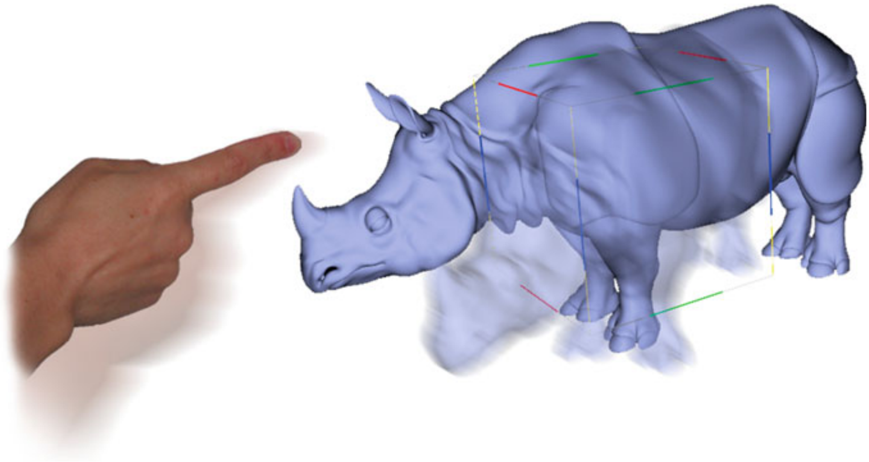


Fig. 6.5 Illustration of the interaction with the tBox [23] which is shown overlaid on a 3D object. Image courtesy of and © Martin Hachet, used by permission

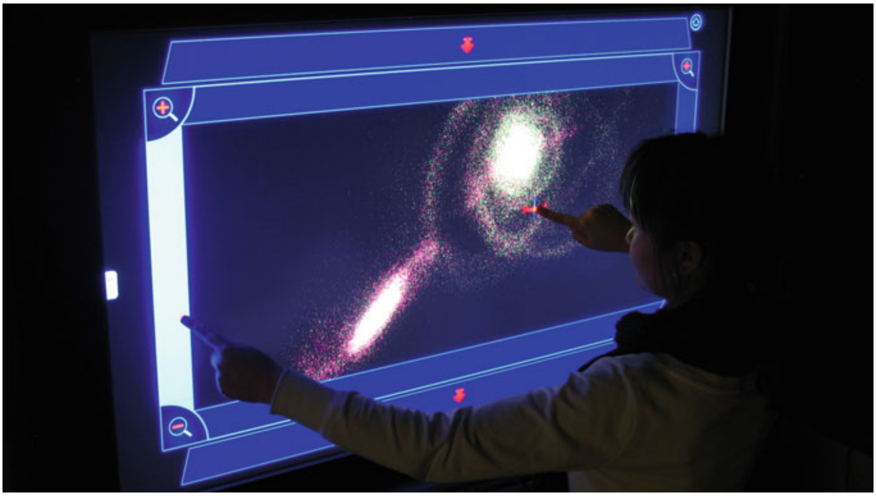
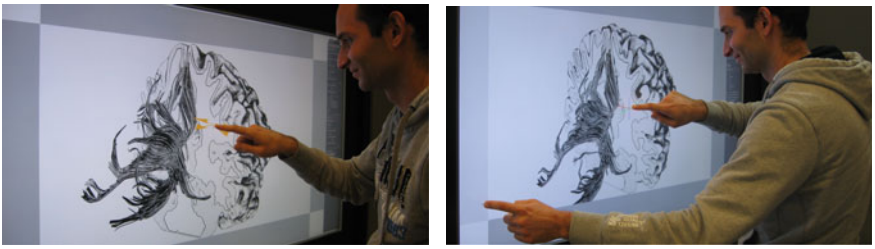


Fig. 6.6 Widget-based 3D navigation using FI3D, using different mappings on the man data view and on the FI3D frame. Image from [90], © IEEE, used by permission

While such precise control is essential, a flexible and fluid 3D navigation may also be important. Unfortunately, Reisman et al.'s [66] 3D-RST which provides such flexibility only facilitates 6 DOF control. So Yu et al. [90] conceived FI3D (Fig. 6.6), a widget-based 3D navigation approach that allows researchers to control up to 7 DOF (3D rotations, 3D translations, and uniform scaling). The approach is based on a widget placed around a central data display, with which controls the interaction mode based on the location of where an input starts as well as its direction. In the the center, Yu et al. map x/y-translation as well as 2D RST manipulation. Interactions started in the frame, initially moving along it start rotations around the z-axis, while interactions from the frame initially into the central data view start trackball rotations. Separate regions are used for uniform scaling and z-translations, and bi-manual techniques allow users to constrain their rotation input also to the x- and y-axis as well as allow to provide different 2D rotation centers.

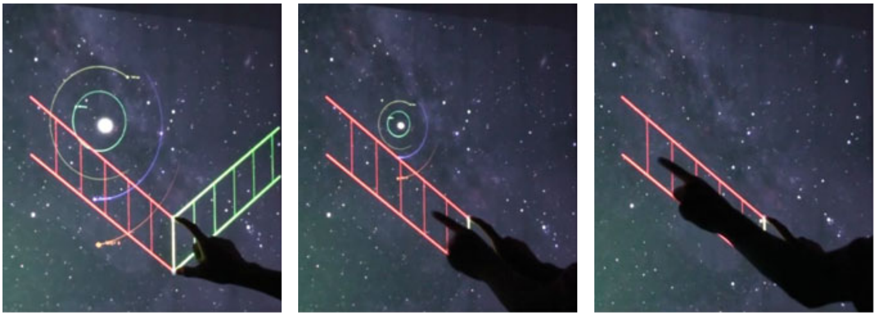
Yu et al. [90] also mention that the specific mappings should depend on the data being shown. Their initial mapping works well for data such as astro-physical particle simulations that does not have an inherent center point and that require a lot of scale changes to explore different aspects—in such cases translations are more important than rotations and, hence, are mapped to one-point input in the central view. Other types of data such as brain scans, however, may have an inherent center and may not require much scaling—in such cases rotations are more important than translations. Yu et al. thus also demonstrate that the two mappings can be flipped (Fig. 6.7a), and that additional functions such as cutting plane manipulations and fibertract selection can be realized using bimanual interaction (Fig. 6.7b).

In addition to these generic navigation techniques, some techniques with domain-specific constraints have also been created. In the context of visualization, two should be mentioned at this point. The first one specifically supports the navigation of 3D astronomical datasets such as models of the solar system, its local neighborhood, the Milky Way, and the spatial arrangement of multiple galaxies. Such a setup has two major constraints: it (a) primarily requires rotations (as opposed to translations) and it (b) requires scaling across multiple orders of magnitude. For this purpose, Fu et al. [29] provide a set of interaction techniques that combine spherical navigation based on a trackball metaphor as well as their unique "powers-of-ten-ladder" for multi-scale navigation (Fig. 6.8). The latter is invoked with two fingers touching the interaction surface in a vertical arrangement, and then a second hand can intiate scale



(a) Rotation-centered navigation mapping. (b) Bimanual cutting plane manipulation.

Fig. 6.7 Aspects of the FI3D-based interaction mapping can and should depend on the data. For data that does not require much zooming, the main interaction is trackball rotation and is thus used for the one-finger mapping, while x/y-translation is accessed from the FI3D frame (**a**). In addition, dedicated interaction techniques such as cutting plane manipulation are supported (**b**). Images from [90], © IEEE, used by permission



(a) Invocation. (b) Small-scale control. (c) Large-scale control.

Fig. 6.8 Powers-of-10-Ladder [29]. Depending on where input is provided with respect to the basis of the interaction widget, either **b** small-scale or **c** large-scale changes to the zoom level are being introduced. Images courtesy of and © Chi-Wing Fu, used by permission

navigation, either initiating small-scale changes (Fig. 6.8b) or large-scale changes (Fig. 6.8c)—depending on the distance of the input to the basis of the widget.

Another domain-specific tactile surface-based navigation technique was introduced by Sultanum et al. [75] for the exploration of geological outcrops. Such 3D datasets are captured, for example, by LiDAR scans and reveal information on geological layers as part of geological surveys. To provide 3D navigation with such outcrops, Sultanum et al. facilitate a first-person fly-by navigation strategy by defining a separate navigation surface to which the camera is constrained. The scientist can then explore the dataset though tactile interaction gestures, controlling the remaining camera parameters such as panning its *x*-/*y*-location, zooming, and tilting.

While all previous techniques restrict themselves to controlling the 3D scene or objects solely based on 2D input captured on the tactile surface, Jackson et al. [45] augment this 2D input with additional information based on the posture of the interacting hands (Fig. 6.9). Their "nailing down multi-touch" set of interaction techniques thus allows users to tilt, bend, or twist objects or datasets within the 3D space, supported by a stereoscopic data display. As they specifically treat the interaction surface as the location where the interaction control widgets are placed (as can be seen in Fig. 6.9), this interaction style does not cause many problems despite the previously discussed issues of tactile interaction with stereoscopically displayed scenes (Sect. 6.2.2).

In addition to the navigation techniques we discussed so far, several other 3D interaction techniques for surface-based 3D navigation have been designed (see, e.g., [12, 36, 46]). Most of them, however, rely on the manipulation of individual 3D objects within the 3D space in a way that is not very well suitable for 3D interaction with data visualizations [40].

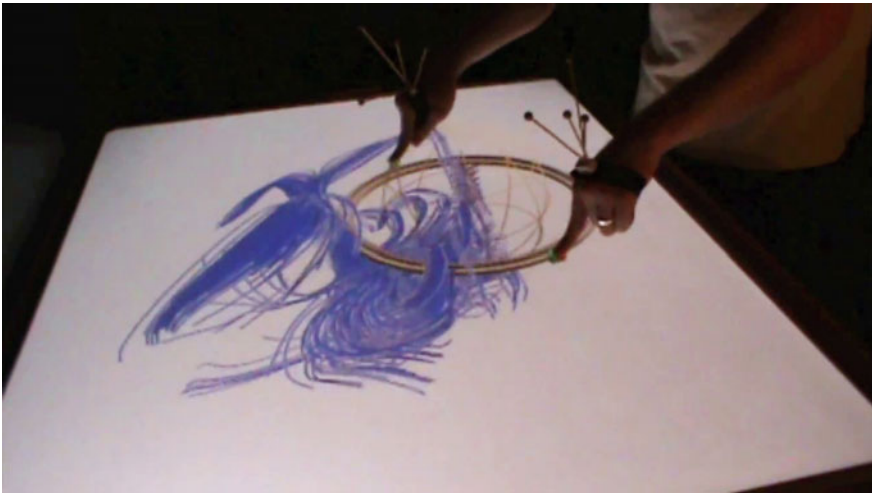


Fig. 6.9 Nailing down multi-touch interaction [45] for providing additional tilting, bending, or twisting of the 3D data. Image courtesy of and © Daniel F. Keefe, used by permission

6.3.2 Data Picking and Data Selection

In addition to 3D navigation, a second interaction technique that is essential for the exploration of 3D data visualizations is the selection of sub-elements of the depicted datasets. While selection techniques for 2D [87] and 3D [4, 5, 7] environments have long been studied, the picking or selection within 3D datasets is not as straightforward as one may think. A first challenge lies in the problem that either no explicit objects make up the dataset (for example, in sampled data of a continuous domain such as volumetric datasets) or that the explicit data objects are too small or narrow to be easily captured by traditional picking or selection techniques (such as in point-based line-based datasets). A second issue arises from the fact that interaction can only be recorded on the two-dimensional input surface, so this input is not able to fully constrain the intended three-dimensional selection.

The ultimate challenge in 3D data selection is thus to effectively and intuitively specify that sub-space of the dataset that contains the elements to be further processed. While data filtering is one approach to arrive at such selections sub-spaces, the characteristics of the intended selections may not be known ahead of time or there may not even be data aspects that would allow such an effective filtering during exploratory data analysis. Below we review a number of spatial input techniques that specify intended selections based on spatial input. While none of the techniques we review in this section were specifically created for the application to surface-based interaction, they all work particularly in surface-based interaction contexts due to their spatial input character and direct manipulation paradigms they support: spatial selection input can thus be directly specified with respect to the displayed data.

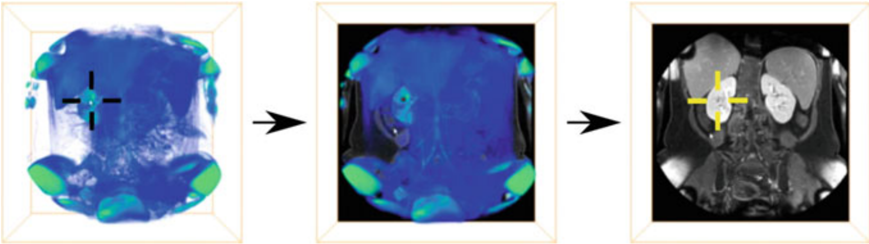


Fig. 6.10 What you see is what you pick interaction [86]. Once a location is picked in the projected view of a volumetric visualization (*left*), its 3D location is deducted and a cutting plane can be placed accordingly, facilitating the exploration of this cutting plane in the context of the volume rendering (*center*) or by itself (*right*). Image courtesy of and © Alexander Wiebel, used by permission

A fundamental interaction technique in this context is picking. While the picking of individual objects is simple using ray-pointing and similar techniques, picking in continuous data such as medical volume scans or physical simulations is far from straight-forward. For this purpose Wiebel et al. [86] created their What You See Is What You Pick interaction technique (Fig. 6.10), a structure- and view-aware picking approach that takes the data along the picking ray as well as the transfer function into account. Specifically, they extract the section along the ray that constitutes the larges jump in accumulated opacity, corresponding to the feature that is visually dominant at the picked 2D position. For this section they then select either its front or center, depending on the user's preference.

While this structure- and view-aware picking technique can only yield a single 3D position within the 3D data, it is sometimes also necessary to select a whole 3D subspace, such as to be able to do carry out a specific data analysis of visually interesting features. For this purpose several structure- and view-aware spatial selection techniques have been created. Based on the initial work by Owada et al. [65], Yu et al. [88] introduced their CloudLasso that bases the selection on an interactively drawn 2D selection lasso and the analysis of a scalar value such as density based on which a selection volume is extrapolated (Fig. 6.11 shows the use of the CloudLasso selection in a surface-based interaction context).

Structure-aware and view-dependent selection techniques such as CloudLasso, however, have the disadvantage that they (a) select everything along the line where the chosen scalar threshold is surpassed, which can lead to multiple selected but unconnected components. Moreover, they (b) do not the shape of drawn lasso itself into account other than to use it as a 2D cut-off constraint for the 3D selection. To address bot issues, Yu et al. [89] extended their initial approach an described the CAST family of context-aware selection techniques (Fig. 6.12). In particular, they describe SpaceCAST (Fig. 6.12a), that works similar to CloudLasso but selects that connected component whose outline is most similar to the drawn lasso. Next, they created TraceCast (Fig. 6.12b) that relaxes the constraint that the lasso cuts off the selection with respect to the 2D view, facilitating the easy selection of complex structures. Finally, they describe PointCAST (Fig. 6.12c) which only requires a point as

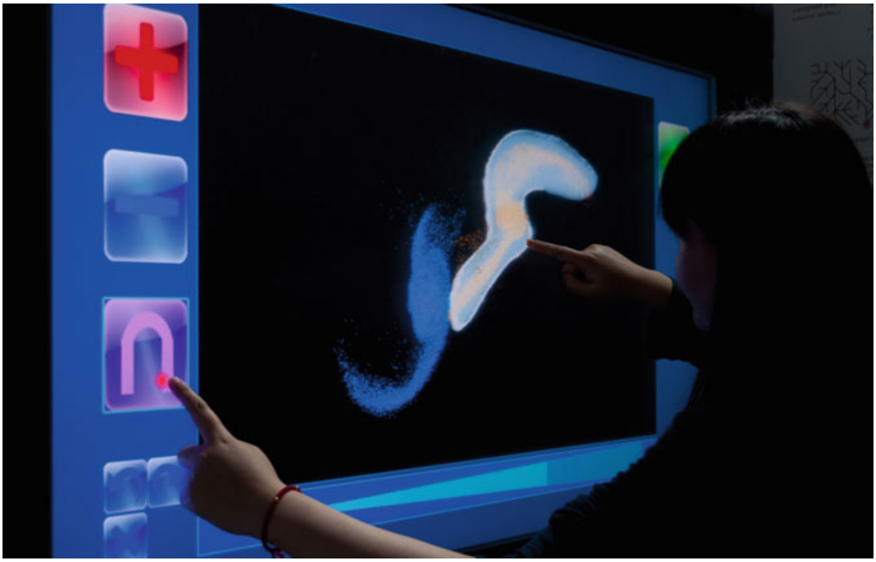
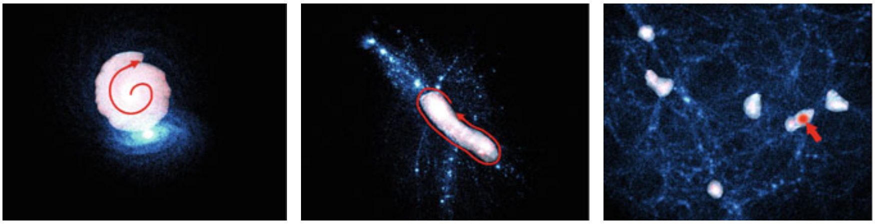


Fig. 6.11 Bimanual CloudLasso selection within an astronomical particle dataset. Image from [88], © IEEE, used by permission



(a) SpaceCAST selection. (b) TraceCast selection. (c) PointCAST selection.

Fig. 6.12 CAST selection for 3D particle data which can be used in a similar surface-based environment as shown in Fig. 6.11. Image from [89], © IEEE, used by permission

an input to facilitate the selection of small clusters. An alternative is to pick that cluster from the selection that has the largest 2D projection [70] but this approach does not take the actual shape of the selection lasso into account.

The approaches discussed so far work well for point-based or scalar 3D data (i.e., particle clouds and volumetric data), but other 3D data types require different approaches. In particular, line based data such as streamlines and similar or fiber tracts are too long for it to be possible to effectively select subsets with a volume-based technique. While it is possible to use dedicated input hardware for a spatial selection of line-based data (e.g., [43, 44, 53]), solely surface-based spatial selection techniques are less common and we are only aware of two approaches. Akers' [3] fibertract selection uses the shape of a drawn selection mark to guide the selection of 3D neurologic pathways in a structure-aware fashion, while Coffey et al.'s [22] Slice WIM widget facilitates the selection of flowline bundles by drawing a selection lasso on a plane that was previously placed roughly perpendicular to the flow. Tong et al.'s [80] interaction techniques, in contrast, are not used for streamline selection but allow users to specify spatial lenses though tactile input that then reveal hidden parts of a streamline dataset.

6.3.3 Summary

The discussed basic navigation and selection techniques demonstrate that 3D navigation and selection can effectively be carried out also in a surface-based interaction context. While other interaction mappings for these data exploration tasks will certainly be explored in the future, the existing ones already provide a good selection for the practical implementation of surface-based visualization tools. In the survey we intentionally did not cover, however, techniques for the manipulation of 3D data elements because data visualization of scanned or simulated data typically does not require the *manipulation of the data*, but focuses on the *exploration of the data*. Yet, many additional interaction tasks for data exploration also have to be supported [52] such as particle seeding at 3D locations, cutting plane manipulation, path planning, data value probing, and visualization parameter adjustment. Some of these have already been explored for surface-based interaction settings, such as the visual exploration of different settings for volume rendering transfer functions [50]. Most of them, however, have been explored within the context of a specific application domain or design study. We thus review several of such existing surface-based 3D data visualization systems[4] next.

6.4 Systems and Design Studies

Surface-based data exploration systems have been created for a variety of target audiences including museum visitors [32], scientific researchers in domains such as fluid mechanics and oceanography, medical doctors and researchers, and exploration geologists. As we focus in this chapter on the interaction techniques, we loosely group them by similar interaction characteristics, rather than chronologically, by the mentioned application domains, or intended target audiences.

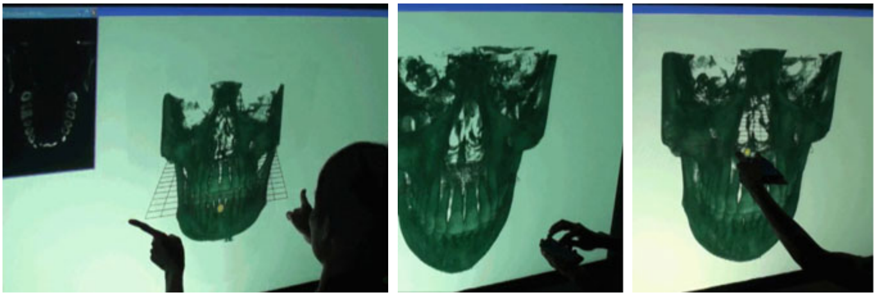
Marton et al. [62] describe IsoCam (Fig. 6.13), a touch-based system for the exploration of 3D scans of archeological artifacts in a museum context. As the target audience for this interactive system is almost exclusively untrained in 3D navigation, they use a constrained navigation that provides a robust exploration of large 3D

[4]The classification into interaction technique and interactive system is not always crystal clear—we used our best judgment to differentiate between the two groups.

Fig. 6.13 IsoCam interaction with massive cultural heritage models [62] at the Digital Mont'e Prama installation by CRS4 Visual Computing at the National Archaeological Museum of Cagliari, Italy. Image courtesy of and © Alberto Jaspe Villanueva, used by permission

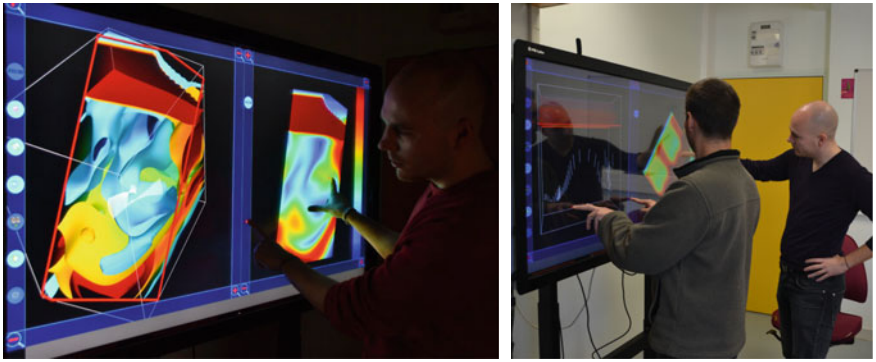
virtual reconstructions. They use an indirect navigation approach [56, 63]: gestural input on the touch surface is mapped to changes in the visualization that is shown on a remote screen. Specifically, they provide 2D navigation along an iso-surface of the distance field of the depicted objects (related to Sultanum et al.'s [75] work), zooming to change the distance to the object (i.e., the iso-value), and twisting to change the camera orientation. In addition, additional information can be accessed about the objects on demand. The interesting aspects about this interaction system is that it was deployed into the real world, with a non-expert target audience. The way Marton et al. thus constrained the degrees of freedom for navigation can thus be an inspiration for future interactive systems "for the masses."

Song et al. [71] combine a large monoscopic touch wall with a mobile secondary touch and orientation input device for the visualization of volumetric data (Fig. 6.14). They explore a combination of direct interaction on the large surface with manipulations of a cutting plane by means of the mobile device. They explore a number of combinations of tangible and tactile interaction techniques on the remote device to enable users to translate, reorient, zoom, and annotate the remotely shown visualization. The interesting aspect of this design study is the combination of tangible interaction with tactile input to create a larger interaction vocabulary including constrained interactions and alternative mappings for the same interaction. The interaction with the large display then facilitates the interaction with the data space itself, similar to some of those described in Sect. 6.3.1. A small user study conducted by Song et al. suggests that their hybrid interaction can outperform a PC-based interface with equivalent controls with respect to interaction time.



(a) General setup and tactile input. (b) Tangible input. (c) Hybrid tactile/tangible input.

Fig. 6.14 Song et al.'s [71] system that combines tactile and tangible interaction on for surface-based medical visualization. Images courtesy of and © Peng Song, Wooi Boon Goh, and Chi-Wing Fu, used by permission



(a) Single-user interaction. (b) Collaborative setting.

Fig. 6.15 Design study of a surface-based system for the exploration of fluid flow simulations [55]. Image (**a**) from [52], © IEEE, used by permission. Image (**b**) from [55], © Eurographics Association, used by permission

Klein et al. [55], instead, concentrate on capturing input only on a single large display, but design their system for the exploration of fluid simulations to provide similar interaction techniques (Fig. 6.15): the navigation of the overall dataset, cutting plane manipulation, data value probing, particle seeding, visualization parameterization, and temporal exploration. Their interaction design is largely based on the FI3D widget [90] but adds additional interaction mappings for manipulating the other mention elements, typically in a widget-based bimanual fashion. The interesting aspect of this system is its dual use of the cutting plane: it is not only used in the normal sense to cut of parts of the volumetric visualization but also is shown in an unprojected and undistorted way to provide a 2D input space to specify locations or regions for seed point placement. Moreover, the systems was designed to allow up to two people to interact at the same time due to the split-screen setup (Fig. 6.15b). Klein et al. also carried out a qualitative evaluation with domain experts from fluid mechanics which showed that the collaboration aspect of their design was liked by most participants, and that despite the use of a vertical display this collaboration worked well—to some degree contradicting previous work on the subject [68]. In addition, the evaluation also revealed the need of precise, constrained, and/or tightly controlled interactions, in addition to the fluid and flexible navigation techniques implemented in the system.

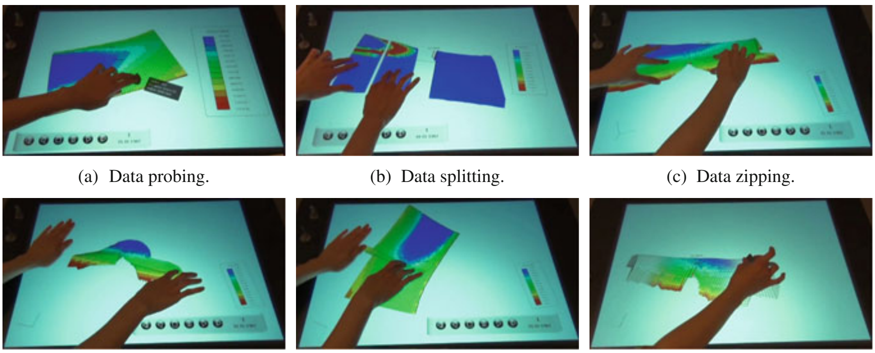
The systems discussed so far relied on vertical displays of the data—as it is common in many visualizations of spatial 3D datasets. Some types of data, however, inherently favor a horizontal mapping, such as surgery-based medical visualizations. Lundström et al. [61] thus use such a horizontal setup to create their virtual surgery table (Fig. 6.16). Similar to the setup described before, the virtual surgery table also was designed with collaboration in mind, but in this case with collaborators located around the table's horizontal surface (Fig. 6.16b). To specifically support this collaboration, they introduce "movable alternator pucks" which allow doctors to switch between the different interaction modalities in the system in a user-controlled fashion. For the main navigation interactions they use a typical 6 DOF one- and two-finger mapping that supports *x*/*y*-panning, *x*/*y*/*z*-rotation, and uniform zoom. Lundström et al. also conducted an observational study with domain experts (five medical doctors). This study provided numerous insights on the usability of the design, its clinical usefulness, and needed additional features for the system to be used in practice. In particular, the study demonstrated that a system such as the virtual surgery table is particularly useful for the analysis of complex cases, an insight that may also be possible to extend to other application domains such as data analysis by scientists in other domains. The participants also reported that a pure 3D interaction is not sufficient—the possibility to view and interact with additional 2D views such as traditional slices is needed. In the meantime, the research on the virtual surgery

(a) Single-user interaction. (b) Collaborative setting.

Fig. 6.16 The virtual surgery table [61], a horizontal interactive surface for the analysis of medical datasets. Image courtesy of and © Sectra, used by permission table has lead to the founding of a company (Sectra) which has continued to develop the system into a product that is now actively being marketed (as it is evident in the pictures shown in Fig. 6.16), and other companies (e.g., Anatomage) are offering similar setups.

The horizontal form factor has also been used by Sultanum et al. [73, 74] for their system to support the analysis of geologic reservoir data (Fig. 6.17). Their system is based on volumetric datasets that capture geological features such as seismic data, different surface layers, permeability levels, oil saturation levels, etc. Sultanum et al.'s system then allows geologists to explore these different aspects of the model, both by looking at the different data attributes (e.g., using physical property cards [73]—similar to Lundström et al.'s [61] virtual alternator pucks) as well as by manipulating the volumetric model itself. The latter is dony by interactions such as splitting (Fig. 6.17b), zipping it (Fig. 6.17c, d), or layer peeling (Fig. 6.17e). The interaction by means of tangible objects also facilitates data readout (Fig. 6.17a) and focus+context visualizations (Fig. 6.17f). One particularly interesting aspect of this interaction design is the large set of interaction techniques that are mapped in a non-conflicting way, enabling both navigation/view correction as well as several dataset manipulations using a coherent interaction design. Sultanum et al. not only base their work on observational sessions with the target users [73] but also conducted an evaluation of their final design [74] with domain experts. This last study revealed that, while the participants liked the overall system design with its flexible and fluid data exploration, they too asked for specific precise views such as 2D projections, similar to what was reported by Klein et al. [55] and Lundström et al. [61].

The systems discussed so far rely on the interaction with 2D projections of the 3D data visualizations—largely due to the interaction problems that arise from the combination of touch input and stereoscopic projection (as discussed in Sect. 6.2.2). We are only aware of two systems that use interaction setups that include stereoscopic projections, the hybrid Slice WIM setup by Coffey et al. [21, 22] for medical



(a) Data probing. (b) Data splitting. (c) Data zipping.

(d) Data zipping, one finger. (e) Layer flipping/peeling. (f) Focus+context.

Fig. 6.17 Data exploration techniques for reservoir visualization [73, 74]. Images courtesy of and © Nicole Sultanum, Sowmya Somanath, Ehud Sharlin, and Mario Costa Sousa, used by permission data analysis and the purely stereoscopic setup by Butkiewicz and Ware [15, 16] for oceanographic data. Coffey et al.'s [21, 22] Slice WIM (Fig. 6.2) combines a large vertical stereoscopic projection of the explored 3D data with a horizontal tactile input surface. The core aspect of their system is the use of a stereoscopically displayed world-in-miniature visualization of the entire dataset using the large vertical display that also casts a shadow/projection onto the horizontal interaction surface. This connects both views and allows users to mentally map their 2D input into the stereoscopic 3D scene. The input itself is based on an interaction widget that allows the user to navigate the 3D view, manipulate exploration elements such as cutting planes, create curves in 3D space for path planning, and select subsets of the data such as bundles of flowlines as mentioned in Sect. 6.3.2.

In contrast to the previous hybrid setup, Butkiewicz and Ware [15, 16] use a purely stereoscopic data display that is unique in several ways. Their setup includes a slightly tilted stereoscopic screen on which also the tactile input is provided (captured using a depth camera) and a data view that is shown at a similarly tilted angle (Fig. 6.18). In addition, they designed their system for the exploration of oceanographic data. All these aspects together make it possible that the stereoscopic display does not conflict with tactile interaction because (a) the oceanographic data is usually rather shallow if viewed at a scale of large water bodies or oceans, (b) this data has an inherent surface with which to interact—the water surface, and (c) the tilted view of the data projection is quite similar to that of the display setup. This means that the touch surface can be easily placed roughly at the inherent interaction

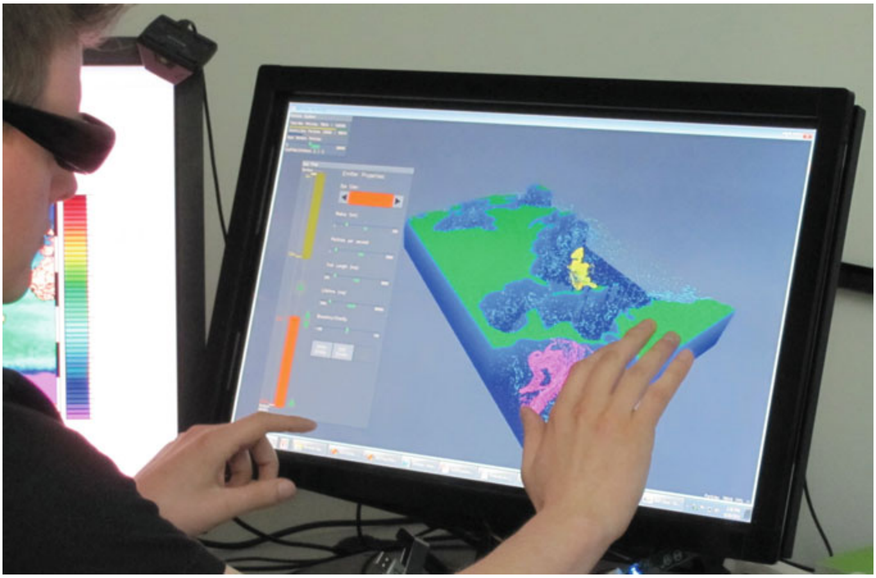


Fig. 6.18 Purely stereoscopic setup for the exploration of oceanographic data [15, 16]. Image courtesy of and © Thomas Butkiewicz, used by permission

surface (i.e., with zero parallax), without flexible 3 DOF rotations being necessary that would disrupt this ideal alignment. Using this setup they allow 2 DOF translations of the visualization, zooming, the placement of dye poles that allow the exploration of dynamic aspects of the displayed flow simulation. Butkiewicz and Ware also use specific precise interaction techniques such as two-handed mappings for the exact placement of dye poles.

Other interaction techniques with stereoscopic 3D data displays and visualizations are being investigated such as the use of a grasping-based metaphor [27] or the use of the monoscopic display in a hybrid setup as a mobile input device [60]. Yet, none of these approaches has yet led to a complete design study so we do not describe them in detail in this list of systems.

6.5 Conclusion and Open Research Challenges

With this survey we have demonstrated that the surface-based exploration of 3D visualizations is not only an active field of research but also has led to approaches that support the basic interaction techniques including 3D navigation, selection, data manipulation, seed particle placement, and more. The surface-based interaction benefits from the ability to provide spatial input directly at the location where the data is shown, thus inherently supports direct manipulation which is essential for the exploration of scientific data.

Yet, several research questions remain open for the field of surface-based exploration of three-dimensional, spatially explicit scientific data. For example, it seems clear that tactile interaction will only become another interaction modality to explore data, it will by no means replace existing approaches such as VR-based environments or traditional workstation settings. This means that the integration of surface-based data exploration into a **practical workflow for domain experts** is a pressing issue with the goal of providing an **interaction continuum** [41] in which the data analysts can choose the best interaction paradigm for the situation as well as easily transition between different paradigms as necessary.

To arrive at such a continuum, we have to continue the work on better **understanding the suitability of different input paradigms** for different data exploration tasks to be able to use tactile, tangible, haptic, traditional, or other sensing as it works best (e.g., [9, 48, 49]). The future continuum can then also include data exploration environments that effectively integrate tactile, surface-based interaction with other visual or interaction paradigms such as stereoscopic views (e.g., [15, 16, 21, 22, 60]) or tangible input (e.g., [71, 74]).

Connected to this issue of creating an interaction continuum is the challenge of providing **coherent interaction designs** even for a single interaction paradigm. The techniques reviewed in Sect. 6.3, in particular, often focus on a single type of manipulation only. Their interaction mappings are thus relatively flexible and without many external constraints. Yet, in practice analysts require a whole toolkit of data exploration techniques. Section 6.4 provided some examples for how several interaction techniques can be combined using coherent mappings. In practice, however, it is likely that many more techniques are needed so that more work is necessary to understand how to best integrate a large set of interaction techniques within the context of surface-based data exploration.

Another open research question is the issue of the applicability of widget-based or of gestural specifications of the interaction intents. In our discussion we, on purpose and with only a few exceptions [6, 59], did not mention the use of gestural interaction techniques—it turns out that almost all published interaction techniques rely on interaction mappings that are clearly specified based on the location of input points with respect to the data or interaction widgets. Such posture-based interaction [42] has the benefit that any input point motion can directly be interpreted as data manipulations. This interaction specification paradigm thus avoids complex mode specification and allows users of respective systems to concentrate on the data analysis. Yet, in some situations such as the initiation of specific data exploration actions the use of gestures may be useful, so this question of **widget-based versus gestural interaction control** still needs to be further explored in the future.

From our survey it also became apparent that the data exploration scenarios that exist today focus primarily on single-user interaction. Only two of the discussed systems [55, 61] were specifically designed with **collaboration and parallel input** in mind. However, collaboration is still possible with the other systems when the different collaborators are taking turns. It would thus be nice to see more work in the future that specifically explores collaborative settings, both those that require turn-taking and those that allow parallel work.

Acknowledgements I would like to thank all the authors of the original papers who kindly provided me with images for their techniques in order for me to be able to illustrate the described interaction techniques and systems. Also, I would like to thank all people who have collaborated with me in the past on surface-based interaction techniques for 3D visualization, in particular my past and present students who worked with me on this topic including Lonni Besançon, Tijmen Klein, David López, Pjotr Svetachov, Thomas ten Cate, and Lingyun Yu.

References

1. Abeele S, Bock O (2001) Mechanisms for sensorimotor adaptation to rotated visual input. Exp Brain Res 139(2):248–253. doi:10.1007/s002210100768
2. Abeele S, Bock O (2001) Sensorimotor adaptation to rotated visual input: different mechanisms for small versus large rotations. Exp Brain Res 140(4):407–410. doi:10.1007/s002210100846
3. Akers D (2006) CINCH: a cooperatively designed marking interface for 3D pathway selection. In: Proceedings of UIST. ACM, New York, pp 33–42. doi:10.1145/1166253.1166260
4. Argelaguet F, Andujar C (2009) Efficient 3D pointing selection in cluttered virtual environments. IEEE Comput Graph Appl 29(6):34–43. doi:10.1109/MCG.2009.117
5. Argelaguet F, Andujar C (2013) A survey of 3D object selection techniques for virtual environments. Comput Graph 37(3):121–136. doi:10.1016/j.cag.2012.12.003
6. Au OKC, Tai CL, Fu, H (2012) Multitouch gestures for constrained transformation of 3D objects. Comput Graph Forum, 31(2pt3): 651–660. doi:10.1111/j.1467-8659.2012.03044.x

7. Banic A (2014) Selection classification for interaction with immersive volumetric visualizations. In: Human interface and the management of information: information and knowledge design and evaluation. LNCS, vol 8521. Springer, Cham/Heidelberg, pp 10–21. doi:10.1007/978-3-319-07731-4_2
8. Bedford F (1993) Perceptual learning. In: Medin DL (ed) The psychology of learning and motivation, vol. 30. Academic Press, Inc., New York, pp 1–60. doi:10.1016/S0079-7421(08)60293-5
9. Besançon L, Issartel P, Ammi M, Isenberg T (2016) Usability comparison of mouse, touch and tangible inputs for 3D data manipulation. Tech. Rep. arXiv:1603.08735
10. Bock O (2013) Basic principles of sensorimotor adaptation to different distortions with different effectors and movement types: a review and synthesis of behavioral findings. Front Hum Neurosci, 7(81) (2013). doi:10.3389/fnhum.2013.00081
11. Bogorin MA, Luderschmidt J, Dörner R, Geiger C (2009) ViNet—Interaction with information visualizations in VR applications via multi-touch and tangible user interfaces. In: Proceedings 6th GI Workshop "Virtuelle und Erweiterte Realität". Shaker-Verlag, Aachen, pp 211–222
12. Bowman DA, Kruijff E, LaViola JJ Jr, Poupyrev I (2005) 3D User interfaces: theory and practice. Addison-Wesley, Boston
13. Bruder G, Steinicke F (2013) 2.5D touch interaction on stereoscopic tabletop surfaces. In: Proceedings ISIS3D, pp 1–4
14. Bruder G, Steinicke F, Stürzlinger W (2013) To touch or not to touch? Comparing 2D touch and 3D mid-air interaction on stereoscopic tabletop surfaces. In: Proceedings SUI. ACM, New York, pp 9–16. doi:10.1145/2491367.2491369
15. Butkiewicz T, Ware C (2011) Exploratory analysis of ocean flow models with stereoscopic multi-touch. In: IEEE Vis Posters
16. Butkiewicz T, Ware C (2011) Multi-touch 3D exploratory analysis of ocean flow models. In: Proceedings OCEANS. IEEE, Los Alamitos, pp 746–755
17. Buxton B (2007) Multi-touch systems that I have known and loved. http://www.billbuxton.com/multitouchOverview.html. Updated last on June 12, 2014, visited in March 2016
18. Card SK, Mackinlay JD, Shneiderman B (1999) Readings in information visualization: using vision to think. Morgan Kaufmann, San Francisco
19. Chan LW, Kao HS, Chen MY, Lee MS, Hsu J, Hung YP (2010) Touching the void: direct-touch interaction for intangible displays. In: Proceedings CHI, ACM, New York, pp 2625–2634. doi:10.1145/1753326.1753725
20. Coffey D, Korsakov F, Keefe DF (2010) Low cost VR meets low cost multi-touch. In: Proceedings ISVC, vol 2. Springer, Berlin, Heidelberg, pp 351–360. doi:10.1007/978-3-642-17274-8_35
21. Coffey D, Malbraaten N, Le T, Borazjani I, Sotiropoulos F, Erdman AG, Keefe DF (2012) Interactive Slice WIM: navigating and interrogating volume datasets using a multi-surface, multi-touch VR interface. IEEE TVCG 18(10):1614–1626. doi:10.1109/TVCG.2011.283
22. Coffey D, Malbraaten N, Le T, Borazjani I, Sotiropoulos F, Keefe DF (2011) Slice WIM: a multi-surface, multi-touch interface for overview+detail exploration of volume datasets in virtual reality. In: Proceedings I3D. ACM, New York, pp 191–198. doi:10.1145/1944745.1944777
23. Cohé A, Dècle F, Hachet M (2011) tBox: a 3D transformation widget designed for touch-screens. In: Proceedings CHI. ACM, New York, pp 3005–3008. doi:10.1145/1978942.1979387
24. Colley A, Häkkilä J, Schöning J, Daiber F, Steinicke F, Krüger A (2015) Touch the 3rd dimension! Understanding stereoscopic 3D touchscreen interaction. In: Computer-Human Interaction: Cognitive Effects of Spatial Interaction, Learning, and Ability (extended papers of OzCHI 2013). LNCS, vol. 8433. Springer, Berlin/Heidelberg, pp 47–67. doi:10.1007/978-3-319-16940-8_3
25. Cunningham HA (1989) Aiming error under transformed spatial mappings suggests a structure for visual-motor maps. J Exp Psychol Hum Percept Perform 15(3):493–506. doi:10.1037/0096-1523.15.3.493

26. Cutler LD, Fröhlich B, Hanrahan P (1997) Two-handed direct manipulation on the responsive workbench. In: Proceedings SI3D. ACM, New York, pp 107–114. doi:10.1145/253284.253315
27. Daiber F (2011) Interaction with stereoscopic data on and above multi-touch surfaces. In: ITS Doctoral Colloquium. ACM, New York, pp 2:1–2:4. doi:10.1145/2076354.2076428
28. van Dam A, Forsberg AS, Laidlaw DH, LaViola JJ, Simpson RM (2000) Immersive VR for scientific visualization: a progress report. IEEE Comput Graph Appl 20(6):26–52. doi:10.1109/38.888006
29. Fu CW, Goh WB, Ng JA (2010) Multi-touch techniques for exploring large-scale 3D astrophysical simulations. In: Proceedings CHI. ACM, New York, pp 2213–2222. doi:10.1145/1753326.1753661
30. Giesler A, Valkov D, Hinrichs K (2014) Void shadows: multi-touch interaction with stereoscopic objects on the tabletop. In: Proceedings SUI. ACM, New York, NY, USA, pp 104–112. doi:10.1145/2659766.2659779
31. Hachet M, Bossavit B, Cohé A, de la Rivière JB (2011) Toucheo: multitouch and stereo combined in a seamless workspace. In: Proceedings UIST. ACM, New York, pp 587–592. doi:10.1145/2047196.2047273
32. Hachet M, de la Rivière JB, Laviole J, Cohé A, Cursan S (2013) Touch-based interfaces for interacting with 3D content in public exhibitions. IEEE Comput Graph Appl 33(2):80–85. doi:10.1109/MCG.2013.34
33. Hancock M, ten Cate T, Carpendale S (2009) Sticky tools: full 6DOF force-based interaction for multi-touch tables. In: Proceedings ITS. ACM, New York, pp 145–152. doi:10.1145/1731903.1731930
34. Hancock M, ten Cate T, Carpendale S, Isenberg T (2010) Supporting sandtray therapy on an interactive tabletop. In: Proceedings CHI. ACM, New York, pp 2133–2142. doi:10.1145/1753326.1753651
35. Hancock MS, Carpendale S, Vernier FD, Wigdor D, Shen C (2006) Rotation and translation mechanisms for tabletop interaction. In: Proceedings Tabletop. IEEE Computer Society, Los Alamitos, pp. 79–88. doi:10.1109/TABLETOP.2006.26
36. Hand C (1997) A survey of 3D interaction techniques. Comput Graph Forum 16(5):269–281. doi:10.1111/1467-8659.00194
37. Hornecker E, Marshall P, Dalton NS, Rogers Y (2008) Collaboration and interference: awareness with mice or touch input. In: Proceedings CSCW. ACM, New York, pp 167–176. doi:10.1145/1460563.1460589
38. Isenberg P, Isenberg T (2013) Visualization on interactive surfaces: a research overview. i-com 12(3), 10–17. doi:10.1524/icom.2013.0020
39. Isenberg P, Isenberg T, Hesselmann T, Lee B, von Zadow U, Tang A (2013) Data visualization on interactive surfaces: a research agenda. IEEE Comput Graph Appl 33(2):16–24. doi:10.1109/MCG.2013.24
40. Isenberg T (2011) Position paper: touch interaction in scientific visualization. In: Proceedings DEXIS, pp 24–27
41. Isenberg T (2014) An interaction continuum for visualization. In: Proceedings IEEE VIS Workshop on Death of the Desktop: Envisioning Visualization without Desktop Computing
42. Isenberg T, Hancock M (2012) Gestures vs. postures: 'Gestural' touch interaction in 3D environments. In: Proceedings 3DCHI, pp 53–61
43. Jackson B, Coffey D, Keefe DF(2012) Force brushes: progressive data-driven haptic selection and filtering for multi-variate flow visualizations. In: Short Paper Proceedings EuroVis. Eurographics Association, Goslar, Germany, pp 7–11. doi:10.2312/PE/EuroVisShort/EuroVisShort2012/007-011
44. Jackson B, Lau TY, Schroeder D, Toussaint KC Jr, Keefe DF (2013) A lightweight tangible 3D interface for interactive visualization of thin fiber structures. IEEE Trans Vis Comput Graph 19(12):2802–2809. doi:10.1109/TVCG.2013.121
45. Jackson B, Schroeder D, Keefe DF (2012) Nailing down multi-touch: anchored above the surface interaction for 3D modeling and navigation. In: Proceedings Graphics Interface. Canadian Information Processing Society, Toronto, pp 181–184. doi:10.20380/GI2012.23

46. Jankowski J, Hachet M (2013) A survey of interaction techniques for interactive 3D environments. In: Eurographics State of the Art Reports. Eurographics Association, Goslar, Germany, pp 65–93. doi:10.2312/conf/EG2013/stars/065-093
47. Jansen Y, Dragicevic P (2013) An interaction model for visualizations beyond the desktop. IEEE Trans Vis Comput Graph 19(12):2396–2405. doi:10.1109/TVCG.2013.134
48. Jansen Y, Dragicevic P, Fekete JD (2012) Tangible remote controllers for wall-size displays. In: Proceedings CHI. ACM, New York, pp 2865–2874. doi:10.1145/2207676.2208691
49. Jansen Y, Dragicevic P, Fekete JD (2013) Evaluating the efficiency of physical visualizations. In: Proceedings CHI. ACM, New York, pp 2593–2602. doi:10.1145/2470654.2481359
50. Jönsson D, Falk M, Ynnerman A (2016) Intuitive exploration of volumetric data using dynamic galleries. IEEE Trans Vis Comput Graph 22(1):896–905. doi:10.1109/TVCG.2015.2467294
51. Keefe DF (2010) Integrating visualization and interaction research to improve scientific workflows. IEEE Comput Graph Appl 30(2):8–13. doi:10.1109/MCG.2010.30
52. Keefe DF, Isenberg T (2013) Reimagining the scientific visualization interaction paradigm. IEEE Comput 46(5):51–57. doi:10.1109/MC.2013.178
53. Keefe DF, Zeleznik RC, Laidlaw DH (2008) Tech-note: dynamic dragging for input of 3D trajectories. In: Proceedings 3DUI. IEEE Computer Society, Los Alamitos, pp 51–54. doi:10.1109/3DUI.2008.4476591
54. Kin K, Agrawala M, DeRose T (2009) Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In: Proceedings Graphics Interface. CIPS, Toronto, pp 119–124. doi:10.20380/GI2009.16
55. Klein T, Guéniat F, Pastur L, Vernier F, Isenberg T (2012) A design study of direct-touch interaction for exploratory 3D scientific visualization. Comput Graph Forum 31(3):1225–1234. doi:10.1111/j.1467-8659.2012.03115.x
56. Knoedel S, Hachet M (2011) Multi-touch RST in 2D and 3D spaces: studying the impact of directness on user performance. In: Proceedings 3DUI. IEEE Computer Society, Los Alamitos, pp 75–78. doi:10.1109/3DUI.2011.5759220
57. Krüger W, Fröhlich B (1994) The responsive workbench. IEEE Comput Graph Appl 14(3):12–15. doi:10.1109/38.279036
58. Kruszyński KJ, van Liere R (2009) Tangible props for scientific visualization: concept, requirements, application. Virtual Real 13(4):235–244. doi:10.1007/s10055-009-0126-1
59. Liu J, Au OKC, Fu H, Tai CL (2012) Two-finger gestures for 6DOF manipulation of 3D objects. Comput Graph Forum 31(7):2047–2055. doi:10.1111/j.1467-8659.2012.03197.x
60. López D, Oehlberg L, Doger C, Isenberg T (2016) Towards an understanding of mobile touch navigation in a stereoscopic viewing environment for 3D data exploration. IEEE Trans Vis Comput Graph 22(5):1616–1629. doi:10.1109/TVCG.2015.2440233
61. Lundström C, Rydell T, Forsell C, Persson A, Ynnerman A (2011) Multi-touch table system for medical visualization: application to orthopedic surgery planning. IEEE Trans Vis Comput Graph, 17(12). doi:10.1109/TVCG.2011.224
62. Marton F, Rodriguez MB, Bettio F, Agus M, Villanueva AJ, Gobbetti E (2014) IsoCam: interactive visual exploration of massive cultural heritage models on large projection setups. J Comput Cult Herit, 7(2), 12:1–12:24. doi:10.1145/2611519
63. Moscovich T, Hughes JF (2008) Indirect mappings of multi-touch input using one and two hands. In: Proceedings CHI. ACM, New York, pp 1275–1284. doi:10.1145/1357054.1357254
64. Novotný M, Lacko J, Samuelčík M (2013) Applications of multi-touch augmented reality system in education and presentation of virtual heritage. Proc Comput Sci 25:231–235. doi:10.1016/j.procs.2013.11.028
65. Owada S, Nielsen F, Igarashi T (2005) Volume catcher. In: Proceedings I3D. ACM, New York, pp 111–116. doi:10.1145/1053427.1053445
66. Reisman JL, Davidson PL, Han JY (2009) A screen-space formulation for 2D and 3D direct manipulation. In: Proceedings UIST. ACM, New York, pp 69–78. doi:10.1145/1622176.1622190
67. Robles-De-La-Torre G (2006) The importance of the sense of touch in virtual and real environments. IEEE MultiMed 13(3):24–30. doi:10.1109/MMUL.2006.69

68. Rogers Y, Lindley S (2004) Collaborating around vertical and horizontal large interactive displays: which way is best? Interact Comput 16(6):1133–1152. doi:10.1016/j.intcom.2004.07.008
69. Schmalstieg D, Encarnação LM, Szalavári Z (1999) Using transparent props for interaction with the virtual table. In: Proceedings I3D. ACM, New York, pp. 147–153. doi:10.1145/300523.300542
70. Shan G, Xie M, Li F, Gao Y, Chi X (2014) Interactive visual exploration of halos in large-scale cosmology simulation. J Vis 17(3):145–156. doi:10.1007/s12650-014-0206-5
71. Song P, Goh WB, Fu CW, Meng Q, Heng PA (2011) WYSIWYF: exploring and annotating volume data with a tangible handheld device. In: Proceedings CHI. ACM, New York, pp 1333–1342. doi:10.1145/1978942.1979140
72. Steinicke F, Hinrichs KH, Schöning J, Krüger A (2008) Multi-touching 3D data: towards direct interaction in stereoscopic display environments coupled with mobile devices. In: Proceedings AVI Workshop on Designing Multi-Touch Interaction Techniques for Coupled Public and Private Displays, pp 46–49
73. Sultanum N, Sharlin E, Sousa MC, Miranda-Filho DN, Eastick R (2010) Touching the depths: introducing tabletop interaction to reservoir engineering. In: Proceedings ITS. ACM, New York, pp 105–108. doi:10.1145/1936652.1936671
74. Sultanum N, Somanath S, Sharlin E, Sousa MC (2011) Point it, split it, peel it, view it: techniques for interactive reservoir visualization on tabletops. In: Proceedings ITS. ACM, New York, pp 192–201. doi:10.1145/2076354.2076390
75. Sultanum N, Vital Brazil E, Costa Sousa M (2013) Navigating and annotating 3D geological outcrops through multi-touch interaction. In: Proceedings ITS. ACM, New York, pp 345–348. doi:10.1145/2512349.2512396
76. Sundén E, Bock A, Jönsson D, Ynnerman A, Ropinski T (2014) Interaction techniques as a communication channel when presenting 3D visualizations. In: Proceedings 3DVis. IEEE, Los Alamitos, pp 61–65. doi:10.1109/3DVis.2014.7160102
77. Tan DS, Gergle D, Scupelli P, Pausch R (2006) Physically large displays improve performance on spatial tasks. ACM Trans Comput Hum Interact 13(1):71–99. doi:10.1145/1143518.1143521
78. Teather RJ, Stuerzlinger W (2011) Pointing at 3D targets in a stereo head-tracked virtual environment. In: Proceedings 3DUI. pp 87–94. doi:10.1109/3DUI.2011.5759222
79. Telea AC (2015) Data visualization: principles and practice, 2nd edn. A K Peters/CRC Press
80. Tong X, Chen CM, Shen HW, Wong PC (2015) Interactive streamline exploration and manipulation using deformation. In: Proceedings PacificVis. IEEE, Los Alamitos, pp 1–8. doi:10.1109/PACIFICVIS.2015.7156349
81. Valkov D, Giesler A, Hinrichs KH (2014) Imperceptible depth shifts for touch interaction with stereoscopic objects. In: Proceedings CHI. ACM, New York, pp 227–236. doi:10.1145/2556288.2557134
82. Valkov D, Steinicke F, Bruder G, Hinrichs K (2011) 2D touching of 3D stereoscopic objects. In: Proceedings CHI. ACM, New York, pp 1353–1362. doi:10.1145/1978942.1979142
83. Valkov D, Steinicke F, Bruder G, Hinrichs KH, Schöning J, Daiber F, Krüger A (2010) Touching floating objects in projection-based virtual reality environments. In: Proceedings EGVE/EuroVR/VEC. Eurographics Association, Goslar, Germany, pp 17–24. doi:10.2312/EGVE/JVRC10/017-024
84. Ware C, Arsenault R (2004) Frames of reference in virtual object rotation. In: Proceedings APGV. ACM, New York, pp 135–141. doi:10.1145/1012551.1012576
85. Welch RB (1978) Perceptual modification: adapting to altered sensory environments. Academic Press, New York. doi:10.1016/B978-0-12-741850-6.50002-1
86. Wiebel A, Vos FM, Foerster D, Hege HC (2012) WYSIWYP: what you see is what you pick. IEEE Trans Vis Comput Graph 18(12):2236–2244. doi:10.1109/TVCG.2012.292
87. Wills GJ (1996) Selection: 524,288 ways to say "This is interesting". In: Proceedings InfoVis. IEEE Computer Society, Los Alamitos, pp 54–60. doi:10.1109/INFVIS.1996.559216

88. Yu L, Efstathiou K, Isenberg P, Isenberg T (2012) Efficient structure-aware selection techniques for 3D point cloud visualizations with 2DOF input. IEEE Trans Vis Comput Graph 18(12):2245–2254. doi:10.1109/TVCG.2012.217
89. Yu L, Efstathiou K, Isenberg P, Isenberg T (2016) CAST: Effective and efficient user interaction for context-aware selection in 3D particle clouds. IEEE Trans Vis Comput Graph 22(1):886–895. doi:10.1109/TVCG.2015.2467202
90. Yu L, Svetachov P, Isenberg P, Everts MH, Isenberg T (2010) FI3D: Direct-touch interaction for the exploration of 3D scientific visualization spaces. IEEE Trans Vis Comput Graph 16(6):1613–1622. doi:10.1109/TVCG.2010.157

Chapter 7
CubIT: Design and Evaluation of a Collaboration-Tool for Large Interactive Wall Surfaces

Markus Rittenbruch

Abstract In this book chapter we describe the design and evaluation of CubIT, a multi-user presentation and collaboration system installed at the Queensland University of Technology's (QUT) Cube facility. The 'Cube' is an interactive visualisation facility made up of five very large-scale interactive multi-panel wall displays, each consisting of up to twelve 55-inch multi-touch screens (48 screens in total) and additional very large projected display screens situated above the display panels. The chapter outlines the unique design challenges, features, implementation and evaluation of CubIT. The system was built to make the Cube facility accessible to QUT's academic and student population. CubIT enables users to easily upload and share their own media content, and allows multiple users to simultaneously interact with the Cube's wall displays. The features of CubIT are made available via three user interfaces, a multi-touch interface working on the wall displays, a mobile phone and tablet application and a web-based content management system. Each of these interfaces play different roles and offers different interaction mechanisms, appropriate to the underlying platform. Through its interfaces CubIT supports a wide range of collaborative features including multi-user shared workspaces, drag and drop upload and sharing between users, session management and dynamic state control between different parts of the system. The results of our longitudinal evaluation study showed that CubIT was successfully used for a variety of tasks, but also highlighted specific challenges with regards to user expectations as well as issues arising from public use.

7.1 Introduction

The Queensland University of Technology (QUT) recently opened an interactive exhibition and learning space as part of its newly established Science and Engineering Centre. The facility named 'The Cube' features five very large interactive

M. Rittenbruch (✉)
Queensland University of Technology, 2 George Street, Brisbane
QLD 4000, Australia
e-mail: m.rittenbruch@qut.edu.au

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_7

multi-panel wall displays, each consisting of up to twelve 55-inch multitouch screens (48 screens in total) as well as very large projected displays situated above the display panels (see Fig. 7.1). The Cube facility constitutes a demanding real-world application environment. It is publicly accessible and has been designed to support large numbers of visitors, which include university staff and students, the general public and potentially large groups of visiting school students. Users can interact with a range of bespoke applications specifically built for the Cube [10].

CubIT, is a large-scale multi-user presentation and collaboration system, specifically designed to allow QUT's staff and students to utilise the display and interaction capabilities of the Cube. CubIT's primary purpose is to enable users to upload, interact with and share their own media content on the Cube's display surfaces using a shared workspace approach. Users can log into CubIT on any of the Cube's wall surfaces using their RFID-enabled (Radio-frequency identification) staff or student card. When they do so, they are given access to their individual user workspace. User workspaces contain media content which users have previously uploaded to the system, including images, video and text files, as well as custom-built presentations. Users can simultaneously open items from their individual workspace and display them on the shared multitouch canvas, as well as the large projection displays. Multiple users can share the canvas at the same time. Since the upload of user-generated content was a core requirement for the system, cross-device interaction was an important consideration in the early design process. CubIT supports user interaction across different devices and screen sizes. The system contains custom-build applications for smartphones and tablets that allow users to create content and easily upload it to the system, and instantly display it on the wall surfaces. Content can also be transferred between the multi-touch wall

Fig. 7.1 Two of the five wall displays in the Cube facility

surfaces and the very-large projection displays, enabling users to view content at different scales.

The design of CubIT, in the context of the Cube facility, posed a range of specific challenges, including:

- What is an appropriate conceptual design that makes use of the existing multiple large-scale multi-touch and projection surfaces to support the collaboration between multiple users?
- What are appropriate interaction mechanisms for large-screens that allow potentially inexperienced users to intuitively use the system?
- How can users be supported to effectively work across different interaction devices and surfaces?
- How do we address issues that arise from the fact that the system is situated in a public setting, including dealing with authentication and content moderation?

Some of the specific interaction techniques used in CubIT have been previously explored in a number of lab and "in the wild" studies (see related work). However, the scale and specific interactive capabilities of the Cube made it necessary to apply known concepts, like the shared desktop metaphor, to very large wall-sized displays in a public setting. In this book chapter we address resulting challenges and discuss how they impacted on the design, implementation and use of the system. We complement this reflection by discussing the results of a longitudinal evaluation study which examined the system's use, usability, user experience and use context. It is important to note that given the nature of the application context and its exposure to the public, CubIT has is not a prototypical development, but rather a fully operational system that has been used extensively over a nine-month period (between February 2013 and March 2014).

7.2 Background and Related Work

Multitouch-based interaction approaches have been studied in a wide range of settings such as collaboration (e.g. [5, 8]) and education (e.g. [14]). Recent advances in display technology, such as stackable thin-bezel LCD displays, have led to the availability of large, high-resolution multitouch displays that can be combined into very large, nearly seamless, interactive surfaces. These large interactive screen surfaces have opened a range of new opportunities, as well as challenges for the design of interactive applications. They allow application developers to create rich interaction environments that enable multiple users to simultaneously interact with digital content across large shared surfaces. There are a number of examples for systems that use very-large interactive screens in public settings. For instance, CityWall [9] was built to allow multiple users to interact with a given set of digital content on a large-scale, rear-projected, multi-touch wall display, installed in the city centre of Helsinki. Peltonen et al. [9] showed that users engaged in a rich set of interaction practices and established social conventions to manage the shared screen real estate. Similarly, Schematic implemented a multi-touch wall display[1] that allowed participants of an international advertising festival to simultaneously log into the system using their RFID pass cards. Once authenticated users were able to browse schedules, access way-finding information and exchange social networking information. However, while both these applications support the exploration of a given set of predefined content, they were not designed to support the direct upload and interaction with user-generated content. There is further an increasing understanding of how people interact using large shared surfaces. For instance, Azad et al. [1] extended Scott et al's. [11] notion of territoriality to large vertical displays. While Scott et al's. original work focussed on relatively small shared tabletop surfaces, Azad et al. explored the collaborative use of space on large vertical surfaces in public spaces, for a given set of tasks. While the work highlighted common patterns of interaction, further work is needed to transfer the results to real-world collaborative environments with non-predetermined tasks.

While the availability very-large multi-touch surfaces is still a relatively recent phenomenon, the broader use of (large) interactive screens to support small-group interaction and collaboration has been extensively studied, in particular in the context of interactive meeting rooms and purpose build interaction labs. One particular focus of study has been the question how to use shared displays to facilitate the interaction between co-located users, and more specifically how to enable users to share application and media content contributed from personal computing devices, such as laptops. For instance, WeSpace [13] allowed multiple co-located users to jointly connect laptops 'on the fly' and share their desktop session in display environment consisting of a shared multitouch and a projection surface. Broughton et al. [4] extended this notion of collaboration and implemented a distributed 'blended interaction space' which supported the distributed collaboration between multiple groups of remote users via replicated tabletop setups combined with high quality vide-conferencing. Earlier research into "Multiple Display Environments" (e.g. [2, 6]) investigated small group collaboration across multiple devices and displays, however these setups generally use non-interactive shared displays. Users in these environments usually control shared application via their laptop mouse pointers. Such setups commonly employ a 'replication' approach, which allows users to share individual off-the-shelf applications or whole desktops on the shared display(s). However, other systems, such as Dynamo [6], implemented a different approach and instead provided custom content viewers, which allowed users to share content-specific information (e.g. URLs, media, documents) rather than whole applications. Both approaches have advantages and disadvantages. A 'replication' approach allows users to share specialised applications, that are specifically suited to a particular target domain. For instance, WeSpace [13] was designed to support collaboration amongst Astrophysicists. However, one particular drawback when implementing the 'replication/approach on multitouch screens is

[1]http://www.possible.com/news-and-events/cannes-lions-touchwall.

that off-the-shelf applications, running on laptops, are almost exclusively single-user, single-mouse applications that are not optimised for the interactive capabilities and scale of large multitouch screens. By contrast, approaches, that support the sharing of content rather than applications are more widely applicable and can be specifically designed to utilise the interactive capabilities of the interactive surfaces they run on.

In addition to the question how to build systems that allow users to access and use their own (media) content, the question of how to implement such systems in a public environment poses additional challenges. Shen et al. [12], for instance, explored the use of collaborative multi-touch tables for ad hoc collaboration in public locations like airport lounges. The research featured the notion of a “walk-up” setup, highlighting the importance of being able to set up collaborative sessions and share content with relative ease and without the need for physical data or display connections. However, the work does not cover questions of content moderation and system access prevalent to real-work settings. Izadi et al. [6] studied how public displays could become a resource for multiple users to interact and share content. While the Dynamo system [6] shares many conceptual similarities with our approach, it differs across a range of dimension including technological setup (e.g. Dynamo used collaborative multi-pointer interaction of a public shared workspace controlled through laptops) and scale.

In summary, there is a large body of research that addresses various aspects of large-interactive screen and multi-device interaction in particular in the context of small group collaboration. However, the specific scenario described in this chapter, supports the collaboration of multiple simultaneous users on very-large multitouch wall surfaces in combination with mobile interaction devices, poses challenges that have not been addressed in detail. We will outline some of these challenges in the following section.

7.3 System Design

The overall design goal for CubIT was to make the Cube accessible to all of QUT’s staff and students and allow them to display and interact with their own media content on the Cube displays. The motivation was based on early discussions with stakeholders who remarked that while the Cube was planned to feature custom-build applications and content it did not sufficiently support academic researchers to utilise the facility and showcase their own work. CubIT’s vision was for the Cube to be become a system that supported “spontaneous walk up interaction and collaboration” on user-generated content. Given that the Cube display was situated in a public space the predominant purpose CubIT was seen as supporting the collaborative display of existing user-content to interested third parties, rather than supporting the collaborative generation of new content, a task that is more suited to non-public environments like meeting rooms and labs.

To address this overall design goal CubIT was developed as part of a user-centred design process that took into account feedback from a wide range of potential users, across different faculties, divisions and student bodies within QUT. The design context for CubIT was predicated on a number of factors. First, the Cube facility itself, in particular its layout, technical infrastructure, multitouch capabilities and public accessibility, had a large impact on the design of CubIT. This meant the system had to work on the different wall setups and include interaction mechanisms for both the multitouch as well as the projected displays. The fact that the facility was public-facing further meant that user authentication and content moderation became crucial. Second, the intended user base of QUT staff and students was very broad and involved a wide range of academic and professional backgrounds. As a result the intended system had to be generically applicable and usable even for casual users. Third, since the system aimed to support the upload of user-generated content, this process needed to be made as easy as possible and integrate cross-device support to allow users to use personal computing devices such as smart phones and tablets.

7.3.1 Functional Scope

Based on the design goal and reflection of the design context we built a series of low- and medium-level prototypes that were presented to potential users in two design workshops. Key technical and usage goals that defined the basic functionality were set early in the process to clarify the design scope. These key capabilities included:

- *Authentication*: Users should be able to authenticate and access their content, without having to use onscreen keyboards to type in their password.
- *Multi-user capable*: The multi-touch surface should support multiple users simultaneously using the system.
- *Ease of upload*: Uploading media content should be as easy as "flinging" content to the screen using a mobile device.
- *Sharing*: Users should be able to easily share content using the multi-touch screens.

In addition to these key capabilities we made a number of decision early in the design process, with regards to the way in which users would interact with the system. The design of the multi-touch interface followed a *messy desktop* metaphor, representing media content as scalable, rotatable and translatable widgets on a large canvas, which allowed users to move content around freely. Since the technology we used could not distinguish different users touching the screens, the system was built around the notion that content on the screen could be controlled by any user. With regard to representing user's media content, we decided to implement a localised, individually identifiable, content container per user, referred to as 'user workspace handle'. Users would share the common canvas to display and interact with content, but access and manage their own content in their respective user workspace handles. Once a user had authenticated with the system their handle would appear on the shared canvas that displayed the username and avatar and allowed users to access their content. Lastly, we designed a number of basic features used to allow users to manage their content on the canvas, including the ability to hide all their content on screen, move all their content simultaneously to a different part of the screen, as well as manage their own workspace by reordering, adding and removing content.

7.3.2 Design Workshops

We ran two design workshops with prospective users of the system, in order to gather user requirements, discuss usage scenarios, receive feedback on low-level prototypes and discuss the potential functional scope of CubIT (see Fig. 7.2). The first workshop was run in February 2012 and consisted of 22 staff and students from a mix of faculties and divisions. The second workshop was held in March 2012 with 15 academics from the Science and Engineering faculty. Both workshops followed the same format and contained three sections. Section one consisted of an introduction of the existing mockups and prototypes (see Fig. 7.3) and a hand-on exploration of the capabilities of the multi-touch screens. The second session aimed to collect 'user stories' which envisioned how the Cube infrastructure and the CubIT concept could be leveraged it the participant's specific work context. Participants were split into small groups and invited to answer three questions with regards to the potential system: *"How does this relate to my work?", "How would this help you?", "What do I need it to do?"*. The last section allowed participants to create paper-based prototypes that featured desired functionality.

The workshops resulted in a rich set of user stories and design concepts. The most commonly mentioned concepts that correlated with the design goals were:

Fig. 7.2 First CubIT design workshop

Fig. 7.3 Mockups of the **a** multi-touch interface (*left*) and **b** mobile interface (*right*)

- *Top screen presentation*: Use the top screen (i.e. the projection display) for presentations (during times of activity) or auto-play content (during idle times).
- *Top screen dock*: A dock along the top of the multi-touch canvas allows users to push content to the top screen (projected). The content is iconised and allows users to control content on the otherwise inaccessible top screen.
- *Session*: Support sessions so users can create specific compositions of content and refer back to them. Open the last saved session when a user logs back in.
- *Demo user*: Create a dedicated demo user that contains interesting material and relevant public information. Users who are not authenticated can use the demo user to interact with the system.
- (*Mobile*) *Annotations*: Allow users to annotate content. A potential input mechanism is to use a smart phone/tablet.
- Rights management: Allow users to specify rights per content item (allow copy, share-alike, etc.)
- *Remote access*: Push content to remote locations (e.g. other campuses) and receive remote feeds (e.g. live lectures).

The outcomes of the design workshops informed the central functionality that was implemented.

7.4 The CubIT System

7.4.1 System Components

CubIT features three distinct user interfaces, each of which provides different functions and interaction mechanisms: a *multi-touch interface* running on the Cube large display walls, a *web-based content platform* and a *mobile interface*. The web-based interface (implemented in Ruby on Rails) allows users to upload and manage content and further supports system administrators in the moderation of content and the administration of user accounts. The multitouch interface (implemented in Python using the Kivy[2] framework) enables users to interact with content on the large-scale multi-touch displays of the Cube and share content between users. The mobile interface (built in iOS, supporting iPhones and iPads) presents a mechanism to upload and create content on the fly. We will discuss each of these interfaces and the functions they support in detail in the next sections.

7.4.2 Multi-touch Interface

The CubIT multi-touch interface allows users to display and interact with the media content which they have uploaded to the system. Users log in by swiping their RFID card on one of the readers located underneath the multi-touch screens.[3] Once a user logs in, their user workspace handle (see below) appears on the shared workspace. The application is location-sensitive, the workspace handle appears on the screens that is associated with the closest RFID reader. This feature allows users to log out from one screen and move to a different part of the screen (or a different wall altogether) to log in again, effectively moving their content to different locations.

User workspace handle. The user workspace handle (see Fig. 7.4) represents a user's content in the system. It consists of an avatar, username label, scrollable workspace containing the media content and two function buttons, "pin/unpin content" and "minimise/maximise". The scrollable workspace displays the media content in the form of thumbnails. CubIT contains four different types of thumbnails for images, videos, text and presentations. Thumbnails can be dragged or clicked to be opened on the workspace. Thumbnails can also be dragged around the workspace handle to be reordered. An option to delete an item from the system is presented if a thumbnail is pressed for a slightly longer period of time (see Fig. 7.5a). The z-order for user handles is set to be higher than any other content on the screen ensuring that the user workspace is always accessible, and not obscured.

Media items. Media content items are *images*, *videos* or textual *notes* that appear as zoomable, rotateable and translatable widgets on the screen (see Fig. 7.5b). The zoom factor is limited to allow images to scale up to no more than the width of three portrait panels (3240 pixels) to prevent individual content items from obscuring the whole canvas. Videos can be played on-screen and have a standard set of video controls (pause, play, seek, volume). When opened from the workspace, each media item can be opened multiple times, spawning multiple instances on the canvas. If

[2] https://kivy.org/.

[3] Each of the Cube's display walls is equipped with a number of RFID readers, generally one reader per 2 panels.

Fig. 7.4 CubIT interface elements

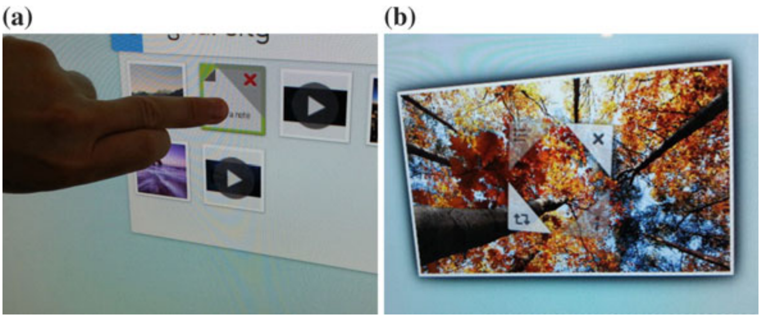


Fig. 7.5 **a** Delete item (*left*), **b** Media item with overlay sub menu (*right*)

items are permanently deleted from the workspace (or the system via the web interface) all of the items currently open instances of an item are closed. All content widgets use dragging physics to allow for content to be thrown. The friction settings are designed to limit the throwing distance to approximately 2–3 panels, preventing users from interfering with the workspace of users at the other end of a display wall.

Pinning. Each user workspace handle has a pinning button (see Fig. 7.4) allowing users to "pin" down the content relative to their handle and move all the content at once. This allows users to navigate the screen and move all their content to a different part of the screen while maintaining the relative content layout.

Minimize/maximize. User workspace handles further contain a minimise/maximise button as part of the item's overlay sub-menu (see Fig. 7.5b). Minimising content means that the widget is animated back into the handle. Minimise and maximise maintain the relative position and layout of media items.

The layout is saved as a session and is persistent across logouts. Sessions are shared between different instances of CubIT running on different walls. As a result, users can lay out their content in a particular way (e.g. for a poster presentation) and re-apply this layout to multiple setups (e.g. CubIT running on 3 different walls).

Presentations. CubIT includes a custom presentation widget (see Fig. 7.6) that allows users to display stacks of images, videos and notes in a more convenient manner. Presentations can be created using the web and mobile interfaces. The presentation widget contains several components. The display section allows content items to be displayed, scaled and swiped like a slideshow. The handle identifies the presentation. The selection box underneath the handle allows easy access to the surrounding slides and can be used to scroll through and navigate the presentation. Presentations can be edited dynamically using the multi-touch interface. In order to edit a presentation, users can press the presentation workspace button and open the presentation's workspace. A presentation workspace provides the same functionality as a user workspace and allows users to reorder, delete and add content on the fly.

Top dock and Top dock view. The layout and design of the Cube includes large projection screens on top of walls of interactive panels. As a result, each project implemented on the Cube had to find ways to design their system to make use of the projection screens while maintaining control over the interaction on the interactive touch panels below. In the case of CubIT, we decided to allow users to "throw" individual media item up to the projection screen to be displayed at full resolution. The rationale for this design option was to allow users to interact with content

Fig. 7.6 **a** CubIT presentation (*left*) and **b** presentation workspace (*right*)

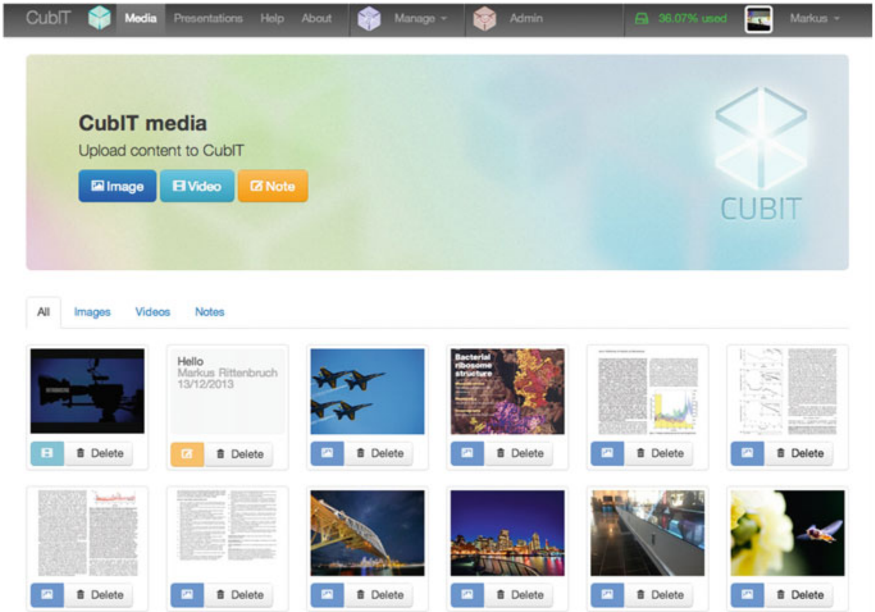


Fig. 7.7 CubIT web interface

closely on the touch panels, while using the projection surfaces for presentations to larger groups.

The mechanism in CubIT that controls the content on the top projection screen is called "top dock". It consists of a docking area stretching along the top border of the multi-touch panels. Media items that are dragged into the dock are displayed on the top screen. The top dock supports all media types images, note and videos, which auto-play when dragged onto the top dock.

Drag and drop sharing. The multi-touch interface supports sharing of content between users. In order to copy content items between accounts, users drag thumbnail representations of images, videos, notes or presentations into a different workspace. This creates a new instance of the copied object, which is now independent of the original. Because the system does not differentiate between users, objects can be freely copied between accounts by any user who touches the screen. To account for this, user accounts can be put into a "safe" exhibit mode to display of content over longer periods of time, in case users want to leave bits of content on screen for others to see (e.g. notice board).

7.4.3 Web Interface

The CubIT web-interface (see Fig. 7.7) is one of the two mechanisms allowing users to upload and maintain content on CubIT. The interface uses a standard user registration and login system. As part of the registration process users can register their RFID cards allowing them to log into the system on the multi-touch wall. The web interface for a standard user account consists of two main sections. The “Media” section allows users to upload image and video content and create notes. Users can browse existing content and delete items. The “Presentation” section enables users to create and manage presentations. Users can add content already uploaded to the system to new and existing presentations, as well as delete existing presentations.

Further sections comprise a page specifying the location and installation instructions for the mobile application, an about page with general information about the project and an accounts page allowing users to change their user details and avatar image.

7.4.3.1 Admin and Moderator Roles

The web interface further supports two roles for users with elevated privileges, admins and moderators. For each of these role an additional section is displayed. Moderators can browse through all existing media content in the system and delete content and ban, unban or remove users. Moderators can set systems parameters like a user’s data quota and change user’s account privileges (e.g. promote to moderator, admin).

The moderator function was added to response to the potential issue of users uploading inappropriate content. The CubIT content is highly visible and potentially exposed to a large number of visitors. Moderation is conducted on a regular basis, after content gets uploaded. Moderation approval prior to uploading was not considered in order to allow users to upload content immediately, without having to wait for approval. If inappropriate content gets detected moderators have several options. They can remove the content and/or ban the user. Banned users will not be able to log into the any of the CubIT interfaces and receive a message informing them that they have been banned. Once the situation has been clarified, banned users can be reinstated. Moderators can further completely remove users from the system. Users who are being banned while they are logged into the multi-touch interface will be logged out and all their content is removed from the display.

7.4.4 Mobile Interface

The CubIT mobile interface is a native iOS application (see Fig. 7.8) which runs on iPhones (Fig. 7.8a) and iPads (Fig. 7.8b). The purpose of the interface is to allow users to easily upload content while away from their desks, and in particular, while standing in front of one of the touch screens. The mobile interface has four modes (represented by four icons at the bottom of the screen). Three of those modes are dedicated to different media types allowing users to upload images, videos, and

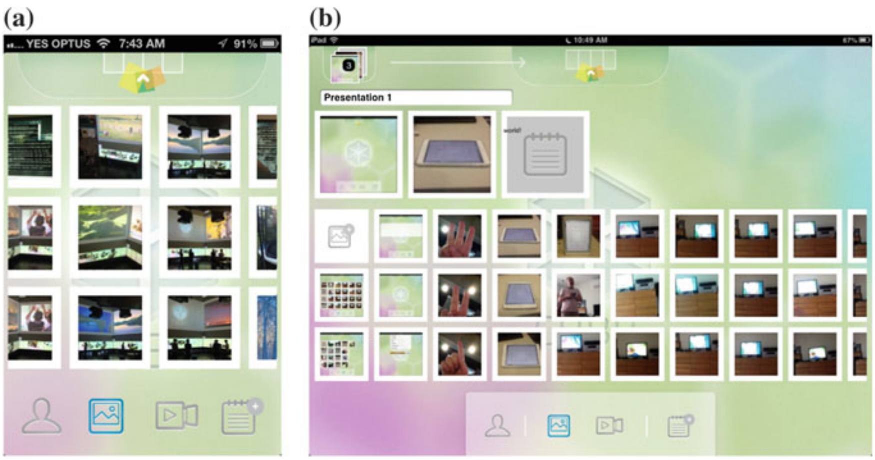


Fig. 7.8 CubIT mobile interface interface, **a** iPhone image upload (*left*), **b** iPad presentation creation (*right*)

notes respectively. The fourth mode allows users to change their avatar picture and log out of the application. The iPad version, due to its larger screen real estate, features an additional function. It allows users to create presentations from existing media sources and upload these presentations to CubIT. Users can scroll through their iPhone/iPad's media library in a scrollable section in the middle of the application. An "add icon" links to the device's camera application and allows users to create and upload content on the fly. The upload mechanism consists of a simple drag and drop mechanism. To upload, users drag images into the upload icon on top of the screen. An animation gives the appearance that the item is "sucked" into the screen and then uploaded. The upload mechanism has been designed to give the appearance of being able to "flick" multiple content items to the multi-touch walls.

In addition to its function as an upload device the mobile interface was also used as an input mechanism. As part of the design process it was decided that using on-screen keyboards on a shared multi-user display was likely to be less efficient, than allowing user to input text via their personal mobile devices. Thus the functionality that requires text input, such as notes as well as creating presentations, was implemented on the mobile as well as the web interface.

7.4.5 CubIT Collaborative Features

The system components described above have been designed to support co-located synchronous collaboration between users, within the context of the Cube. We summarise some of the collaborative features in turn:

Shared workspace and workspace control: Multiple users can share a large workspace canvas, where each user provides content using their user workspace handle. The system provides several mechanisms for users to manage the shared space. Users can "pin" their content and move it simultaneously to a different part of the screen. Users can minimize content, thus saving the layout of their current session and move it to a different part of the screen or a different display wall altogether. Interface elements have been specifically designed so that users can work together without obscuring each other's view of the workspace.

Drag and drop sharing: Users can simply share content by dragging and dropping content between user workspaces. This function extends to presentations by allowing users to create shared presentations on screen, with content provided by several users.

Easy upload from mobile devices: Drag and drop upload of content into workspaces allows users to dynamically add content to a shared workspace while working with others. Users can, for instance, capture the outcome of a joint discussion in an image or video and upload this directly to the shared workspace.

Dynamic state control between different parts of the system: The system dynamically synchronises state changes between the multi-touch mobile and web-interfaces. This allows users to dynamically update content on screen from a remote location (e.g. as part of a share co-located and remote design session).

7.5 System Setup

7.5.1 System Components

The CubIT system is made up of a number of system components (see Fig. 7.9). The *CubIT server* manages all aspect related to content and user management, including content upload (images, videos, notes), the creation of custom presentations, content delivery and maintaining workspace, session and authentication states. The CubIT *multitouch UI* manages touch interactions and widgets on the display panels, as well as the syncing and distribution of the interface's state across a series of multitouch screens and computing nodes. The *mobile UI* manages creation and upload of content as well as updates to user profiles.

The Cube's multitouch displays are driven by a series of graphics nodes, whereby each node drives two display panels. As a result the multitouch UI was implemented as a distributed application that is executed across a set of graphics nodes used to drive a particular wall surface. For instance, in case of a 12-panel wall the application is synchronised across 7 graphics nodes (2 panels per node plus one node for the projection screen).

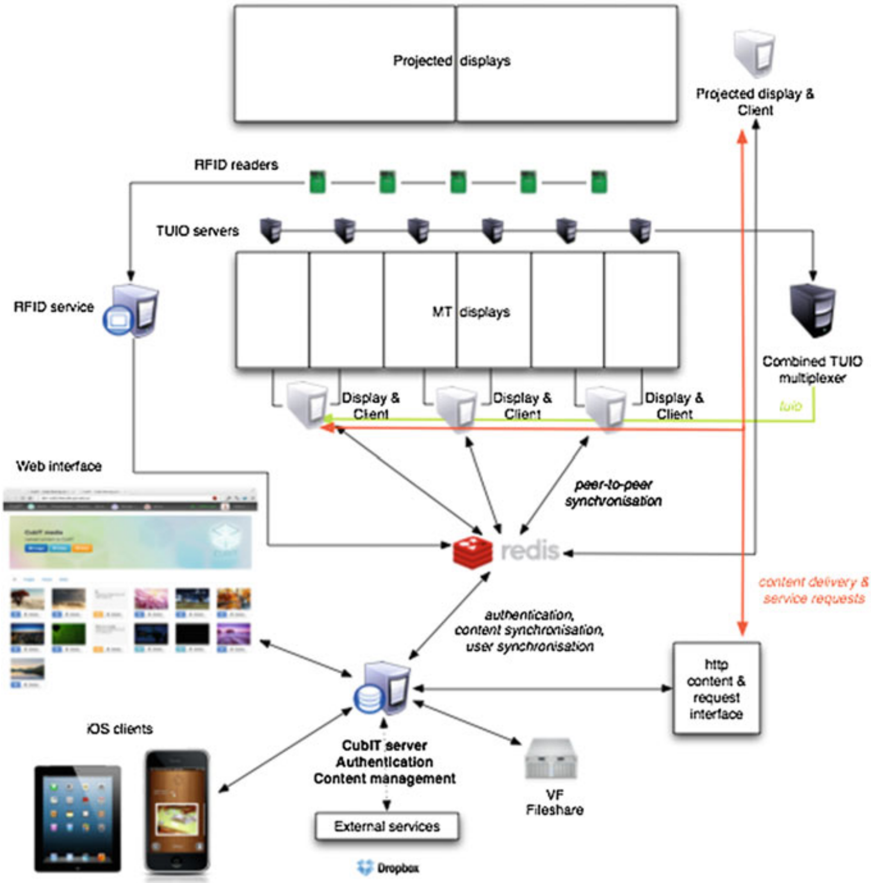


Fig. 7.9 CubIT system architecture

A *Redis server*[4] is used to maintain the consistent state of interface elements, send notifications between different system components and ensure a consistent state between the distributed graphics nodes that execute the multitouch UI. Each multitouch display panel has an integrated *TUIO server* recording touch events. These touch events get merged into a combined TUIO stream via a multiplexer, which flexibly reacts to the setup and number of CubIT instances running. The *RFID server* maintains the state of all RFID readers installed in the Cube and relays RFID event information via Redis to the CubIT server. The system diagram (see Fig. 7.9) is schematic and depicts a simplified version of our architecture, showing a single wall display consisting of 6 screens.

[4]http://redis.io/.

7.5.2 System Runtime Setup

CubIT can be simultaneously deployed to any of the five wall surfaces of the Cube. The CubIT (web) server maintains the state of logins across walls allowing users to log into multiple walls simultaneously. This functionality is mostly useful in cases where content gets displayed to the general public, for instance during exhibitions or conferences (see 'exhibition user' functionality).

7.5.3 Organisational Setup and Use

CubIT was deployed in January 2013 and has currently over 550 registered users. Since its' release the system has been used for a variety of different purposes. We will briefly outline some of the uses that have been observed since deployment:

Teaching: CubIT has been used to present student work in a number of classes taught at QUT. Students were encouraged to sign up to CubIT and create their own account. They uploaded their project work and displayed it during critique and student presentation sessions.

Events and Conferences: CubIT has been extensively used during conferences and events. Conference use included the display of posters and general conference related information such as sponsorship slides, videos and other promotional material. Many organisers specifically used the top dock presentation, by designing content that fitted the maximum resolution of the screen and allowed them to present wide posters (e.g. see Fig. 7.10b, top).

Visitors and demos: CubIT has been commonly used by academics to showcase research and other content to visitors. Several users regularly showcase their content to (groups of) visitors, such as potential industry partners and collaborators.

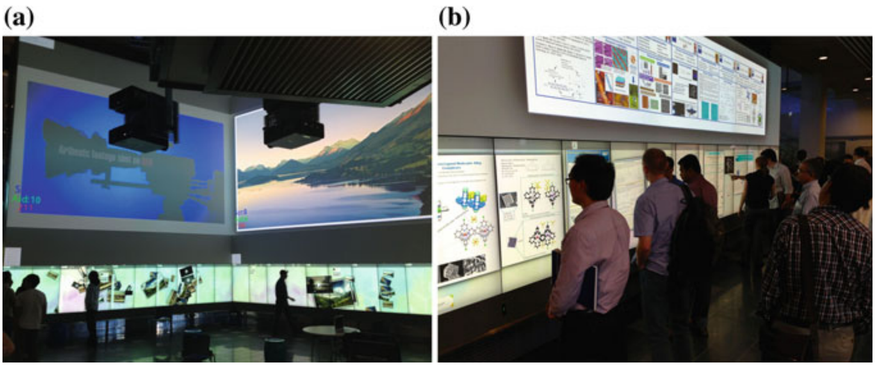


Fig. 7.10 Cubit on an **a** 20-screen (*left*) and **b** 12-screen wall surface (*right*)

School engagement: CubIT has been used as part of QUT's effort to engage school students. The school student program involved a guided tour of the Science and Technology Centre as well as the participation in various workshops and activities. These activities were documented by educators and uploaded to CubIT for students to browse.

It is important to note that as the Cube is a multi-purpose facility the software displayed on each of the wall surfaces, including CubIT, is subject to scheduling. During the usage period covered in this chapter (January–September 2013) CubIT was generally available by default on at least one of the wall surfaces and would run on other surfaces on request. However, scheduling could lead to situations where CubIT was not available when or where users expected it.

7.6 System Evaluation

7.6.1 Study Design

The study took place approximately 9 months after the system had been made available for public use in early 2013. Study participants were recruited amongst the 470 users who had signed up to use CubIT at that point in time. CubIT user consisted of QUT academics, professional staff, and students. An email was sent out to all users to invite them to participate in a 20-item questionnaire on the use, usability and user experience of CubIT. The questionnaire was open for 2 weeks and 48 participants completed the questionnaire. To protect the participant's privacy and to counter any potential demand characteristics effects the study was fully anonymised, so that the identity of participants could not be determined by the researchers conducting the study.

The questionnaire consisted of four different sections, *general information*, *system use*, *user experience* and *use context*. The *general information* section covered basic information about the background of the participants. The *system use* section queried which of the various aspect and functionalities of the system participants had used. The *user experience* section covered a range of usability and user experience measures. Last, the *use context* section consisted of questions that explored for which tasks the system had been used and contained open questions to determine attitudes towards the use of the system.

The user experience section contained a series of questions which were based on SUS (System Usability Scale) [3], a widely used usability questionnaire. We added one additional question in this part of the questionnaire, which queried participants' perception of the availability of CubIT on the Cube's wall surfaces (based on scheduling in the Cube). In addition to the usability questions, we ran a set of question relating to the user experience using UEQ (User Experience Questionnaire) [7]. Both, the usability as well as the user experience instruments were of a general nature and did not specifically target the multi-user or multi-touch capabilities of the application. However, they were coupled with a set of open-ended questions relating to people's experience with the system which allowed for a broader, qualitative assessment of the results.

7.6.2 Results

7.6.2.1 Functionality Use

We asked participants to rate whether they had used different functionalities of the system. The answers included *yes*, *no* and *do not know how* options.

The results (see Table 7.1) show that the fundamental functions of the system (how to sign up, log in, upload and display media content) were known to almost all users. More than half of the users had used functions to manage content on the screen (delete content, display presentation, used the pin button, etc.). And a smaller subsection of users had used the mobile features and installed the mobile app as well as uploaded content from their mobile device. Surprisingly, relatively few users had used the system to share content by dragging it to or from other user's workspaces to their own workspace (23 % and 10.5 % respectively).

7.6.2.2 User Experience

Tables 7.2 and 7.3 summarise the results of the usability and user experience evaluation of CubIT, ordered by mean values. All items were rated on a scale between 5 "strongly agree" and 1 "strongly disagree".

SUS and UEQ include both positively and negatively worded item. While we used alternating questions in our questionnaire, we inverted the scores and wording of the negative items when reporting our results, to achieve better comparability (inverted items are marked in both tables).

The outcomes of the usability evaluation were generally positive. A majority of participants felt that the system was easy to use and felt confident in using it. The question that received the most positive answers was whether CubIT should run on the Cube more often.

The participants on average agreed that the system meets all positive user experience factors. *Innovative* and *enjoyable* were the two highest rated items with a median of 5. All other factors with the exception of *secure* and *predictable* were rated with a median of 4. *Secure* received the lowest median score of 3. *Predictable* received the second lowest score with a median value of 3.5, indicating that the system was on average perceived to be just slightly more *predictable* than *unpredictable*.

Table 7.1 CubIT functionality use

Question	% Yes-No-Do not know how (n/a)
Logged into one of the display walls at the Cube using your QUT staff/student card	**92**-8-0 (0)
Signed up to CubIT using the web interface	**92**-8-0 (0)
Used CubIT on one of the display walls at the Cube	**90**-10-0 (0)
Used the web interface to upload media content	**85**-13-2 (0)
Dragged media content into the top dock	**81**-17-2 (0)
Used the minimise/maximise button	**79**-19-0 (2)
Used your workspace handle to open and display content	**77**-19-4 (0)
Used the CubIT web interface	**77**-19-0 (4)
Used the web interface to delete content	**58.5**-33.5-2 (6)
Used the pin button	**52**-36-6 (6)
Reordered content in your workspace	**52**-35.5-10.5 (2)
Displayed a presentation	**52**-42-4 (2)
Deleted content from your workspace	**50**-37.5-10.5 (2)
Used the mobile app to upload images or videos to CubIT	**38**-54-2 (6)
Used the web interface to create notes	**35.5**-54-6.5 (4)
Used the web interface to create presentations	**33.5**-54-8.5 (4)
Downloaded and installed the CubIT mobile iPhone app	**31**-61-2 (6)
Used the mobile app to upload content while standing in front of a CubIT display at the Cube	**31**-56-6 (6)
Used the web interface to delete presentations	**29**-56-11 (4)
Downloaded and installed the CubIT mobile iPad app	**25**-67-2 (6)
Copied content from another user's workspace into your workspace	**23**-60.5-14.5 (2)
Used the mobile app to create notes and upload them to CubIT	**17**-71-4 (8)
Copied content from your workspace into another user's workspace	**10.5**-71-14.5 (4)
Used the mobile app to change your avatar picture	**8**-75-11 (6)

7.6.2.3 Use Context

We asked participant to select multiple ways in which they used the system from a number of predetermined alternatives. The selection contained an open question allowing the participants to specify "other" activities. Table 7.4 shows the chosen activities in order of preference.

There was one entry for "other" activities, which indicated that the system was used as part of a *"high school competition"*.

The most common reported uses of the system included displaying own content, either generally, to colleagues, external visitors or as part of a presentation. About a third of the participants had left content on the screen for others to see. About 20 % had used CubIT as part of a conference presentation. The two activities that scored lowest were *exchanging content with others* and *giving a lecture*. The relatively low rate of participants who used the system to exchange content with other users

Table 7.2 Usability evaluation (SUS) results

Question	Med. (SD)	Mode	Mean
I think that CubIT should be running on the Cube more often	4(0.96)	5	4.07
I thought that CubIT was easy to use	4(1.06)	5	4.00
I did not find CubIT very cumbersome to use [inverted]	4(1.13)	4	3.84
I do not think that I would need the support of a technical person to be able to use CubIT [inverted]	4(1.39)	5	3.82
I felt very confident using CubIT	4(1.17)	5	3.80
I would imagine that most people would learn to use CubIT very quickly	4(1.18)	4	3.73
I did not find CubIT unnecessarily complex [inverted]	4(1.30)	5	3.73
I did not need to learn a lot of things before I could get going with CubIT [inverted]	4(1.4)	5	3.67
I am likely to use CubIT [inverted]	4(1.35)	5	3.62
I did not think there was too much inconsistency in CubIT [inverted]	3(1.12)	3	3.54
I am likely to share content with others using CubIT [inverted]	4(1.34)	5	3.51
I found the various functions in CubIT were well integrated	4(1.23)	4	3.43
I think that I would like to use CubIT frequently	3(1.38)	5	3.33

Table 7.3 User experience (UEQ) evaluation

Question	Med. (SD)	Mode	Mean
Innovative/conservative	5(1.09)	5	4.35
Enjoyable/annoying [inverted]	5(1.23)	5	4.16
Creative/dull	4(1.00)	4	4.07
Attractive/unattractive	4(1.06)	5	4.05
Exciting/boring	4(1.12)	5	4.00
Practical/impractical [inverted]	4(1.33)	5	3.88
Organized/cluttered [inverted]	4(1.16)	5	3.86
Clear/confusing [inverted]	4(1.18)	5	3.84
Efficient/inefficient [inverted]	4(1.21)	5	3.79
Fast/slow [inverted]	4(1.30)	5	2.64
Supportive/obstructive	4(1.14)	4	3.53
Secure/not secure	3(1.22)	3	3.26
Predictable/unpredictable [inverted]	3.5(1.45)	5	3.24

matches our observation, that the sharing function was only used by at most 23 % of participants. A total of two participants specified that they used the system to deliver a lecture.

Table 7.4 Activities CubIT was used for

Question	% Yes	% No
To display your own content	66.67	33.33
To test CubIT and understand how it works	64.58	35.42
To present content to colleagues/fellow students	58.33	41.67
To present content to a group of people	45.83	54.17
To present content to external visitors	41.67	58.33
To leave content on the screen for others to see	35.42	64.58
To display content as part of a conference/seminar	20.83	79.1
To exchange content with other users	10.42	89.58
To give a lecture	4.17	91.83

The second part of the use context section consisted of a series of open questions asking what people liked best and least about the system, as well as an open question that allowed participants to comment on their use of the system. We used a grounded theory approach to analyse the qualitative data and conducted open coding on the set of answers in order to determine relevant concepts and categories to structure the results. Answers to the question *"Do you have any other comments about CubIT, or this questionnaire?"* closely mirrored answers received in the questions regarding best and least liked aspects of CubIT and were coded together with these question.

7.6.2.4 Best Liked Aspects of CubIT

Regarding the question: *"which aspects of CubIT did you like best"*, we identified the following categories, which are ordered from most to least relevant.

Presentation of content: This category received the highest number of mentions across all participants. Participants generally appreciated being able to use CubIT to present content to colleagues and the general public. The category covers the general ability to present to different audiences as well as the ability to simultaneously display many content items on a the wall displays.

Interactive capabilities: The second most relevant category relates to the interactive capabilities that CubIT offers. Participants appreciated the scalability of content, moving content across different surfaces, support for different media types and being able to physically manipulate content through the multitouch interface.

Flexibility and openness: This category relates to the flexibility and openness of the system. These aspects were related to ability to display different content and use CubIT on different screen configurations. Participants also perceived that the system had many different uses. One participant remarked: *"CubIT can turn from an academic board to a social networking board instantly, depending on who is using it. As a social networking board, I love it."*.

Scale and wow-factor: The fourth-most relevant category is related on the impact that CubIT had on users and visitors. The size of the screen displays played and important role in how users perceived the system. One participant opined: *"CubIT's size is impressive. It's large enough to get anyone excited about using it"*. In addition to the screen size, CubIT was perceived as "cutting edge". Another participant mentioned: *"Its like Iron Mans office!"*.

Ease of use: The last category that received frequent mentions is how easy the system is to use. This includes numerous comments regarding the simplicity of use of the multi-touch interface, as well as the easy authentication via RFID Cards.

In addition to the categories mentioned above there are a number of other categories that were of relevance, but were overall less common. These include: *Multi-user capabilities*—Supporting multiple users at the same time; *Web and mobile integration:* Content upload via different interfaces; *Remote repository:* The notion of using CubIT as a remote repository for content accessed by ones' staff/student card.

7.6.2.5 Least Liked Aspects of CubIT

Regarding the question: "**which aspects of CubIT did you like least**", we identified the following categories. Like in the previous section, the categories are ordered from most to least relevant:

Interface improvements: This was the most commonly mentioned category, which related to a varied range of requests and suggestions to improve aspects of the user interface(s) and the overall system functionality. The issues mentioned were very diverse with no clear trend indicating one specific area that was of more pressing concern than others. The issues ranged from controlling video playback volume, additional remote presenter functionality for the top dock, to requests to allow users to reset passwords and RFID Card IDs.

Public use: A diverse set of issues arose around the public use of the system. The reported issues ranged from privacy and security concerns, to concerns about inappropriate content and behaviour to the question how suitable the public space is to deliver lectures. One participant raised their concern regarding inappropriate use of the system: "*Other people unrelated to our course/presentation playing loud, intrusive and offensive content during the time we were using it*".

Creation: One of the more common requests for additional functionality centred around tools that allowed users to create and annotate content directly on the multi-touch screen. The most mentioned functions were "interactive whiteboard" and "annotation of media items".

Media types: There were a number of requests for the system to support additional media types, such as Word documents and Web pages.

Reliability: some users reported reliability issues ranging from the feeling that elements were "freezing" to system crashes.

Availability: The next commonly mentioned category related to an organisational matter. Some participants commented that they would have liked to be CubIT to be more regularly available in the Cube or be available on a different screen/wall setup.

Other categories were mentioned occasionally. Some participants requested to make an *Android mobile* application available. Other participants made comments regarding the availability of *documentation*. These comments did not refer to the availability of general system documentation, but requested information about specific uses, e.g. how to use the system in the context of a particular class: "*No documentation I can get to guide me through how I might integrate it with my unit. Or run an assignment. This may be because it has not been used in this way previously*".

7.7 Discussion

The results of the study revealed which system functions were most commonly understood, how its usability and user experiences were rated, in which context the system was used, and which aspects of the systems and its use were most liked or disliked. The results generally indicated that CubIT fulfilled its purpose. However, there are a number of more subtle aspects that highlighted challenges related to the public use of the system and its ability to implement a wide range of functions, yet remain intuitive and flexible.

7.7.1 Usability, User Experience and Context

Regarding system use, the study showed that the majority of participants appreciated and had used the fundamental system functions. In particular functions related to the presentation of, and interaction with, media content on the multitouch screens were well understood. Surprisingly, two collaborative functions, the sharing of content by dragging it to or from other user's workspaces scored comparatively low. This matches the results from the "use context" part of the study, which showed that "*to exchange content with other users*" was the second least commonly engaged activity amongst our participants. Two other use aspects that scored low were the use of the mobile app to create and upload notes and changing the avatar picture via the mobile interface. While we predicted the latter function was likely to be used occasionally, the copying content functions and upload of notes were considered core functions during the design process.

One possible explanation for the lower than expected use, lies in the deployment strategy. As a side effect of the 'word of mouth' strategy, users received no formal training in the use of the system. While the system functions were generally perceived as being intuitive, some functions, like the ability to copy content between user workspace handles had to be discovered. An online manual was available through the web-interface, which covered this and many other functions. However, it is possible that this "cross-device" help approach was too removed from users who were interacting with the multitouch interface. Interestingly, we frequently observed that existing users would explain the system to their friends, but these explanations were often limited by what the explaining person knew about the function of the system.

The outcomes of the usability and the user experience evaluation were overwhelmingly positive. The number of positive responses to the question as to whether CubIT should run on the Cube more often, indicates that many of the participants were interested in using the system regularly. While availability is an obvious requirement for a 'walk up and use' system like CubIT, the system could however not always be made available, due to the multi-purpose nature of the Cube facility.

The results regarding the use context of CubIT closely matched the suggested categories. The most common reported uses of the system were those that matched the anticipated use of the system and represented its core functionality. Using the system to deliver lectures was uncommon, since all screens were in publicly accessible areas with significant amounts of thoroughfare and only public lectures would have been considered appropriate. This sentiment is mirrored by comments participants made regarding the public use of CubIT.

7.7.2 Public Use

The qualitative evaluation of CubIT resulted in rich set of categories. Some of the most interesting were *Flexibility and openness, Scale and wow factor, Public use, Creation and Media types*.

The *Flexibility and Openness* of the system was appreciated by most users and matches the fact that the system was perceived as usable, intuitive and well integrated. These aspects lead us to conclude the design goals of providing easy and intuitive access to the Cube and allowing users to interact with their own media content have been met. Comments made with regards to *Scale and wow factor* indicated the CubIT has used the display infrastructure of the Cube efficiently and that the scale of the interaction had a significant positive impact on the user experience.

Issues surrounding *public use* highlighted some of the tensions that can arise when placing an open user-generated content platform in a public space. The comments regarding the inappropriate behaviour of some users were particularly interesting. There is an obvious trade-off between the risk involved in managing content in a public environment and giving users the freedom to directly upload and interact with content on the display surfaces. Content moderation was implemented as part of the web-backend of CubIT. However, a conscious decision was made, not to moderate content upfront in order to give users the experience of "immediacy" when uploading content to the system. This strategy generally worked very well. Only one known case of inappropriate content had to be dealt with during the trial. We attribute this low number of incidents to the fact that all users of the system were identified by their QUT email address, which was required to sign up to the system. However, this strategy did not help to prevent the 'inconsiderate behaviour' reported by one of the participants.

7.7.3 Functional Scope Dilemma

Creation and *media types* were related categories that highlighted the challenge of building a generically applicable system for a diverse user population. Some participants requested both specialised tools (e.g. whiteboard functionality) and additional media formats (e.g. Word documents). A conscious decision was made early in the design process to limit the number of potentially complex functionality the system offered in favour of easy-to-understand functions (upload, display, present and share). While functionality like electronic whiteboards have been successfully implemented in electronic meeting rooms environments, they do add additional complexity and modalities to the user interaction, in particular when added on top of multi-user workspaces. Similar challenges arose from request for additional media content. While these requests were understandable they opened up the system to a multitude of integration issues. They would have required the integration potentially proprietary viewers (e.g. Microsoft Word viewer), and a modal interface that would switch the focus between the viewer and the workspace. Very few proprietary viewers have been designed for multitouch input or are likely to be consistent with the multitouch gestures used in CubIT. The challenge in the further development of CubIT and comparable systems is to integrate additional collaborative functionality within a consistent interaction framework that is suitable for casual users, does not require multiple modes of interaction and supports the simultaneous interaction of multiple users within a large shared workspace.

7.8 Conclusions

This chapter described the design, implementation, use and evaluation of CubIT, a large-scale, multi-user collaboration and presentation system. CubIT was specifically built to allow a broad user population to upload user-generated content to the Cube's interactive surfaces. Thus the systems' design not only had to take into account the Cube's physical and technical setup, but also define interaction paradigms that would allow casual users to jointly interact with and share content across a large shared multitouch canvas, as well as integrate interaction across different devices and surfaces, at different scales.

The resulting system was implemented across three user-interfaces: each of which fulfilled a different purpose. The multitouch interface was designed to allow users to display content on large-scale displays, authenticate with ease using RFID, present content to larger audiences on very large-scale projection surfaces, and easily share content across user accounts using various widgets and multitouch interaction mechanisms. The mobile interface was designed to provide textual input, allow for the grouping and creation of content (notes and presentations) and specifically, to allow users to upload content to the wall surfaces by 'flinging' content to the screens. Lastly, the web-based interface supported the same functionalities as the mobile interface. It, additionally handled user management tasks (authentication, user management, quota), help, and content management and administrative tasks for selected system administrators.

The evaluation of CubIT revealed that the system was generally perceived positively. It also highlighted some conceptual challenges, particularly questions related to the public use of the system, and managing the expectations of a broad user base as to what functionality the system should support. While CubIT has been built within the specific context of the Cube, we believe that many of its design and interaction principles, as well as the lessons learnt from the evaluation, transcend the physical setup and can be applied to different contexts and systems. We hope that software designers who develop systems for similar settings can learn from our experiences.

References

1. Azad A, Ruiz J, Vogel D, Hancock M, Lank E (2012) Territoriality and behaviour on and around large vertical publicly-shared displays. Paper presented at the ACM designing interactive systems conference (DIS '12). Newcastle, UK, 11–15 June 2012
2. Biehl JT, Baker WT, Bailey BP, Tan DS, Inkpen KM, Czerwinski M (2008) Impromptu: a new interaction framework for supporting collaboration in multiple display environments and its field evaluation for co-located software development. Paper presented at the SIGCHI conference on human factors in computing systems (CHI'2008). Florence, Italy, 5–10 Apr 2008
3. Brooke J (1996) SUS: a 'quick and dirty' usability scale. In: Jordon PW, Thomas B, Weerdmeester BA, McClelland IL (eds) Usability evaluation in industry. Taylor & Francis, London, pp 189–194
4. Broughton M, Paay J, Kjeldskov J, O'Hara K, Li J, Phillips M, Rittenbruch M (2009) Being here: designing for distributed hands-on collaboration in blended interaction spaces. Paper presented at the 21st annual conference of the australian computer-human interaction special interest group: design: open 24/7 (OzCHI '09). Melbourne, Australia, 23–27 Nov 2009
5. Conversy S, Gaspard-Boulinc H, Chatty S, Valès S, Dupré C, Ollagnon C (2011) Supporting air traffic control collaboration with a tabletop system. Paper presented at the ACM 2011 conference on computer supported cooperative work (CSCW '11). Hangzhou, China, 19–23 Mar 2011
6. Izadi S, Brignull H, Rodden T, Rogers Y, Underwood M (2003) Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media. Paper presented at the 16th annual ACM symposium on user interface software and technology (UIST '03). Vancouver, Canada, 2–5 Nov 2003
7. Laugwitz B, Held T, Schrepp M (2008) Construction and evaluation of a user experience questionnaire. Paper presented at the 4th symposium of the workgroup human-computer interaction and usability engineering of the austrian computer society (USAB'08). Graz, Austria, 20–21 Nov 2008

8. Morris MR, Lombardo J, Wigdor D (2010) WeSearch: supporting collaborative search and sensemaking on a tabletop display. Paper presented at the 2010 ACM conference on computer supported cooperative work (CSCW ‘10). Savannah, Georgia, USA, 6–10 Feb 2010
9. 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. Paper presented at the twenty-sixth annual SIGCHI conference on human factors in computing systems (CHI ‘08). Florence, Italy, 5–10 Apr 2008
10. Rittenbruch M, Sorensen A, Donovan J, Polson D, Docherty M, Jones J (2013) The Cube: a very large-scale interactive engagement space. Paper presented at the 2013 ACM international conference on interactive tabletops and surfaces (ITS’13). St Andrews, UK, 6–9 Oct 2013
11. Scott SD, Carpendale MST, Inkpen KM (2004) Territoriality in collaborative tabletop workspaces. Paper presented at the 2004 ACM conference on computer supported cooperative work (CSCW ‘04). Chicago, Illinois, USA, 6–10 Nov 2004
12. Shen C, Everitt K, Ryall K (2003) UbiTable: impromptu face-to-face collaboration on horizontal interactive surfaces. Paper presented at the fifth international conference on ubiquitous computing (UbiComp 2003). Seattle, Washington USA, 5–12 Oct 2003
13. Wigdor D, Jiang H, Forlines C, Borkin M, Shen C (2009) WeSpace: the design development and deployment of a walk-up and share multi-surface visual collaboration system. Paper presented at the SIGCHI conference on human factors in computing systems (CHI ‘09). Boston, MA, USA, 4–9 Apr 2009
14. Zadow UV, Buron S, Harms T, Behringer F, Sostmann K, Dachselt R (2013) SimMed: combining simulation and interactive tabletops for medical education. Paper presented at the SIGCHI conference on human factors in computing systems (CHI’13). Paris, France, 27 Apr–2 May 2013

Chapter 8
Shared Façades: Surface-Embedded Layout Management for Ad Hoc Collaboration Using Head-Worn Displays

Barrett Ens, Eyal Ofek, Neil Bruce and Pourang Irani

Abstract Collaboration is a necessary, everyday human activity, yet computing environments specifically designed to support collaborative tasks have typically been aimed toward groups of experts in extensive, purpose-built environments. The cost constraints and design complexities of fully-networked, multi-display environments have left everyday computer users in the lurch. However, the age of ubiquitous networking and wearable technologies has been accompanied by functional head-worn displays (HWDs), which are now capable of creating rich, interactive environments by overlaying virtual content onto real-world objects and surfaces. These immersive interfaces can be leveraged to transform the abundance of ordinary surfaces in our built environment into ad hoc collaborative multi-display environments. This paper introduces an approach for distributing virtual information displays for multiple users. We first describe a method for producing spatially-constant virtual window layouts in the context of single users. This method applies a random walk algorithm to balance multiple constraints, such as spatial constancy of displayed information, visual saliency of the background, surface-fit, occlusion and relative position of multiple windows, to produce layouts that remain consistent across multiple environments while respecting the local geometric features of the surroundings. We then describe how this method can be

Portions of this paper, including content from Sect. 8.1 through 8.4, and Figs. 8.2 and 8.4, were originally published as a conference paper, presented at SUI '15 [14].

B. Ens (✉) · N. Bruce · P. Irani
Department of Computer Science, University of Manitoba, Winnipeg, Canada
e-mail: bens@cs.umanitoba.ca

N. Bruce
e-mail: bruce@cs.umanitoba.ca

P. Irani
e-mail: irani@cs.umanitoba.ca

E. Ofek
Microsoft Research, Redmond, USA
e-mail: eyalofek@microsoft.com

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_8

generalized to include additional constraints from multiple users. For example, the algorithm can take the relative poses of two or more users into account, to prevent information from being occluded by objects in the environment from the perspective of each participant. In this paper, we however focus on describing how to make the content spatially-constant for one user, and discuss how it scales from one to multiple closely confined users. We provide an initial validation of this approach including quantitative and qualitative data in a user study. We evaluate weighting schemes with contrasting emphasis on spatial constancy and visual saliency, to determine how easily a user can locate spatially-situated information within the restricted viewing field of current head-worn display technology. Results show that our balanced constraint weighting schema produces better results than schemas that consider spatial constancy or visual saliency alone, when applied to models of two real-world test environments. Finally, we discuss our plans for future work, which will apply our window layout method in collaborative environments, to assist wearable technology users to engage in ad hoc collaboration with everyday analytic tasks.

8.1 Introduction

A new generation of lightweight head-worn displays (HWDs) is emerging. Due to advances in miniaturized sensing technology and computer vision and localization algorithms [2, 27, 41], these displays will be soon able to build reliable models of the user's surroundings in real time. The availability of such spatial information makes it possible to integrate personal information displays into the user's surroundings to assist navigation and sense-making in analytic tasks that rely on multiple sources of information [5, 13, 14]. If this transfer is done adequately, such systems should be able to support ad hoc collaborative work using HWDs.

These virtual displays are not bound by the constraints of physical displays and can float freely around the mobile user. However, there are situations where mapping virtual content to surrounding surfaces (Fig. 8.1a, b) can be beneficial. For example, dual disparity that results from placing virtual content at a different depth than the real-world background, can cause perceptual difficulties [28] and lead to eye fatigue [22]. Also, surface-aligned windows can make use of real-world landmarks to assist spatial memory and potentially to make use of tangible surfaces for direct input. However, the ideal placement of information remains an open research question; although some research has explored *display* placement for augmented reality (AR), little attention has focused on arranging virtual displays on a detailed 3D model of the environment.

This paper explores the *transition* of body-centric, single-user layouts to shared, world-based reference frames (Fig. 8.1c). This form of layout, of ***Shared Façades***, is necessary to facilitate collaborative tasks using HWDs. For example, imagine going out for a walk with your most common applications arranged in a 'bubble'

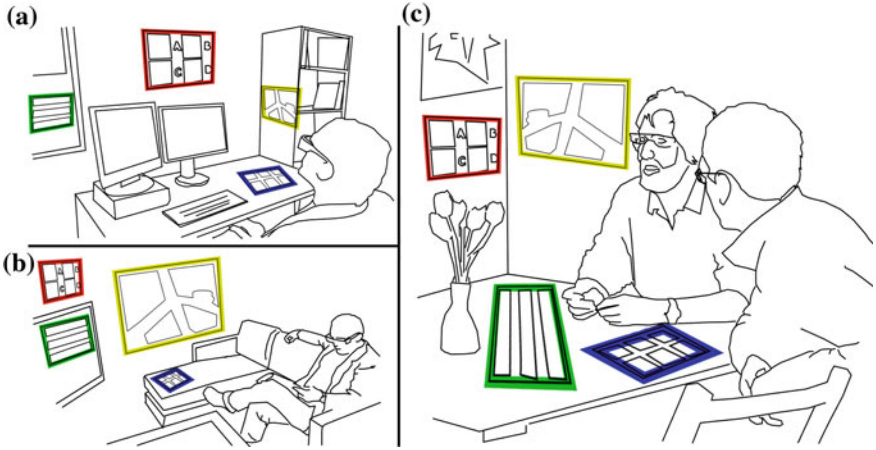


Fig. 8.1 Shared Façades embeds multiple applications on surfaces in the user's surroundings. Application window layouts remain spatially consistent in different environments (**a**, **b**) while avoiding occlusion of important scene objects. We argue that such a spatially consistent layout is key for collaboration (**c**) using Head-Worn Displays

that follows you; when you get home you can command these windows to drift to walls of your living room. At the same time, you can share some of these applications with your roommate, for instance a map location and review of a restaurant you passed earlier, to plan a night out. As the information is spatially-constant, 'looking-up' such content and finding it in the environment becomes trivial. Whereas the concept of the Personal Cockpit [13] focuses on a single-user, body-centric layout for organizing content on HWDs, Shared Façades focuses on where to situate application windows, to make it easier for a group of users to use. Furthermore, as we envision HWDs to become commonplace for the general consumer, Shared Façades is not necessarily focused on collaborative augmented reality applications [6, 44] but instead extends to any information space that can be embedded in the user's immediate surroundings.

We propose basing world-fixed spatial layouts on the user's egocentric reference frame (a) so that users can find applications using spatial memory of their default body-fixed layout; and (b) to maintain consistency of layouts between different environments. However, there are differences in the physical makeup of different spaces that must also be taken into account, thus we developed a layout manager that can balance multiple, sometimes opposing, constraints. For instance, a direct projection of an egocentric layout may cause an information display to be placed overhanging the edge of a desk or on a wall region containing important information. In such cases, our layout manager avoids surface boundaries and occlusion of important objects by nudging application windows to the nearest suitable location. Moreover, we provide tools for users to manually arrange such layouts to their liking. Once arranged, the user can transition the layout back to its egocentric (Personal Cockpit-style) form, then project that layout on a new environment

(Fig. 8.1c). This allows users to transition from a collaborative layout of content to one that is personal with minimal disruption to human visual search.

Our work makes several important contributions towards spatial layout management. First, we develop a layout manager that balances multiple constraints, including spatial constancy, visual salience, surface fit, window overlap and relative order. We implement several layout constraints and compare various constraint weighting schemes using an opportunistic algorithm. We explore a variety of interaction possibilities for configuring and managing these layouts. We conduct a formative user study, which verifies the layout manager's constraint weighting schemes and provides a wealth of qualitative user feedback that will benefit future designers. Finally, we discuss how the layout manager scales from a single user perspective to multiple users.

8.2 Related Work

The concept of window managers for assisting task organization can be traced back to early developments of personal computers. Their incorporation into spatial user interfaces (SUIs) occurred early in the history of virtual reality [16]. With the introduction of see-through head-mounted displays, researchers such as Feiner et al. [14] and Billinghurst et al. [5] imagined multiple windows being anchored to different objects in the environment or arranged in body-centric configurations. Later work recognized the potential of a 3D spatial environment for leveraging spatial memory to assist the recall of items [1, 40]. Computer users have been shown to be extremely adept at using spatial memory to find previously seen items, however this ability requires that items remain spatially constant [42, 45]. The application of spatial constancy to location recall has received little attention in the context of SUIs, despite foundational developments making it possible in AR applications (i.e. registration) [29].

In addition to optimizing spatial constancy, the Shared Façades layout manager takes into account surface geometry and background visual appearance for determining the placement of application windows in the surrounding environment. The exploitation of surface geometry is of potential benefit to HWD interfaces, for instance to improve content legibility by mitigating dual disparity [28] or to provide a tangible input surface. However surface structure has been explored primarily in the context of projection-based interfaces, which explicitly require a projection surface such as the workplace walls in the visionary Office of the Future [39]. The introduction of portable handheld projectors led to systems that dynamically adapt to the environment's surface geometry [9, 38]. An early goal of these systems was to correct distortion for legibility. Such perspective correction can also be used when the observer is mobile, for example in multi-display environments [35].

Surface detection has also inspired the development sophisticated 'immersive room' environments, in which projection surfaces encompass entire walls while maintaining awareness of objects within the room [26, 47]. Surfaces can be detected dynamically to allow projection onto moving surfaces such as paper or people's hands [40, 49]. Advanced prototype systems consisting of sensors and projectors have been developed to simultaneously map the environment and support projection-based interactions [34]. Surface detection has also been incorporated into AR interfaces through the exploration of low-cost techniques such as vanishing line detection [20, 30]. In contrast, the Shared Façades assumes the existence of a complete spatial model, which may in future be routinely stored and made available on demand.

In contrast to surface geometry, issues of interference with a display's background have been primarily explored in the realm of augmented and mixed reality. These applications require thoughtful placement of content with respect to the real-world background, particularly on see-through HWD screens, on which foreground content cannot be made fully opaque. For example, to mitigate the negative effects of background texture and luminosity on text legibility [18, 31], researchers proposed text color and contrast adjustments [18] or algorithms to move text to an optimal region of the display for readability [31]. Researchers further elaborated on such techniques by repositioning content dynamically for a moving background [36, 37] or by considering components such as background color [24, 46] or visual saliency [20]. Our layout manager is the first to our knowledge to combine both visual saliency and 3D geometric constraints within the same implementation.

In relation to window management, little research has been done to specifically provide efficient access to multiple applications in SUIs. Bell and Feiner [3] introduced an efficient algorithm for dynamically keeping track of available space. One particular work that is closely related to ours, describes the implementation of a window-manager for multi-projector displays, wherein windows are arranged to maximize the available projection space [48]. However, unlike our method, this layout manager does not maintain spatial constancy of application windows as it is not concerned with changing environments. Other work on HWD interfaces however has explored dynamic content placement, for example preventing occlusion of important objects [4]. Some early research explored the concept of attaching application windows to objects using fiducial markers [11, 14]. The Shared Façades determines layouts dynamically using only information extracted form camera images and a mesh model.

8.3 The Shared Façades

The Shared Façades is a multi-application management tool for stereoscopic HWDs. Its main component is a window layout generator that embeds virtual 2D application windows in the environment using camera image and depth sensor data. Automatic layouts are created at run time based on the user's current position and orientation, and take into account the geometry and layout of the room. Manual operations are also provided to manually configure layouts for analytic multitasking.

We developed Shared Façades layouts for single users, and validated their ability to maintain a spatially constant layout for this case, before discussing the scalability of this concept to multi-user collaboration.

8.3.1 Window Layouts in the Environment

The Shared Façades' layout generator uses a variety of constraints such as surface geometry and visual saliency to determine where to place application windows in the user's environment. Following is a list of the constraints that might be considered by a content manager for stereoscopic, see-through HWDs. Below, Table 8.1 provides a summary of these constraints along with a list of prior implementations that have considered each constraint. This list is not comprehensive but shows how our implementation fits within the current state of the art.

Surface structure—Indoor environments contain an abundance of flat, smooth surfaces, which are ideal for placing 2D content. Additional structural considerations are the size and shape of a given region, its 'orthogonality' (the facing direction of a window relative to the user's view), and its location, including relative direction and visibility.

Background appearance—Many regions in an environment will contain important visual information that should not be occluded. Visual saliency algorithms (e.g. [8]) model regions of a scene that are visually important to human observers. Additional semantic information can be used to identify highly important objects such as faces and text [4, 10]. In addition, transparent displays are susceptible to visual effects of texture, color and luminosity, thus regions that interfere with content legibility should be avoided.

Table 8.1 Constraints for placing application content in the environment and prior art that has used each constraint. Items uses in the shared Façades are marked with 'SF'

	Constraint	Usage
Surface structure	Surface normal	[26, 29, 30, 47], SF
	Size and shape	[48], SF
	Orthogonality	[35]
	View direction	SF
	Visibility	SF
Background appearance	Visual saliency	[20], SF
	Semantic importance	[4]
	Texture	[18, 31]
	Color	[24, 46]
	Luminosity	[18, 31, 36, 37]
Layout consistency	Spatial constancy	SF
	Window overlap	[48], SF
	Relative order	SF

Layout consistency—A layout manager should make it as easy as possible for users to find information. Spatial constancy, which has been shown to improve task switching time in standard desktop interfaces [45], should be preserved from one environment to another, despite environmental differences in surface structure and background appearance. The arrangement of windows should also minimize overlap and maintain the relative order of windows (i.e. left-to-right and top-to-bottom).

8.3.2 Generalizing to Multiple Users

One advantage of the layout generator's constraint-based random walk approach, is that any number of constraints may be added to the 'goodness' function (described in Sect. 4.1). Whereas Table 8.1 describes a number of such constraints for a single-user layout, it is equally possible to modify these constraints, or describe additional constraints for a collaborative multi-user layout.

For example, the concept of view direction can be modified to accommodate multiple users. In a single user interface, the view direction constraint prevents the layout from dispersing windows too far from the user's forward view. In a collaborative situation, a similar rule would keep windows near the group's shared point of focus in the collaborative workspace. However, the implementation of this constraint must take into account the viewing pose and perspective of each individual, and attempt to provide a common balance. Similarly, a constraint that prevents windows from being occluded by world objects must consider the viewing positions of each individual user.

New constraints, with no specific analogs in the single-user case, are required to deal with issues of privacy. In a shared workspace, different interface windows may have different levels of privacy for different users [12]. Each user may have their own personal documents, and some documents will be equally shared. In a Shared Façades implementation, a constraint can be added to keep windows containing personal information close to their owners, and shared windows near the center of the shared viewpoint. Alternatively, if limited open space is available for display placement, private information can be placed along shared information within the same space [20]. In either case, each instance of private information can be made visible only to its respective user.

Other constraints, such as visual saliency, size, and texture of a particular surface, apply equally well in single-user and collaborative scenarios.

8.3.3 Implementation

We implemented the Shared Façades algorithm using Unity3D on a desktop computer with an NVIDIA Quadro 600 GPU. We created two mock environments for development and testing, made to resemble a typical office and living room

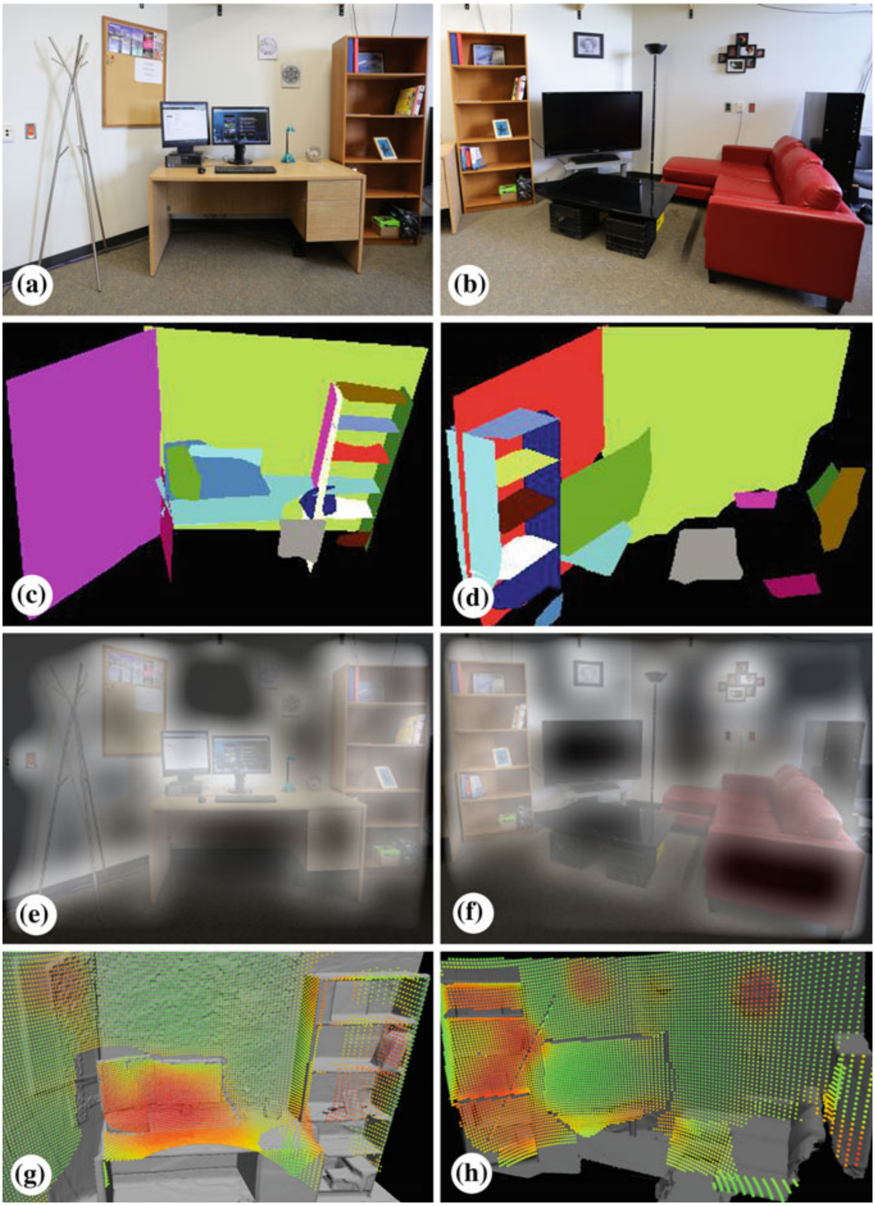


Fig. 8.2 The office (**a**) and living room test environments (**b**). Surface polygons generated from the mesh models (**c**, **d**). Saliency maps using AIM 8 (light regions are high salience, contrast increased for demonstration; **e**, **f** Saliency maps projected onto mesh models (*red* nodes are high salience, power law transformation is applied; **g**, **h**)

(Fig. 8.2a, b). Users are able to view the Shared Façades on an Epson BT-100 stereoscopic HWD with 23° diagonal FoV, tethered by a composite video input. We track the HWD using a high precision, low latency Vicon tracking system, thus the virtual content appears through the HWD to be accurately superimposed on the physical environments.

8.4 Automatic Layout Generator

Our window layout generator places windows in the mock environments using constraint-based constraints, such as the layout's spatial configuration and visual salience of the occluded background. A key contribution and component of this layout method is the application of spatial constancy in a real-world spatial layout. However, the concept of spatial constancy in a 3D spatial interface has several possible interpretations. Unlike the fixed space of a display screen, spatial interfaces inhabit various possible coordinate systems, for instance room-fixed [14] or body-centric [5]. Thus, constancy could imply that windows are fixed within the environment, or that they stay fixed relative to the user's body. In the Shared Façades, we take a hybrid approach; we use a body-centric reference frame to keep the layout consistent in different environments. However we assume the existence of a 'primary' viewing location and direction, which we use to transform the body centric reference frame on to room coordinates. This assumption holds in many real-world environments, for example in an office with a single desk chair. However many environments in reality have several such locations; it remains an interesting topic for future work to explore how and when users would opt to update their window layouts as they move about an environment.

Adherence to spatial constancy is furthermore complicated by the visual appearance of the surroundings. The layout generator displaces windows when important background objects are detected, but attempts to minimize such displacement. The primary goal of our layout generator is to perform a balancing act between these opposing constraints to provide satisfactory but efficient layouts.

Our layout generator uses a Monte Carlo approach derived from the Metropolis-Hastings algorithm [23], following from similar implementations that have been shown to be effective for creating constraint-based layouts of objects in space [19, 33]. This algorithm evaluates a series of proposed solutions, which are incrementally improved or for a fixed number of iterations, with some allowance for random perturbations. Within the evaluation, we define a set of weighted constraints that help us find a suitable layout for the application windows. A constraint is a function that generates a positive score indicating the 'goodness' of a window location (or set of window locations). While a great number of such constraints are imaginable, we used a minimalistic set in our implementation, with the primary constraints defined as follows:

Adherence indicates the location of a window with respect to its location in the default configuration. Windows with a high adherence are obeying the principal of spatial constancy, which makes them easy to find in different environments. The score calculated as the angle distance of a candidate position from the default position normalized over an arbitrarily chosen maximum angle of 30°.

Non-occlusion measures the degree to which a window is occluding, or overlapping important background information. To quantify this constraint, we measure the background saliency of a region that a window in the candidate position would occupy. High non-occlusion scores are given to windows in regions with low visual saliency.

We also apply constraints taking into account the *View Direction* (to align windows as closely as possible to the user's forward view), the *Surface Fit* (whether a window lies fully in a polygon), users's *Line-of-Sight* (all window corners are in visible locations), *Relative Order* of windows (whether windows maintain their spatial relations e.g. left-of), and *Overlap* (whether windows overlap others). Additional constraints, which we leave for future work could include *Color and Contrast* (choose locations to maximize legibility), *Predictable Locations* (align windows with landmarks such as room corners, wall centers or viewer horizon), *Maximal Size* (choose locations that allow large window sizes) and *Application Context* (by placing windows in locations that best suit the application context [17], for example a clock above a door or a weather report affixed to an outdoor window).

8.4.1 Window Layout Algorithm

The algorithm input consists only of data extracted from a mesh model and a single photo of each environment. The mesh models (Fig. 8.2g, h) were made using Kinect Fusion [25] and the photos (Fig. 8.2a, b) were taken with a typical SLR camera with a wide-angle lens (110°). In our current implementation we generate a static 3D Model of the scene beforehand, although we envision such a system working in dynamic settings in real time (see Summary and Future Work, below). We begin by searching the vertices of the mesh models for regions of uniform surface normal, from which we extract a set of surface polygons (Fig. 8.2c, d) using a greedy search with Hough transforms [43]. Meanwhile, we compute a saliency map of both scenes using the AIM saliency algorithm of Bruce and Tsotsos [8] (Fig. 8.2e, f). We chose this saliency method from many available options because of the high contrast and preserved boundaries regions in the saliency map. Finally, we calibrate the 3D model with the 2D image space of the saliency map [7] (Fig. 8.2g, h). This provides all of the information needed to enable a rich number of layout options for indoor scenes.

For calculating window layouts, we use a region 90° wide × 45° high, centered on the forward view, discretized into increments of 5°. Since finding the optimal layout L for a set of n windows in a given environment is not currently obtainable at interactive speed, we instead use a Monte Carlo approach derived from the Metropolis-Hastings algorithm [23]. Similar implementation have been shown to be effective for creating constraint-based layouts of objects in space [19, 33]. This algorithm evaluates proposed solutions which are incrementally improved or for a fixed number of iterations, with some allowance for random perturbations.

First, we define the *layout solution space* as the set of all possible assignments of a set of application windows W to unique points in P_E. We define a 'goodness' function

$$Goodness(L) := \sum_i \alpha_i \cdot r_i(L_i)$$

where α_i is an optional weight, $r_i: (O_i \subseteq O) \rightarrow \mathbb{R}$ is a constraint operating on a subset of the parameters O, L is a proposed layout solution, and L_i is a subset of the layout containing only the windows with constraints O_i.

The algorithm follows the procedure in Fig. 8.3. In each iteration we randomly select a position for one of the windows and re-evaluate the goodness function. We update the solution if improvement was found or with probability p ($p = 0.005$ in our case). This random factor allows the algorithm to escape local maxima to find better solutions. In our evaluations, we run 2000 iterations of this algorithm to generate an initial solution, then an additional 500 iterations for a 'fine-tuning' phase, in which the pool of possible positions for each window is restricted to within 0.2 m of the current best position. The primary phase finds a 'good' layout from the whole available space and the fine-tuning phase optimizes that layout within the local maxima. The mean run-time of the procedure in the Unity framework is 3.26 s, however this time can be substantially optimized, for instance by eliminating the mesh model and by cropping to reduce the number of raycasting operations.

Random Walk Algorithm

```
Iterate(∞)
Iterate(0.2m)
```

Function `Iterate(radius r)`

```
L :=RandomSolution()
bestLayout := L
bestGoodness:= Goodness(L)
for i :=1 to numIterations do
      w := RandomWindow(W)
      position (w):=RandomPosition(PE,w,r)
      L := L ∩ w
      if Goodness(L) > bestGoodness then
            bestLayout := L
            bestGoodness := Goodness(L)
      else
            if rand < p then
                  bestLayout := L
            end if
      end if
end for
return bestLayout
```

Fig. 8.3 The random walk algorithm we use to find window layouts, similar to that of Gal et al. [19], which is based on the Metropolis-Hastings algorithm [23]

8.4.2 Generated Layouts

Some typical outputs produced by the layout generator are shown in Fig. 8.4. These outputs are generated using different possible weighting schemas of our constraint functions as shown in Fig. 8.4. Each promotes a different balance of *Adherence* and *Nonocclusion*. The Balanced condition is ideally tuned for the Shared Façades to balance both of these important yet contrasting factors in our test environments (Fig. 8.4b). Through trial and error, we found that the *Nonocclusion* constraint requires a higher weight than *Adherence* to prevent windows from frequently overlapping high salience regions, such as the area surrounding the desktop monitors in the office setting (Fig. 8.2g). Two alternative layout approaches are generated for comparison. The Constancy layout is given a *Nonocclusion* weight of zero. This theoretically causes each window to be projected onto the nearest surface in line with its default position, however the other constraints and the algorithm's random element cause some deviation (Fig. 8.4c). Conversely, the Saliency layout has an *Adherence* weight of zero. This causes windows to congregate in low salience basins of the environment's saliency map, regardless of their distance from the default location (Fig. 8.4d). However, we provide the *View-direction* function in place of constancy to help prevent windows from moving to extreme distances from the user's forward view.

For comparison, Fig. 8.5 shows several additional examples of generated layouts. These include four- and six-window layouts in both the office and living room

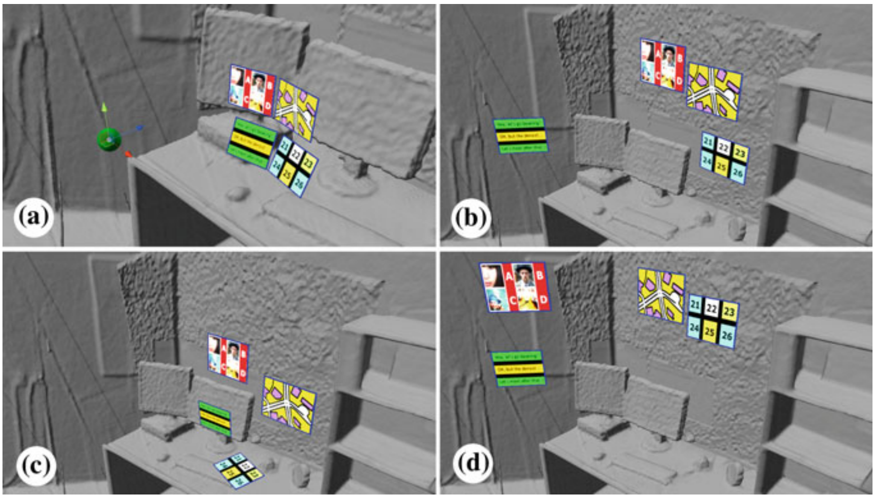


Fig. 8.4 Default window locations (**a**), set in 'floating' array 50 cm from viewing position (*green sphere*). Resulting surface layouts with constraint weights shown In Table 8.2: Balanced (**b**), Constancy (**c**) and Saliency (**d**)

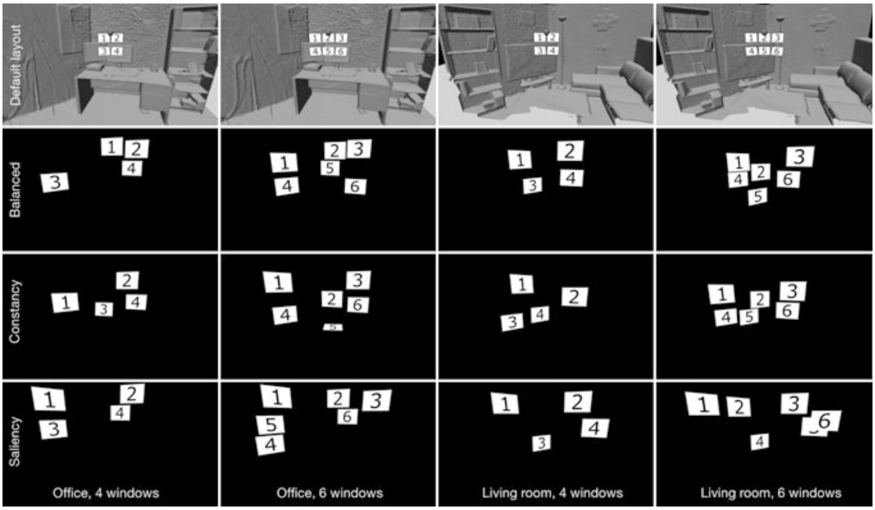


Fig. 8.5 Results of each layout weighting scheme in both environments, with layouts of 4 and 6 windows. The user viewpoint (*green sphere*) is positioned 1.5 m above the floor, 1.5 m from the wall in the office environment and 2.0 m in the living room

Table 8.2 Three possible constraint weighting schemas for the Shared Façades' layout generator. Each layout promotes a different balance of spatial constancy and visual saliency. All other weights are set to their default value of 1

Layout	Adherence	Nonocclusion	View-direction
Balanced	1	2	0
Constancy	1	0	0
Saliency	0	2	1

environments. For each combination we show one example of each constraint weighting scheme from Table 8.2. Because the layout algorithm is nondeterministic, it does not produce these same results each time it is run. However, we found in our trials that the layouts generated are relatively consistent between different runs, although layouts will occasionally show noticeable differences. For example, a window may sometimes be placed on the left side of a salient region, rather than the right. Such inconsistencies can be partially mitigated by storing layouts; once a layout is generated for a particular setting and user configuration, the same layout can be repeated in future. We evaluate these layouts in a User Study, described in the following section.

8.5 User Study

We designed a user study of the Shared Façades layout manager with two objectives: (1) to determine if the layout weighting schemes produced layouts with the intended qualities and (2) to observe real users interacting with the system through a HWD and collect qualitative feedback. To achieve these goals, we timed participants finding windows in the three layout alternatives introduced in the previous section (Balanced, Constancy and Saliency; Table 8.2) and conducted follow-up interviews.

To determine effects of the environment, we conducted the study in both of our test environments described above (Fig. 8.2a, b). We arranged these environments to contain different degrees of surface complexity (Fig. 8.2c, d) and visual salience (Fig. 8.2g, h). The *Office* was denser and more constrained, while the Living Room provided more open space.

Twelve participants (4 female, 2 left-handed, ages 18–40), volunteered for the study. All participants had normal or corrected vision and were screened for deficiency in perception of color and stereoscopy. All were regular smartphone users and none had previous experience with a HWD nor were familiar with the concept of window layout interfaces with such devices.

8.5.1 *Task and Procedure*

To probe the effects of spatial constancy and saliency in the Shared Façades' layout approach, we implemented a visual search task, a typical task for investigating the effects of a visual layout on spatial memory [13, 42, 45]. Given the generic nature of a visual search task, we postulate that if the algorithm is suited to one user, it should also scale to multiple users, searching for content. Our task was composed of two phases. In the first phase, participants scanned a pre-defined layout of six windows, arranged in a body-centric array (Fig. 8.6a) similar to that described in

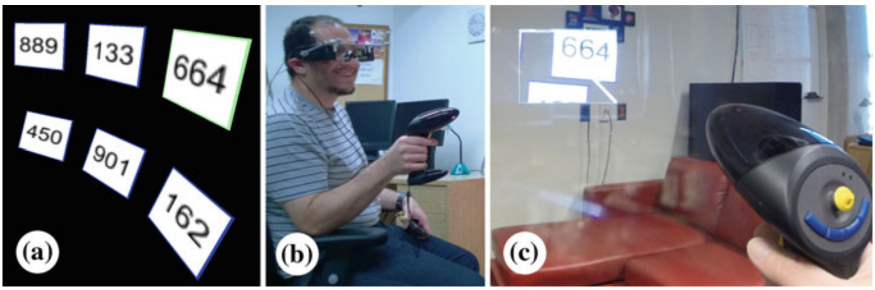


Fig. 8.6 Stimulus (**a**) and apparatus (**b**) used in the user study. The search task as viewed through the HWD (**c**)

the Personal Cockpit interface [13]. Each window showed a randomly chosen three-digit number, with the target window highlighted in light green. This initial phase allowed participants to register the contents of the target window stimulus and the window's relative location in the array. When ready, the participant pressed a button on a handheld wand (Fig. 8.6b) to begin the second phase. This triggered the Shared Façades' layout generator, after which windows reappeared on surfaces in the environment. After locating the target window the participant selected it using a virtual ray appearing to extend from the wand (Fig. 8.6c).

Participants were allowed to experiment with the interface and get familiar with the apparatus. They were given sufficient training in each environment, and adequate breaks. We asked participants to be as efficient as possible.

8.5.2 Design

The experiment used a 3 × 3 × 2 within-subjects design with the following factors:

- *Layout:* Balanced, Constancy, and Saliency
- *Room:* Office and Living Room
- *Viewing Angle:* Left, Center and Right

We set the viewing position to roughly 1.5 m from the wall in the *Office* setting, facing the desk. In the *Living Room*, the viewing position is farther back, about 2.3 m from the wall, giving a wider, more open view. For experimental validity, we controlled the user viewing position in each environment, however to prevent overly-repetitious layouts, we altered the initial Viewing Angle by increments of 30°. All factors were balanced to mitigate learning effects, with half of the participants starting in either Room. For each Room, participants completed 2 blocks of rotating Viewing Angle within each Layout. Timeouts (30 s) and incorrect selections were requeued at the end of each block, resulting in a total of 432 data points (12 participants × 3 Layouts × 2 Rooms × 3 Viewing Angles × 2 Blocks). We recorded the search time for each trial, measured from the completion of the layout calculation until the target selection.

While the search task allows us to measure layout efficiency, we followed with a second task to gauge the layout quality. In each environment, we showed participants one instance of each Layout in the *Center* Viewing Angle, this time allowing participants to take their time to explore the layout in detail. To quantify the amount of overlap with highly salient objects, we asked participants to count the number of windows covering the objects in cluttered regions of the *Office* (i.e. the desktop monitors, keyboard and mouse). We also collected responses to questions such as which windows were covering objects the participants thought were important and what they liked or disliked about the layout in general.

8.5.3 Results

Search Time—The mean search time of the 432 successful trials (excluding 4 timeouts and 4 incorrect selections) was 4.11 s (SD 3.31). Mean times for each Layout between Rooms are shown in Fig. 8.7a. We applied a log-transform on the non-normal search time data before using a univariate ANOVA. Our analysis showed a main effect of *Layout* ($F2,22 = 29.759$, $p < 0.001$). Post hoc comparisons with Bonferroni corrections showed significant differences between *Balanced* versus *Saliency* ($p < 0.001$) and *Constancy vs Saliency* ($p < 0.001$), but not *Balanced* versus *Constancy* ($p = 1.0$). *Saliency* was slowest overall (mean 5.07 s), with *Balanced* and *Constancy* taking similar times on average (3.66 s and 3.61 s, respectively).

The mean search time was greater for the *Office* (4.44 s) than the *Living Room* (3.79 s), however the effect was not statistically significant ($F1, 11 = 4.566$, $p = 0.56$). There was however a main effect of *Viewing Angle* ($F2, 22 = 11.930$, $p < 0.001$) due to differences in the facing surface complexity at different viewing angles. We also found an interaction effect between Layout and Room ($F2,22 = 5.693$, $p < 0.05$).

Overlap with salient objects—We ran Friedman's ANOVA on the reported number of windows overlapping the highly-salient region of the *Office* environment (i.e. the desktop monitors, keyboard and mouse) to look for effects of Layout. We found a significant effect ($\chi^2(2) = 20.591$, $p < 0.001$) and post hoc Wilcoxon tests showed differences between all pairs ($p < 0.05$). Mean counts are in Fig. 8.7b.

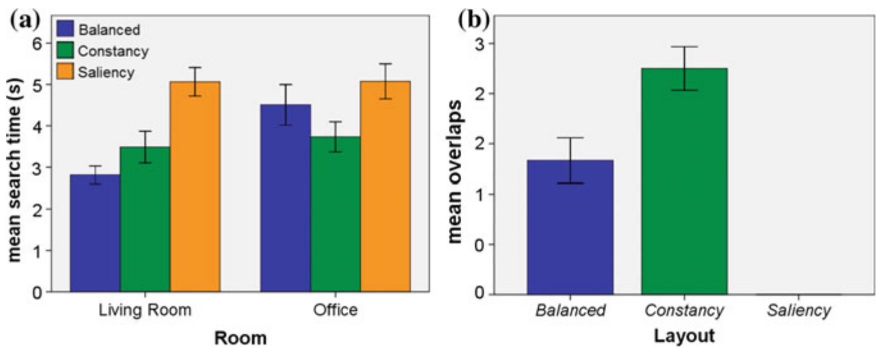


Fig. 8.7 Task time versus Layout for each environment (**a**). Average number of windows overlapping objects (desktop monitors, keyboard and mouse) in the central high-saliency region of the Office environment (**b**). *Bars* indicate ± 1SE

8.6 Discussion

In general, the different layout weightings produced the results we expected. For instance, the *Constancy* and *Balanced* layouts place windows close to their initial starting position, reducing search time. In contrast, the *Saliency* layout tries to place windows in open flat spaces with low saliency (Fig. 8.2g, h), which causes participants to engage in a prolonged search. Although the *Saliency* layout's *View-direction* constraint helps keep windows near the forward view, it is seemingly not as effective at reducing search time as the *Constancy* Layout's *Adherence* constraint.

The mean search times for the *Balanced* and *Constancy* Layouts are statistically equivalent; however we observed that the *Balanced* layout is less likely to overlap 'important' objects. Due to the nondeterministic nature of the layout generator's algorithm, windows will occasionally occlude highly salient objects, however less frequently than with the *Constancy* Layout. The results of the Freidman test on Layout support these observations, although we acknowledge the generalizability of this result is limited.

We believe the effect of Viewing Angle was due to a greater complexity in surface structure along the direction of the *Right* Viewing Angle in the test environments. More interesting, however, is the interaction effect between Layout and Room. Although the difference between Room conditions was on the outside margin of significance, we believe differences in these settings played a large role in the observed interaction effect. It seems the conflicting constraints of the *Balanced* Layout sometimes caused windows to be placed in unpredictable locations in the cluttered *Office* environment. Conversely, the avoidance of salient regions in the *Living Room* environment could be achieve with smaller window displacements and may have actually *reduced* search time by increasing legibility.

This outcome highlights a tradeoff in the dual application of saliency and constancy in the Shared Façades; while our results clearly show that spatial constancy allows windows to be found efficiently, attention to background saliency may counter the benefits of constancy in environments with a high visual density. Although users will eventually learn the window positions in any regularly-visited environment, those with an abundance of salient regions may cause some windows to be more difficult to initially locate. One possible response to this finding would be to design future interfaces with multiple modes for different scenarios. For example, one mode would boost *Adherence* to minimize window search time. An alternate mode would boost *Nonocclusion* to minimize occlusion.

Qualitative Results—Participants provided many insightful comments about the layouts and locations of individual windows. Below we summarize the general trends, which we believe will be useful for informing the design of constraints for future versions of the Shared Façade:

Important scene objects—Participants expressed many individual opinions about what objects should not be overlapped. However, there was a general consensus that the office computer equipment should not be covered, particularly the monitors.

Most participants did not like occlusions of the keyboard or mouse, although one participant felt the "keyboard is not important" because he can type by touch, while others were not concerned about the number pad. Conversely, most participants did not mind when windows covered other utilitarian or decorative objects such as the coat rack, books and wall hangings, although some noted it depends on the particular content of the item or situational context (e.g. pictures are acceptable to cover in a work environment). These results highlight the importance of customization in Shared Façade. The interface would benefit from additional knowledge about what objects are important to users.

Context—Window locations can have strong contextual associations. For instance, one participant particularly liked windows on the office desk surface, because it "fits the office paradigm". One participant considered high windows as "urgent", while a low window was "ready for the recycle bin" and windows below the desktop monitor were akin to "sticky notes". Contextual input would help the Shared Façade to determine suitable window placements.

Temporal considerations—Some window locations were not liked by participants because of anticipated future events. For instance a window covering the living room's TV or wall plugs is not ideal, because those objects might be used. Similarly, a window should not be placed directly above the couch because someone might sit there. Future improvements can enable the Shared Façades to continuously scan environments in real time, allowing such temporal considerations to be incorporated.

Relative Layout—Participants tended to prefer windows in "clusters" as opposed to being "spread out". Similarly, one participant disliked a "big gap in the middle" of the layout. A single window separated from the others was often noted as undesirable because such "outlier" windows could be "hard to find". However, separation between windows was considered acceptable if windows were in groups or even pairs. Several participants said they would prefer if windows were "lined up" with one another or with existing edges in the scene, as opposed to being "staggered". These findings reinforce the importance of aesthetic as well as functional design considerations in Shared Façades' window layouts. We look forward to its future application in different task scenarios to explore how different configurations can best suit multiple users' needs.

8.7 Manual Window Management

While Shared Façades uses an automatic layout to stabilize window locations across different environments, there will be instances where manual control is required. For instance, a user may want to place two linked data visualizations side-by-side for comparison, or carefully overlay multiple layers of a map. We implemented several manual operations to give users flexibility in managing windows. Below we describe several such operations that we implemented. However, we limit our current exploration to the operations themselves and not the interaction techniques required to invoke them. Thus for interaction we use a simple ray-casting medium with a handheld wand and the wand's embedded buttons for invoking commands.

Moving and Resizing windows—Core to the usability of all window layout managers is the ability to manually rearrange content. In our implementation, users can reposition a window by selecting it with the wand and then pointing to a new location (Fig. 8.8a). Users can similarly resize a selected window with the wand buttons. If a collision is detected given the new configuration then other windows are locally repositioned while the moved/resized window is 'pinned' in place.

Stitching and piling—Additional operations involve relations between multiple windows. Users can select two or more windows and stitch them together; selected windows move alongside the target window resize along the adjoining seam. Likewise, windows can be piled on top of a target window (Fig. 8.8b). The stitched/piled windows can subsequently be repositioned as a single object.

Saving and Restoring Configurations—Once a configuration has been manually created, users can save this configuration in the form of a body-centric array for mobile use [13]. The configuration can later be integrated into a new environment (Fig. 8.1c). Furthermore, the user can choose the appropriate layout mode (i.e. *Balanced, Constancy, or Saliency*) to suit the situation.

Orthogonality adjustment—Windows in the Shared Façades can potentially be placed on awkwardly aligned surfaces that affect legibility, such as along a long hallway or on desktops. For such situations, we implemented an operation that corrects a window's orientation, making it co-planar to the user's FoV. Selecting a

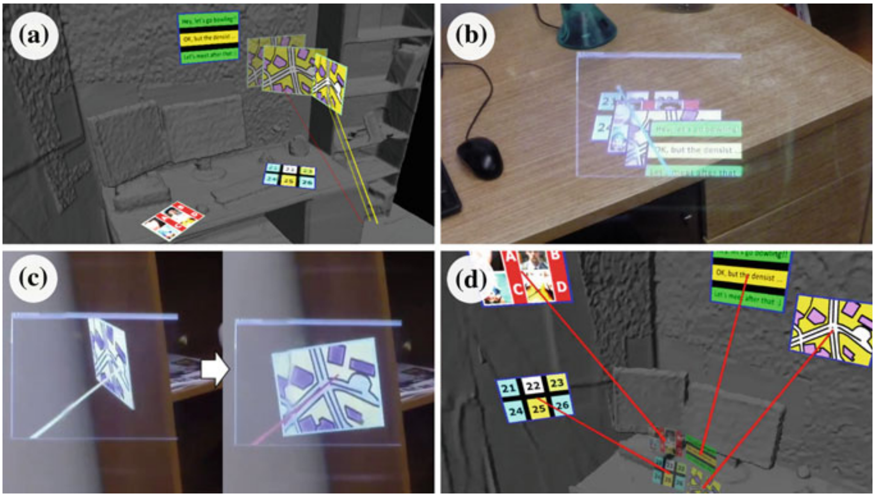


Fig. 8.8 Users can reposition windows to create custom configurations (**a**), sort windows into piles (**b**) or adjusting an awkwardly angled window for readability (**c**). If a layout makes windows hard to find, the user can see each window's relation to the default body-centric layout (**d**)

window and holding the wand trigger makes a window ‘stand-up’ (Fig. 8.8c). It realigns with the surface when the trigger is let go.

‘Space-saving’ windows—In consideration of highly cluttered environments, we can set the Shared Façades to a space-saving mode, which minimizes all windows to thumbnails. Windows remain in this form until the user holds her gaze in a window’s vicinity, at which point the thumbnail grows to the full-size of the user’s display. It returns to normal size once the user’s gaze is diverted. This mode can however be adjusted according to each users’ need. While one may user may be in ‘space-saving’ mode, another may have the layout in full-screen mode.

Help with finding windows—The environment’s structure can significantly impact the layout of windows, particularly in Saliency-based layouts. To assist with recall of windows in unfamiliar environments, we implemented visual cues that link each window’s current position to a transparent proxy in its body-centric, default position (Fig. 8.8d). This allows users to familiarize themselves with a new layout, or to re-formulate their mental model in new surroundings.

While other operations on the windows are possible, such as adding a physics engine to ‘bump’ windows and move them around as in Bumptop [1], we settled on these basic operations as they demonstrate the ability to deviate from an automated layout to provide the user with fuller control.

8.8 Summary and Future Work

Through this first exploration of the Shared Façade, we take away several lessons: (1) Spatial constancy is key to application switching efficiency in a limited-field-of-view window layout interface; (2) It is possible to strike a balance between the conflicting constraints imposed by spatial constancy and visual saliency, with the impending tradeoffs in efficiency determined by the environment’s complexity; (3) Environmental interference can be to some extent overlooked (e.g. placing a window on a partially oblique surface, occluding objects of lesser importance) to favor purely spatial concerns (e.g. avoiding large head motions, close grouping of related windows).

We acknowledge that our contributions contain several limitations. First, we have explored only two environments of a great possible variety. Also, beyond application layouts, techniques both for initiating layout operations and for interacting with window content require investigation. We acknowledge these and other drawbacks as we outline several directions for future exploration:

Computational load—Our Shared Façades prototype requires greater hardware capabilities than are typical of current wearable displays. Computational efforts are used for reconstructing the environments, estimating saliency and optimizing layouts. Nevertheless, individual components of the implementation are highly amenable to optimization. Most popular models of visual saliency are based on methods amenable to parallel computing (e.g. on GPUs). In the case of this work, there also exist highly optimized lightweight FPGA solutions [2] that imply very fast operation and low power consumption. Furthermore, there exist alternative processing models, including directly leveraging RGB-D data and foregoing the need for a mesh model. Surfaces for displaying content might be detected during geometry recovery using techniques such as Dense Planar SLAM [41] or Parallel Tracking and Mapping [27].

Future wearables and smart environments—Forthcoming developments in both display and sensor technology increase the likelihood of a wearable system becoming capable of supporting our envisioned prototype. This includes significantly broader fields of view (e.g. [32]), and mobile and miniaturized depth based sensors (e.g. Google Tango, Occipital, Microsoft HoloLens) especially suitable for portable and wearable applications. It may also be expected that our future everyday surroundings may be equipped with sensors that detect the environment's structure, internal motion and additive saliency. These environmental data could then be accessed by multiple client wearables to save device load. However, these forthcoming improvements bear implications toward our findings; in particular, device FoVs that allow increased use of peripheral vision may somewhat diminish or findings on the importance of spatial constancy.

Temporal considerations—Unlike our test environments, real-world environs are not static. For example, lighting conditions may change throughout the day, surfaces such as window blinds often move frequently and there may be people moving to and fro. These issues present additional design problems; for instance, if a passerby enters a scene, does the user prefer the window to be temporarily occluded, or to dynamically shift out of the way to remain visible? We may easily adjust our system to include additional rules, for instance to regard as salient any region where salient objects regularly appear during long-term sensing. Additional detectors, such as specific objects detectors, will allow complex semantic rules. For example, placing a clock application above a room's door may carry semantic inferences that some users are accustomed to.

Additional constraints—This work has shown how various conflicting constraints can be weighted but is given only a preliminary evaluation. In future, we would like to produce more complex layout managers that include a greater mix of constraints, such as size and shape to produce layouts that better blend into the surroundings. Exploration is also needed to determine appropriate constraints for the placement of information shared between multiple users. These developments will require thorough evaluation to determine how well the layouts meet user expectations and the given constraints.

8.9 Conclusion

We introduce the Shared Façades, a HWD interface that integrates application windows into the built environment. We identify the need for such integration to support ad hoc collaborative scenarios. Our implementation of the Shared Façades focuses on blending the principles of spatial constancy and visual saliency into a spatial window management interface. We implement an algorithm that applies these and other constraints to produce window layouts that we demonstrate in two test environments with varying visual information density. We run a user study to show the effects of different combinations of constancy and saliency in these environments. We further discuss the implications of scaling Shared Façades for multi-user environments. In summary, we successfully demonstrate layouts that provide efficient application search while also observing physical differences between user environments.

References

1. Agarawala A, Balakrishnan R (2006) Keepin' it real: Pushing the desktop metaphor with physics, piles and the pen. In: Proceedings of the CHI'06, ACM (2006), pp 1283–1292
2. Bae S, Cho YCP, Park S, Irick KM, Jin Y, Narayanan V (2011) An FPGA implementation of information theoretic visual-saliency system and its optimization. In: Proceedings of the FCCM '11, IEEE (2011), pp 41–48
3. Bell BA, Feiner SK (2000) Dynamic space management for user interfaces. In: Proceedings of the UIST '00, ACM (2000), pp 239–248
4. Bell B, Feiner S, Höllerer T (2001) View management for virtual and augmented reality. In: Proceedings of the UIST '01, ACM (2001), pp 101–110
5. Billinghurst M, Bowskill J, Jessop M, Morphett J (1998) A wearable spatial conferencing space. In: Proceedings of the ISWC '98, IEEE (1998), 76–83
6. Billinghurst M, Kato H (2002) Collaborative augmented reality. Commun ACM 45(7):64–70
7. Bouguet JY Camera calibration toolbox for Matlab: http://www.vision.caltech.edu/bouguetj/calib_doc/
8. Bruce N, Tsotsos J (2005) Saliency based on information maximization. In: Proceedings of the NIPS '05 (2005), pp 155, 162
9. Cao X, Balakrishnan R (2006) Interacting with dynamically defined information spaces using a handheld projector and a pen. In: Proceedings of the UIST '06, ACM (2006), pp 225–234
10. Cerf M, Frady EP, Koch C (2009) Faces and text attract gaze independent of the task: Experimental data and computer model. J Vis 9(12):1–15
11. Di Verdi S, Nurmi D, Höllerer T (2003) ARWin—A desktop augmented reality window manager. In: Proceedings of the ISMAR '03, IEEE (2003), pp 123–124
12. Dourish P, Bellotti V (1992) Awareness and coordination in shared workspaces. In: Proceedings of the CSCW '92. ACM (1992), pp 107–114
13. Ens B, Finnegan R, Irani P The Personal Cockpit: a spatial interface for effective task switching on head-worn displays. In: Proceedings of the CHI '14, ACM, pp 3171–3180
14. Ens B, Ofek E, Bruce N, Irani P (2015) Spatial constancy of surface-embedded layouts across multiple environments. In: Proceedings of the SUI '15, ACM, pp 65–68
15. Feiner S, MacIntyre B, Haupt M, Solomon E (1993) Windows on the world: 2D windows for 3D augmented reality. In: Proceedings of the UIST '93, ACM, pp 145–155
16. Fisher SS, McGreevy M, Humphries J, Robinett W (1986) Virtual environment display system. In: Proceedings of the I3D '86, ACM, pp 77–87
17. Fitzmaurice GW (1993) Situated information spaces and spatially aware computers. Commun ACM **36**, **7**:39–49
18. Gabbard JL, Swan JE II, Hix D (2002) The effects of text drawing styles, background textures, and natural lighting on text legibility in outdoor augmented reality. Presence: Teleoperat Virtual Environ 15(1):16–32

19. Gal R, Shapira L, Ofek E, Kohli P (2014) FLARE: fast layout for augmented reality applications. In: Proceedings of the ISMAR ’14, ACM
20. Garth B, Shoemaker D, Inkpen KM (2001) Single display privacyware: Augmenting public displays with private information. In: Proceedings of the CHI ‘10 ACM, pp 522–529
21. Grasset R, Langlotz T, Kalkofen D, Tatzgern M, Schmalstieg D (2012) Image-driven view management for augmented reality browsers. In: Proceedings of the ISMAR ‘12, IEEE, pp 177–186
22. Gupta D (2004) An empirical study of the effects of context-switch, object distance, and focus depth on human performance in augmented reality. Virginia Polytechnic Institute and State University
23. Hastings WK (1970) Monte Carlo sampling methods using markov chains and their applications. Biometrika 57(1):97–109
24. Hincapié-Ramos JD, Ivanchuk L, Sridharan SK, Irani P (2014) SmartColor: Real-time color correction and contrast for optical see-through head-mounted displays. In: Proceedings of the ISMAR ’14. ACM
25. Izadi S, Kim D, Hilliges O, Molyneaux D, Newcombe R, Kohli P, Shotton J, Hodges S, Freeman D, Davison A, Fitzgibbon A (2011) KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In: Proceedings of the UIST ‘11, ACM, pp 559–568
26. Jones BR, Benko H, Ofek E, Wilson AD (2013) IllumiRoom: peripheral projected illusions for interactive experiences. In: Proceedings of the CHI ’13, ACM, pp 869–878
27. Klein G, Murray D (2009) Parallel tracking and mapping on a camera phone. In: Proceedings of the ISMAR ’09, IEEE, pp 83–86
28. Kruijff E, Swan J, Feiner S (2010) Perceptual issues in augmented reality revisited. In Mixed and Augmented Reality. In: Proceedings of the ISMAR ‘10, IEEE, pp 3–12
29. Langlotz T, Nguyen T, Schmalstieg D, Grasset R (2014) Next-generation augmented reality browsers: rich, seamless, and adaptive. Proceedings of the IEEE 120, 2, pp 155–169
30. Lee W, Park Y, Lepetit V, Woo W (2011) Video-based in situ tagging on mobile phones. In: TCSVT 21, 10, IEEE, pp 1487–1496
31. Leykin A, Tuceryan M (2004) Automatic determination of text readability over textured backgrounds for augmented reality systems. In: Proceedings of the ISMAR ’04, IEEE, pp 224–230
32. Maimone A, Lanman D, Rathinavel K, Keller K, Luebke D, Fuchs H (2014) Pinlight displays: Wide field of view augmented reality eyeglasses using defocussed point sources of light. In: TOG 33, 4, ACM
33. Merrell P, Schkufza E, Li Z, Agrawala M, Koltun V (2011) Interactive furniture layout using interior design guidelines. In: TOG 30, 4, ACM
34. Molyneaux D, Izadi S, Kim D, Hilliges O, Hodges S, Cao X, Butler A, Gellersen H (2012) Interactive environment-aware handheld projectors for pervasive computing spaces. In: Proceedings of the Pervasive ’12, Springer, pp 197–215
35. Nacenta MA, Sakurai S, Yamaguchi T, Miki Y, Itoh Y, Kitamura Y, Subramanian S, Gutwin C (2007) E-conic: a perspective-aware interface for multi-display environments. In: Proceedings of the UIST ’07, ACM, pp 279–288
36. Orlosky J, Kiyokawa K, Takemura H (2013) Dynamic text management for see-through wearable and heads-up display systems. In: Proceedings of the IUI ’13, ACM, pp 363–370
37. Orlosky J, Kiyokawa K, Takemura H (2014) Managing mobile text in head mounted displays: studies on visual preference and text placement. SIGMOBILE Mob Comput Commun Rev 18 (2):20–31
38. Raskar R, van Baar J, Beardsley P, Willwacher T, Rao S, Forlines C (2003) iLamps: Geometrically aware and self-configuring projectors. In: Proceedings of the SIGGRAPH ’03, ACM, pp 809–818
39. Raskar R, Welch G, Cutts M, Lake A, Stesin L, Fuchs H (1998) The office of the future: a unified approach to image-based modelling and spatially immersive displays. In: Proceedings of the SIGGRAPH ’98, ACM, pp 179–188

40. Rekimoto J, Saitoh M (1999) Augmented surfaces: a spatially continuous work space for hybrid computing environments. In: Proceedings of the CHI '99, ACM, pp 378–385
41. Salas-Moreno R, Glocker B, Kelly P, Davison A (2014) Dense planar SLAM. In: Proceedings of the ISMAR '14, IEEE (in press)
42. Scarr J, Cockburn A, Gutwin C, Bunt A (2012) Improving command selection with CommandMaps. In: Proceedings of the CHI '12, ACM, pp 257–266
43. Silberman N, Shapira L, Gal R, Kohli P (2014) A contour completion model for augmenting surface reconstructions. In: Proceedings of the ECCV '14, ACM, pp 488–503
44. Szalavári Z, Schmalstieg D, Fuhrmann A, Gervautz M (1998) "Studierstube": An environment for collaboration in augmented reality. Virtual Reality 3(1):37–48
45. Tak S, Cockburn A, Humm K, Ahlström D, Gutwin G, Scarr J (2009) Improving window switching interfaces. In: Proceedings of the INTERACT '09, Springer, pp 187–200
46. Tanaka K, Kishino Y, Miyamae M, Terada T, Nishio S (2008) An information layout method for an optical see-through head mounted display focusing on the viewability. In: Proceedings of the ISMAR '08, IEEE, pp 139–142
47. Vilk J, Molnar D, Ofek E, Rossbach C, Livshits B, Moshchuk A, Wang HJ, Gal R (2014) SurroundWeb: least privilege for immersive "Web Rooms". Technical Report MSR-TR-2014–25, Microsoft Research
48. Waldner M, Grasset R, Steinberger M, Schmalstieg D (2011) Display adaptive window management for irregular surfaces. In: Proceedings of the ITS '11, ACM (2011), 222–231
49. Wilson AD, Benko H (2010) Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In: Proceedings of the UIST '10 (2010), 273–282

Chapter 9
Is It in Your Eyes? Explorations in Using Gaze Cues for Remote Collaboration

Mark Billinghurst, Kunal Gupta, Masai Katsutoshi, Youngho Lee, Gun Lee, Kai Kunze and Maki Sugimoto

Abstract According to previous research, head mounted displays (HMDs) and head worn cameras (HWCs) are useful for remote collaboration. These systems can be especially helpful for remote assistance on physical tasks, when a remote expert can see the workspace of the local user and provide feedback. However, a HWC often has a wide field of view and so it may be difficult to know exactly where the local user is looking. In this chapter we explore how head mounted eye-tracking can be used to convey gaze cues to a remote collaborator. We describe two prototypes developed that integrate an eye-tracker with a HWC and see-through HMD, and results from user studies conducted with the systems. Overall, we found that showing gaze cues on a shared video appears to be better than just providing the video on its own, and combining gaze and pointing cues is the most effective interface for remote collaboration among the conditions tested. We also discuss the limitations of this work and present directions for future research.

9.1 Introduction

If.. the principle of sight is applied to the telephone as well as that of sound, earth will be in truth a paradise, and distance will lose its enchantment by being abolished altogether.

Arthur Mee 1898

M. Billinghurst (✉) · Y. Lee
Empathic Computing Laboratory, University of South Australia,
Adelaide, Australia
e-mail: mark.billinghurst@unisa.edu.au

K. Gupta · G. Lee
The HIT Lab NZ, University of Canterbury, Christchurch, New Zealand

M. Katsutoshi · K. Kunze · M. Sugimoto
Keio University, Minato, Tokyo, Japan

Y. Lee
Mokpo National University, Mokpo, South Korea

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_9

Twenty years after the invention of the first telephone, writing in the Strand Magazine, journalist Arthur Mee imagined a future where people would be able see and hear each other remotely, so distance would be abolished altogether [30]. In the hundred years since, Mee's vision has largely been realized, with a wide variety of communications technology devoted to enabling people to work together while being far apart.

The first commercial video conferencing services such as AT&T's PicturePhone [32] required fixed terminals and wired connections. In contrast, current mobile phones and wearable technologies allow users to connect wirelessly from wherever they are. Applications like Skype [37], or Google Hangouts [20], make seeing and talking to a remote person as easy as pressing an icon on a screen.

The development of new technologies such as interactive surfaces, handheld tablets, and head worn computers, among others, have changed the nature of remote collaboration, creating more opportunities for shared experiences. The first video conferencing systems focused on sharing a view of a person's face and creating the illusion that a distant person was sitting across the table [21]. More recently, head worn cameras can be used to enable a remote user to see the wearer's point of view and their task space in front of them.

The use of this "Task Space" video conferencing [5] can be especially useful for remote assistance on a physical task, enabling a remote expert to see the workspace of the local worker and provide feedback. For example, a worker trying to repair a broken machine, or perform a medical procedure. The head worn camera (HWC) allows the remote expert to see what the local user is doing, while a head mounted display (HMD) can allow the remote expert to provide Augmented Reality (AR) virtual cues overlaid on the local user's view of the real world to help them complete the task.

Previous research has found that wearable systems can significantly improve performance and co-presence on a wide range of physical tasks such as machine operation [25], bicycle repair [23], and robot construction [14], among others. For example, in remote maintenance, workers using an AR interface were able to reduce their task performance time by up to 30 % [17]. However, a HWC may have a wide field of view and so it may be difficult for the remote expert to know exactly where the local user is looking, impacting communication. In this chapter we explore how this problem can be addressed by using head mounted eye-tracking to convey gaze and attention cues.

Although eye-tracking technology has been available for decades it is only recently that small wearable eye-trackers have been developed. So there has been little research conducted on how eye-tracking can be used in head mounted remote collaboration systems. The work presented here makes several important contributions; (1) it provides two of the first examples of combining head worn eye-trackers with HMD for remote collaboration, (2) it presents experimental results showing the impact of sharing gaze cues, and (3) describes some novel designs for systems that share non-verbal cues in remote collaboration.

In the remainder of the chapter we first summarize earlier related work (Sect. 9.2), then present two prototypes developed (Sect. 9.3), and two experiments conducted with the prototypes (Sects. 9.4 and 9.5). Next we discuss the implications of the experimental results (Sect. 9.6) and give some design guidelines, before ending with a conclusion and directions for future work (Sect. 9.7). The research presented in this chapter will be useful for people conducting work in the field of remote collaboration and wearable technologies.

9.2 Related Work

Our research builds on earlier work in the areas of communication models, remote collaboration, gaze tracking, and emotion detection. Combining this enables us to develop novel head worn systems for remote assistance that allow users to share gaze and other non-verbal cues. In this section we review related research in each of these areas.

9.2.1 Communication Models

In face-to-face conversation people use a variety of verbal and non-verbal cues to communicate with one another. A number of models have been developed to conceptualize how people send, receive and process information. For example, Berlo's SMCR model describes communication in terms of a Source, Message, Channel, and Receiver [2]. Some of these models have been applied to teleconferencing to predict the impact of technology on remote collaboration, e.g. Social Presence [19], Grounding [8], and Media Richness Theory [10]. For example, current video conferencing systems cannot convey the spatial cues typically used in face-to-face conversation and so people may compensate by using more verbal cues. Whittaker provides an excellent review of communication theories applied to remote collaboration [42].

One of the most popular communication models is Clarke and Brennan's Grounding model [8]. This is based on the idea that people communicating interactively work together to exchange information to achieve common ground or shared understanding. Clark and Marshall [9] show that one of the sources of common ground is physical co-presence, and especially having a shared visual space. Fussell has shown how the Grounding model can be applied in wearable teleconferencing systems [15], and that shared views of the workspace can maintain situational awareness, and promote sense of co-presence. Further research has shown the importance of sharing remote pointing or gesture cues for conversational grounding in task space collaboration [22]. However, other non-verbal cues such as gaze and facial expression are also important in face-to-face conversation, but these cues are not clearly transmitted in current task space collaboration systems.

9.2.2 Remote Collaboration

In our research we are interested in developing collaborative systems that help people perform tasks in the real world, so a shared view of the local worker's task space is particularly important. SharedView [25] was one of the earliest examples of using a HWC and HMD for remote collaboration. A factory worker wore a camera on their head and streamed video to a monitor watched by a remote expert, who could point out important features on the video with their hand gestures. A camera was used to capture the expert's gestures and show them back into the worker's HMD. So the worker could easily show the remote expert what they were working on, and get visual feedback to help them. In a user study, this system enabled the local worker to complete a task faster than with a fixed camera.

Since then many people have explored the use of head worn systems for remote task space collaboration. Fussell et al. [12] combined a HWC and HMD to enable a local worker to share their view with a remote expert. They conducted a number of user studies, finding that using shared video caused people to complete tasks together more slowly than face to face [12], had a significant impact on conversation compared to an audio only connection [23], and could improve performance compared to an audio-only connection [14]. One of the limitations of these systems was that the remote expert could not provide visual feedback, but Bauer et al. [1], and others [16, 33], showed how remote pointing could be added to significantly improve collaboration.

More recent work has shown that computer vision techniques can be used to fix virtual pointing and drawing cues on the video of the real world, providing AR cues that further enhance collaboration [18]. Depth sensors can also be used to capture the remote expert's hands and allow them to appear as AR virtual cues in the local workers view, providing support for very natural remote collaboration with rich communication cues [40]. Comparing between using a HMD and handheld display for remote collaboration with AR cues, local workers felt than the HMD was more useful because it allowed them to have both hands free [22].

Head worn cameras can share a local worker's view with a remote helper, but what the remote expert can see is fixed to the local worker's viewpoint. One way to overcome this is to use body worn cameras [24] that can be moved independently and allow the remote helper to have an independent view. Researchers have also explored providing the remote expert with a 3D virtual model of the local worker's environment that they can freely look around [38]. A second limitation is that the remote expert doesn't know exactly where the local user is looking in the shared view. Previous research has shown that having a different field of view into a shared workspace can significantly affect task performance [26]. This could be addressed by using gaze tracking to clearly show where the local worker is looking, but there has been little research on this.

9.2.3 Wearable Gaze Tracking

In face-to-face conversation gaze provides information about where the person is directing his or her attention. So gaze could also be an important cue in remote collaboration. There has been some research on the use of eye-tracking in desktop teleconferencing. Brennan et al. [3] and others [6, 28] found that sharing gaze between two remote collaborators significantly improved performance on desktop visual search tasks, compared to audio only communication. Velichkovsky [41] has found similar benefits for remote problem solving, and Li et al. [27] report that two-way sharing of gaze information using a desktop GUI application facilitated coordinated behaviour. Finally, Muller et al. [31] found that sharing gaze in a remote puzzle solving task using a desktop GUI enabled people to solve the puzzle significantly faster than with speech alone. Overall, it seems that gaze provides an excellent cue for inferring user attention, helping with the grounding process and improving remote collaboration.

However this work was with GUI systems and there has been little research on using head worn eye-tracking and remote collaboration. One of the few systems that explore this is the work of Fussell et al. [13, 14] who developed a system with a HMC with an attached eye-tracker. This sent video of the local worker's workspace to the remote expert's monitor along with the local workers' eye gaze details. However, the local worker was not wearing a HMD, so the remote expert was not able to provide virtual cues to help them. They found that performance with a HMC with eye gaze information was not significantly different than using audio only, and performance was significantly worse than with a fixed camera showing the entire workspace. However, in a later experiment, Ou et al. [34] found that the focus of attention can be predicted from monitoring eye gaze in a remote collaboration task, and eye-tracking provides a benefit.

Overall this shows that tracking gaze is a useful measure of attention in desktop systems and may improve remote collaboration, but there is significant research that still needs to be done for head worn systems.

9.2.4 Emotion and Non-verbal Feedback

Finally, our work also explores the transmission of non-verbal cues in remote collaboration systems. Inspired by Picard's Affective Computing [35], researchers have developed systems that detect emotion through facial expression, voice, body language and physiological cues [7]. Detecting facial expression in a wearable system is difficult, but we previously developed a system that does this in an unobtrusive manner using photo reflective sensors on an eyeglass frame [29]. However most work in this area is for recognizing a single user's emotional display and not in a shared application.

There has been some earlier research showing that sharing physiological data can improve remote collaboration. Tan et al. [39] developed a desktop video conferencing system that shared heart rate, galvanic skin response and respiratory rate. Sharing physiological data significantly increased the positive affect score compared to audio only conferencing, to the same level of the video condition. However, this research used a desktop computer with no support for remote task space shared viewing, and the remote expert couldn't see the local users view.

9.2.5 Summary

There have been previous head worn systems that support remote collaboration, demonstrating that sharing a remote view with virtual annotations on that view can significantly improve collaboration. However, there is little research that has investigated eye-tracking in head worn systems for remote collaboration. The existing research on gaze-tracking has mostly used desktop interfaces where user studies have found benefit from sharing gaze cues. We could find no research that combines a HMC with a HWD and an eye-tracker for remote collaboration. So our research fills an important gap in the research literature.

9.3 Prototype Systems

In order to explore how augmented cues could be used for head worn collaboration we have developed several prototypes for testing. In this section we describe two of our earliest systems.

9.3.1 Hardware Prototype One

The first prototype system combined: (1) a head mounted eye-tracker, (2) a HWC, and (3) a HMD. The HMD was the Brother AirScouter [4], an optical see-through monocular display with an 800 by 600 pixel resolution and 22.4° field of view. For the eye tracker we used the Microsoft Lifecam HD 5000 (with the IR filter removed and an IR LED added) with a custom printed 3D enclosure pointing at the user's eye. Finally, a Logitech Webcam C920 camera was mounted facing outwards as the HWC. The eye-tracking camera was connected to a computer running open source eye tracking software from Pupil Labs [36]. This tracks the eye pupil in the video stream from the HD 5000 webcam and maps it over the video from the HWC. All of the components were mounted together on a single frame (Fig. 9.1).

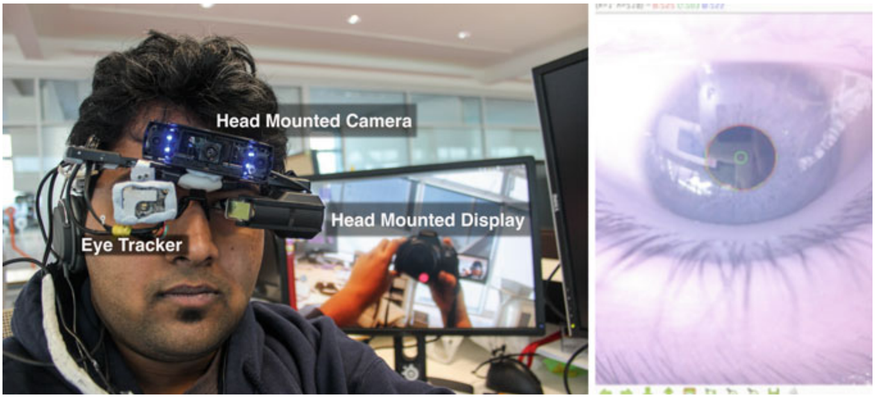


Fig. 9.1 Hardware Prototype One and eye tracking view

9.3.2 Hardware Prototype Two

A second prototype reduced the size and bulk of the first prototype and added the ability to convey facial expression cues. This combined three hardware systems; (1) the Pupil eye-tracker [32], (2) the Epson Moverio BT-200 HMD [11] and (3) the AffectiveWear facial expression tracker [29] (see Fig. 9.2).

The Pupil eye-tracker has a camera to track the user's right eye and a HWC to capture the user's view. The camera views are sent via USB to a computer running the eye tracking software, tracking gaze at a 120 Hz with 0.6° accuracy. The BT-200 is a stereo optical see-through HMD with 960 by 540 pixel resolution and 23 degree field of view. The AffectiveWear module uses eight photo reflective sensors mounted on the right lens of the BT-200 display with the sensor output used to detect the distance to the user's skin surface, and so their facial expression. The data obtained from the sensors is classified by an SVM based machine-learning algorithm into one of four facial expressions: neutral, positive, negative, and surprise, with a recognition accuracy of over 90 % [29].

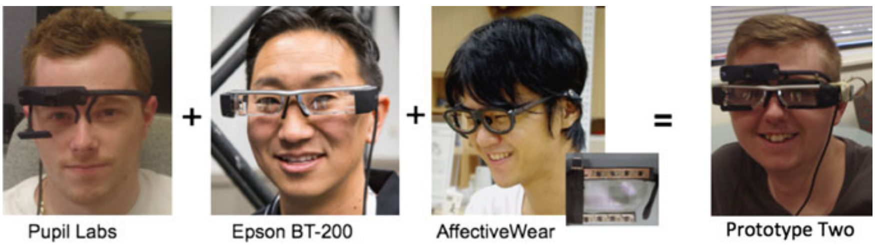


Fig. 9.2 Systems combined together for Prototype Two

9.3.3 Remote Expert View Interface

Both hardware prototypes were connected to a computer that performed eye-tracking and showed gaze information superimposed as a green circle on the HWC live video view. This was the remote expert user interface. For Prototype One, software was written that allows the user to move a red pointer circle on the screen using their mouse (Fig. 9.3a). This screen view was then connected back to the Brother HMD, allowing the person wearing the display to see the remote expert's pointing gestures.

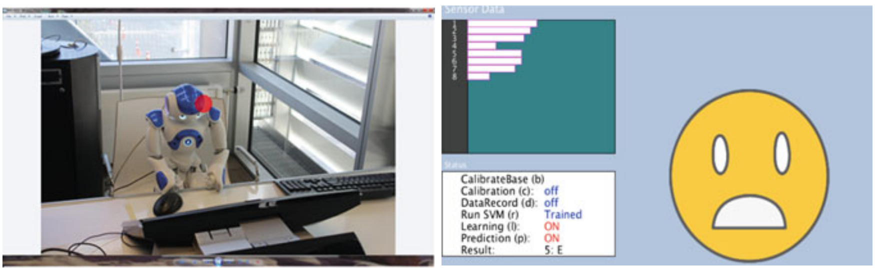
The Prototype Two interface consisted of two monitors. One showed the live camera view from the HMD and supported remote pointing and screen sharing back to the BT-200 HMD, the same as Prototype One. A second screen showed a representation of the HMD user's facial expression and the output from the AffectiveWear sensors (Fig. 9.3b).

Using this hardware and software, the remote expert could see what the local worker was doing, where he or she was looking (and also his/her facial expression in Prototype Two), and provide feedback using a virtual pointer to help with their task. The next two sections report on experiments conducted with the prototypes.

9.4 Experiment One: Gaze Tracking

The first experiment was conducted with Prototype One and was designed to explore the effect of providing gaze and pointer attention cues on remote collaboration. The main hypotheses were:

- H1: There will be a significant difference in performance time between conditions that provide additional attention cues and those that don't.



(a) Remote expert view with pointer (b) Facial expression feedback

Fig. 9.3 Remote expert view interfaces

- H2: There will be a significant difference in co-presence measures between conditions that provide additional attention cues and those that don’t.

The additional cues were gaze information from the HMD user (Local Worker) and pointer feedback from the Remote Helper.

9.4.1 Experiment Setup

The experiment involved a remote collaboration task in a room with the Local Worker on one side and the Remote Helper on the other, and a divider between them so they couldn’t see each other (Fig. 9.4). The Prototype One HMD used by the Local Worker, was connected to a computer, with the monitor placed on the Remote Helper’s side, so they could both see the same task space video. The participants could freely speak with each other to complete the task. All of the experimental conditions were using the same physical interface setup.

The task involved the Remote Helper helping the Local Worker to construct three-dimensional LEGO structures. We created four different structures each with 17 pieces, with a similar level of difficulty checked through a pilot test (Fig. 9.5). We counterbalanced which structure assigned to which condition to further reduce the effects of the structures used in each condition. The Remote Helper had a manual showing how the pieces should be assembled and used speech (and pointing gestures if available) to communicate with the Local Worker. Although the task is simple, it contains many of the spatial manipulation steps required in more complicated industrial applications, such as object selection, rotation and translation. Similar tasks using toy bricks have previously been used in papers related to remote task space collaboration (e.g. [12, 24]).

We had two independent variables, (1) POINTER—whether a virtual pointer was available to the Remote Helper, and (2) EYETRACKER—whether sharing gaze information from the Local Worker. We used a 2 × 2 design with each of these variables either present or absent, for a total of four conditions (Table 9.1).

(a) Remote Helper (b) Local Worker

Fig. 9.4 Participants performing the experimental task

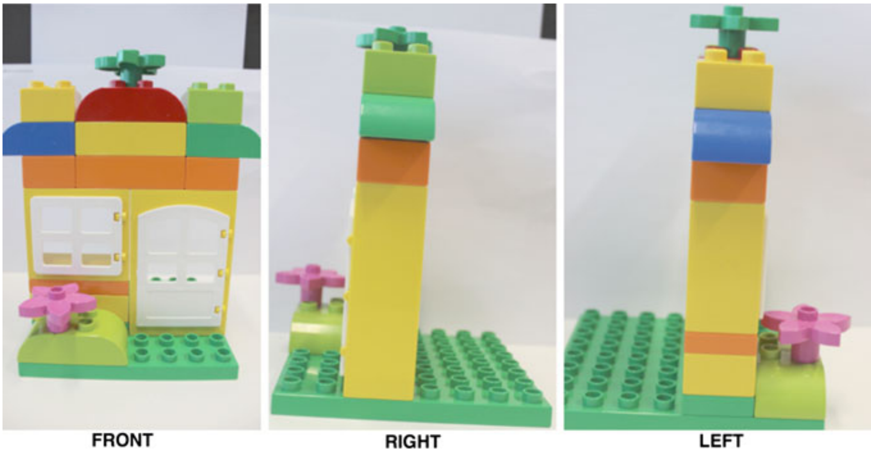


Fig. 9.5 Example of a typical Lego model

Table 9.1 Experimental conditions

	EYETRACKER: No	EYETRACKER: Yes
POINTER: No	No cue (NONE)	Eye tracker cue (E)
POINTER: Yes	Pointer Cue (P)	Both cues (BOTH)

A within-subject design was used where the participants experienced all conditions in an order under a Balanced Latin Square design that counterbalanced the carryover effects between conditions. Participants were recruited from the university. On arrival, they were randomly assigned as Local Worker or Remote Helper, and asked to practice building the LEGO structure in face-to-face collaboration without using the prototype system. The Remote Helper was provided with a manual and asked to assist the Local Worker.

Subjects were then separated to sit at their desks and perform the tasks using the provided interface in each condition. The Remote Helper first built the model following the manual to make sure he or she understood the instructions before helping the Local Worker. The time taken for the Local Worker to complete the model was recorded, and both participants completed a questionnaire with 7-level Likert scale rating items after each condition (Table 9.2). These questions are based on the questionnaire from [22] which was successfully used to measure co-presence and communication in an earlier remote collaboration experiment. After finishing all of the conditions, participants ranked the interface conditions based on various criteria.

Table 9.2 Likert scale rating questions (1 = Disagree, 7 = Agree)

Q#	Statement
Q1	I felt **connected** with my partner
Q2	I felt I was **present** with my partner
Q3	My **partner** was able to **sense my presence**
Q4	LW: My partner (*For Remote Helper:* I) **could tell when** I (*For Remote Helper:* my partner) **needed assistance**
Q5	I **enjoyed** the experience
Q6	I was able to **focus** on the task activity
Q7	I am **confident** that we **completed the task correctly**
Q8	My partner and I **worked together** well
Q9	I was able to **express** myself **clearly**
Q10	I was able to **understand partner's message**
Q11	Information from partner was helpful

9.4.2 *Experiment Results*

A total of 26 subjects (13 pairs) completed the experiment, 20 male and 6 female, with ages ranging from 21 to 33 years old. All of them had good English.

Both the pointer and eye tracking cues helped participants perform the task significantly faster (Fig. 9.6). On average participants took less time to complete the task in E (M = 245.7, SE = 17.2) and P (M = 234.5, SE = 20.7) conditions compared to the baseline NONE condition (M = 258.3, SE = 19.6). Overall participants completed the task fastest while using both the P and E together (M = 200.5, SE = 14.1), saving about 22 % of time compared to the NONE condition. A repeated measure two-way ANOVA ($\alpha = 0.05$) found a significant main effect of both P ($F_{1,12} = 4.91$, $p < 0.05$) and E ($F_{1,12} = 5.81$, $p < 0.05$) factors on the average time taken to complete a task. No significant interaction was found between the pointer (P) and eye-tracker (E) factors ($F_{1,12} = 0.57$, $p = 0.47$).

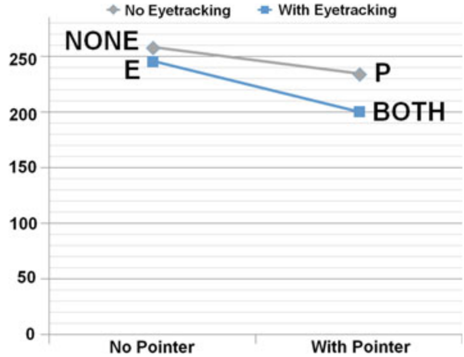


Fig. 9.6 Average task performance time (s)

The pointer cue significantly improved the perceived quality of communication, collaboration, and co-presence. For factorial analysis of the Likert scale rating results, we used the Aligned Rank Transform for non-parametric factorial analysis using ANOVA procedures ($\alpha = 0.05$) [43]. Where we wanted to investigate pair-wise comparison between conditions, we used Wilcoxon Signed Rank tests. See Table 9.3 for the results. There was a significant main effect of the POINTER factor in Q1–Q4 (Co-Presence questions) and Q9–Q11 for the Local Worker participants, but not for Q5–Q8. For the Remote Helper participants, a significant main effect of the POINTER factor was found in Q1–Q4 and Q8–Q11, but not for Q5–Q7. This shows having a pointer significantly affected both the Local Workers' and the Remote Helpers' subjective quality of communication and co-presence (Q1–Q4 and Q8–11) but not enjoyment (Q5), focus (Q6), or self-confidence (Q7).

The eye tracking significantly improved the communication and collaboration quality, and sense of being focused. The EYETRACKER factor had a significant main effect for the Local Workers on Q1–Q4, Q6, and Q8–Q11, but no significant effect on Q5 or Q7. For the Remote Helpers, the EYETRACKER had a significant

Table 9.3 ART Repeated measure two-way ANOVA on rating questions (significant results in bold, L = Local Worker, R = Remote Helper)

Q#	User	Pointer		EYETRACKER		Interaction	
	L/R	$F_{1,\ 12}$	p	$F_{1,\ 12}$	p	$F_{1,\ 12}$	p
Q1	L	9.763	**0.009**	8.220	**0.014**	8.291	**0.014**
	R	15.096	**0.002**	17.153	**0.001**	6.052	**0.030**
Q2	L	5.086	**0.044**	14.153	**0.003**	0.205	**0.**659
	R	9.412	**0.010**	7.926	**0.016**	5.781	**0.033**
Q3	L	6.858	**0.022**	9.179	**0.010**	1.185	**0.**298
	R	22.511	**<0.001**	4.648	0.052	8.359	**0.014**
Q4	L	8.372	**0.013**	19.761	**0.001**	5.501	**0.037**
	R	9.828	**0.009**	3.434	0.089	8.715	**0.012**
Q5	L	0.011	0.917	4.195	0.063	0.436	0.521
	R	2.753	0.123	5.589	**0.036**	0.230	0.640
Q6	L	0.321	0.581	5.334	**0.040**	0.260	0.619
	R	4.110	0.065	1.423	0.256	0.589	0.458
Q7	L	0.161	0.695	4.248	0.062	3.636	0.081
	R	2.585	0.134	2.500	0.140	2.678	0.128
Q8	L	0.386	0.546	7.303	**0.019**	1.581	0.233
	R	5.172	**0.042**	0.777	0.395	9.060	**0.011**
Q9	L	6.381	**0.027**	17.388	**0.001**	3.275	0.095
	R	13.119	**0.004**	14.944	**0.002**	0.059	0.813
Q10	L	14.690	**0.002**	14.739	**0.002**	1.389	0.261
	R	9.273	**0.010**	18.381	**0.001**	1.523	0.241
Q11	L	11.766	**0.005**	9.946	**0.008**	4.206	0.063
	R	7.531	**0.018**	18.036	**0.001**	1.937	0.189

main effect on Q1, Q2, Q5, Q9–Q11, but not for Q3, Q4, and Q6–Q8. This reveals that the EYETRACKER had significant effect on communication quality and co-presence (Q1–Q4 and Q8–Q11), but also on other aspects such as sense of being focused for the Local Workers (Q6) and enjoyment for the Remote Helpers (Q5).

After finishing, participants ranked the conditions with respect to different criteria (1 = the best, 4 = the worst). Table 9.4 shows the list of the ranking criteria.

The condition with both of the visual cues (BOTH), was ranked the best in most aspects, while the NONE condition was ranked worst. Figure 9.7 shows the average results of ranking questionnaire. In most cases the BOTH condition was ranked in first place, while the baseline condition (NONE) was ranked last. The other conditions (P and E) were mostly ranked in-between with no clear distinction between them.

To test for significantly difference, we used Friedman tests ($\alpha = 0.05$). The Remote Helpers ranked the conditions significantly different in all criteria (C1: $\chi^2(3) = 17.031$, $p = 0.001$; C2: $\chi^2(3) = 19.062$, $p < 0.001$; C3: $\chi^2(3) = 12.046$, $p = 0.007$; C4: $\chi^2(3) = 24.969$, $p < 0.001$; C5: $\chi^2(3) = 18.256$, $p < 0.001$; C6: $\chi^2(3) = 15.738$, $p = 0.001$).

With the Local Workers, only C2 ($\chi^2(3) = 22.907$, $p < 0.001$), C5 ($\chi^2(3) = 22.256$, $p < 0.001$), and C6 ($\chi^2(3) = 24.535$, $p < 0.001$) showed significant difference in ranking, while the rest of the criteria showed no significant difference (C1: $\chi^2(3) = 3.185$, $p = 0.364$; C3: $\chi^2(3) = 4.302$, $p = 0.231$; C4: $\chi^2(3) = 7.031$, $p = 0.071$). This indicates that while POINTER and EYETRACKER affected the Remote Helpers in all aspects, they affected Local Workers in terms of the quality of fluent communication (C2: feeling connected, C5: feeling their partner knew their needs, C6: understanding their partner's message), but not on enjoyment (C1), focus (C3), or sense of presence (C4).

When asked in which condition they could better understand where their partner was focusing, almost 85 % (11/13) of the Local Workers preferred the condition in which the pointer feature was available (i.e. P and BOTH) as it was helping them to understand their partner's focus and clear instructions. One participant said: "*With the Pointer, I can relate to what he is talking about, because I could understand him more.*" In the case of the Remote Helpers, around 70 % (9/13) chose the BOTH condition since they were able to see the place where their partner was looking and also point on the video. One participant said: "*The Eye Tracker helps me to look in the same view of my partner, and I know what he is doing and will do next*".

Table 9.4 Ranking questions (1 = best, 4 = worst)

No.	Ranking criteria
Which condition was the best…	
C1	at helping you to **enjoy** the task?
C2	at making you feel **connected** with your partner?
C3	at helping you stay **focused** on the task?
C4	at making you feel that you were **present** with your partner at the same workspace?
C5	for you (or the partner) to **know** that the partner (or you) **needed assistance**?
C6	at helping you **understand** the partner's message?

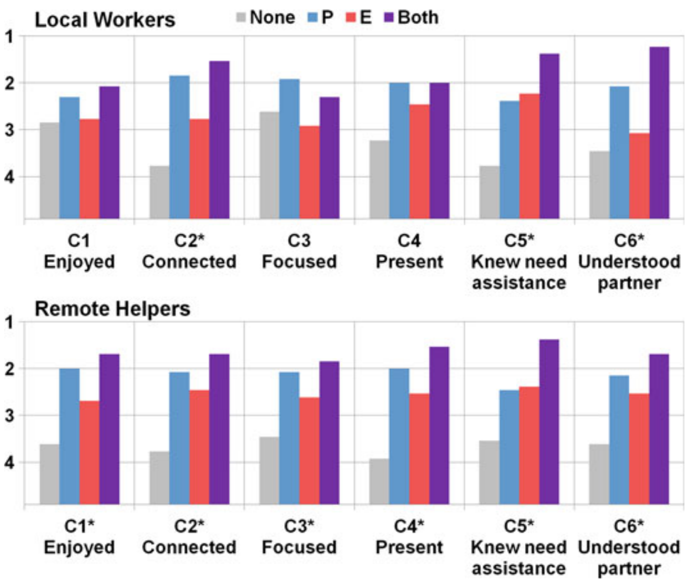


Fig. 9.7 Average ranking results (* = significant difference)

When asked which condition helped them perform the task efficiently and quickly, 77 % (10/13) of Local Workers said the BOTH condition was the best. Most Remote Helpers (85 %, 11/13) also picked the BOTH condition as the best. One of the Local Worker said, "*Eye tracker was giving my partner more information about where I looked at, while the pointer was for giving me the instruction from my partner, where I should look at and which piece I should take*".

We also recorded the user conversation and counted the number of phrases said (Fig. 9.8). One recording was lost, so we analyzed the data from 12 pairs of participants. We found that sharing visual cues made the conversation more

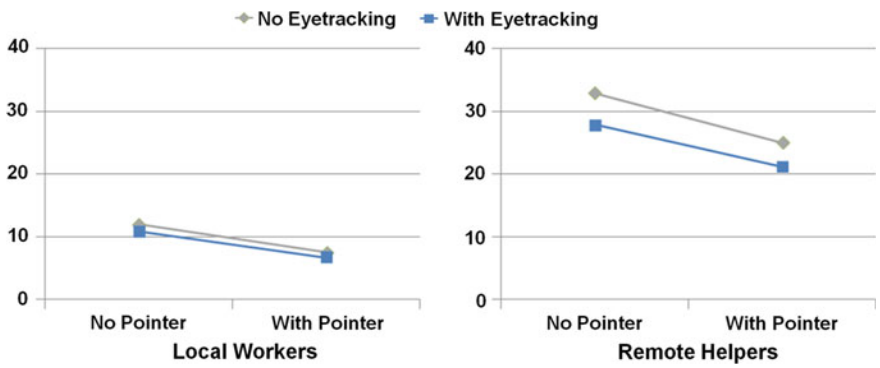


Fig. 9.8 Results of conversation analysis (*number of phrases*)

efficient, changed the wording in deictic expressions, and helped subjects feel connected.

Overall, Remote Helpers spoke more than Local Workers, and use of a pointer reduced the amount of communication needed. A two-way repeated measure ANOVA showed a significant effect of POINTER on the number of phrases said by both Local Worker ($F_{1,\,11} = 6.532$, $p = 0.027$) and Remote Helper ($F_{1,\,11} = 8.479$, $p = 0.014$). No statistically significant effect of EYETRACKER (Local Worker: $F_{1,11} = 0.538$, $p = 0.479$; Remote Helper: $F_{1,11} = 3.057$, $p = 0.108$) nor interaction between the two factors (Local Worker: $F_{1,11} = 0.012$, $p = 0.914$; Remote Helper: $F_{1,11} = 0.175$, $p = 0.683$) were found.

The Remote Helper's role was to instruct the Local Worker, so the Remote Helper's mostly led the conversation and talked more (three times more phrases; $F_{1,94} = 103.004$, $p < 0.001$) than the Local Workers. The Local Workers mostly said short phrases to confirm that they understood the instructions (e.g. "*Okay*", or "*Yes*") or to ask the Helper to confirm if what was done was correct (e.g. "*Here?*" or "*Like this?*").

We also noticed the Remote Helpers using different words to identify objects or specify directions. Without the pointer, participants identified objects in terms of colour, shape, or size (e.g. "*green piece*", "*square one*"). With the pointer, participants simply said "*this one*" while pointing at the object. A similar pattern was observed for directions, such as participants saying "*put it here*" or "*next to this*" while using the pointer. In the conditions with no pointer, participants used words that describe directions (e.g. "*in front of your hand*", "*move right*", "*more left*").

9.4.3 Discussion

The results support hypothesis H1, namely that the task performance time was significantly improved by using both the pointer and eye tracking cues. This is because the Remote Helper could use the pointer to give direct guidance to the Local Worker, and clearly disambiguate speech commands. Similarly, gaze cues showed the Local Worker's focus of attention to the Remote Helper, making it easier for them to collaborate. This pointer result agrees with earlier work on the value of using visual cues for remote collaboration [1, 16, 33], but the gaze results show a benefit compared to earlier wearable eye-tracking [13, 14].

Interestingly, the user interface for the local worker in the EYETRACKER condition (E) was the same as in the condition NONE (i.e. the instructions from helper to worker were only in verbal form). However, participants still took less time to complete the task than in the condition NONE, and they rated the condition E higher in most of the questions asking about the quality of the communication. This shows the benefit of making the remote helper aware of the local worker's focus of attention. Just by seeing where the local worker was looking helped the remote user to be more efficient in the instructions given and take less time to complete the task.

The subjective survey results supported hypothesis H2, with the answers to the co-presence questions (Q1–Q4) showing that both the Local Workers' and Remote Helpers' sense of being present together was improved significantly by both of the visual cues. The eye-tracking cue not only influenced the quality of the communication (Q1, Q9, Q10, Q11, C2, and C6) but also other aspects of the user experience. The Remote Helpers said they enjoyed the task more when the eye-tracking cue was present (Q5 and C1). On the other hand, Local Workers reported they were able to focus more on the task (Q6) when the eye-tracking cue was shared. These results agree with the results from the Local Workers saying that they felt their partner could tell when they needed assistance better when the eye tracking cue was present (Q4 and C5), hence the Remote Helpers disturbed the Local Worker less, and the Local Workers were able to focus more on the task.

9.5 Experiment Two: Gaze Tracking and Facial Expression

Prototype Two was a less bulky version of Prototype One, with the addition of the AffectiveWear sensors for detecting facial expression. In this section we report on an initial pilot study exploring using gaze tracking, pointing and facial expression sensing in remote collaboration. We tested four different interface conditions;

- **V:** A video only condition, in which the remote user can only see video from the local user.
- **P:** Video plus pointer condition, which is the same as the V condition with the addition of gaze cues and pointing on the HWC video.
- **E:** Video plus expression condition, which is the same as the V condition with the addition of the facial expression monitor.
- **A:** All condition that adds both pointing, gaze tracking and facial expression monitoring to the V condition.

The main hypothesis was:

- H1: There will be a significant difference in collaboration measures between conditions that provide additional expression cues and those that don't.

9.5.1 Experiment Setup

The Local Worker wore the HMD and sat at a table, while a Remote Helper sat next to the Local worker, separated by a divider, and watching the output from the HWC and sensors on two monitors (Fig. 9.9). Before starting, the Local Worker had a calibration completed for eye gaze and facial expression settings.

The Local Worker had real wooden blocks on the table in front of him or her, and the task was for them to work with the Remote Helper to create a 2D picture out

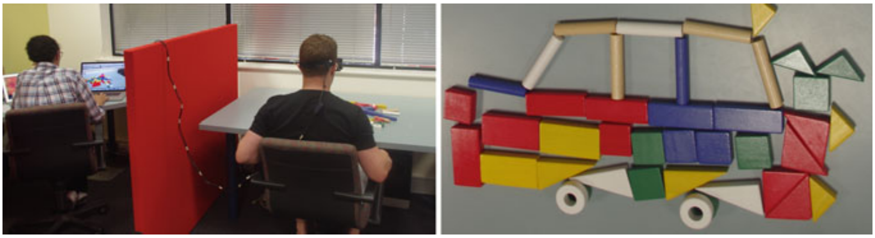


Fig. 9.9 Experimental setup and a sample picture created (Car)

of the blocks, based on a topic given by the experimenter, such as car, castle, animal, etc. The Remote Helper did not have a manual showing how to build the picture, so both participants collaborated together equally. This is similar to previous tasks that have been validated in the literature (e.g. [12, 24]).

The subjects used each of the four different interface conditions with a different target picture for each condition. The order of the conditions and the picture topics were counterbalanced to reduce any order effects. Subject pairs were given five minutes to construct a model for each condition and were told to use as many of the blocks as possible. After each condition subjects were asked a number of Likert scale questions about how well they thought they worked together, could understand each other, and communicated together, etc. (see Table 9.5). These questions are a slightly modified version of the questions asked after the first experiment, with additional questions about recognizing user feelings. After all the conditions they were asked to rank each interface in order according to how well they communicated with their partner, and worked together, etc.

9.5.2 Experiment Results

A total of 5 pairs of subjects (6 men, 4 women) completed the pilot test with ages ranging from 20 to 45 years old. Overall, subjects had no trouble completing the object construction task in the 5 min allocated.

Table 9.5 Likert scale rating questions (1 = Strongly disagree, 7 = Strongly agree)

Q#	Statement
Q1	My partner and I worked well together on the task
Q2	It was easy to be aware of what my partner was doing
Q3	I felt connected with my partner
Q4	My partner and I communicated together well
Q5	I understood how my partner was feeling
Q6	My partner understood how I was feeling
Q7	I was satisfied with the output of the task

Using a Friedman test, there was no significant difference in the average Likert scale scores for each of the conditions. Table 9.6 shows the average score for each of the questions across all the conditions, separated into the Remote Helper and Local Worker scores. The Pval column contains the p-value scores from the Friedman test. As can be seen, non of the p-value scores approach significance ($p < 0.05$) which is not surprising given the low number of participants in the user study.

However there was a significant difference in the ranking questions (Fig. 8.11). Subjects were asked to rank the four conditions in order from best (1) to worst (4) in response to the following questions: Which condition did you:

(Q1) work best with your partner,
(Q2) feel that you communicated best with your partner,
(Q3) feel that you understood best how your partner was feeling.

Figure 9.10 shows the average rankings for each condition for each of these questions (1 = best, 4 = worst). It is interesting to note that for Q2 almost all of the subjects ranked the A condition the best, while for Q1 the Remote Helpers felt that they worked best with their partner in conditions that allowed them to point (A, P).

A Friedman test found a significant difference between rankings by the Local Workers for Q2 ($\chi 2(3) = 8.3$, $p < 0.05$)) and near significance for Remote Helpers ($\chi\ 2(3) = 7.3$, $p = 0.06$)). Similarly there was a significant difference between rankings by the Local Workers for Q3 ($\chi\ 2(3) = 8.3$, $p < 0.05$)) and for Remote

Table 9.6 Average results from the rating questions (Pval = Friedman p-value)

	Remote helper					Local worker				
Q	V	P	E	A	Pval	V	P	E	A	Pval
#1	4.8	5.2	5.0	6.0	0.71	5.0	5.2	4.4	5.6	0.24
#2	5.0	4.8	5.6	5.4	0.75	4.0	5.0	3.6	5.0	0.66
#3	4.2	5.0	4.8	4.8	0.87	4.2	5.0	4.0	5.2	0.40
#4	4.6	5.2	4.6	5.6	0.74	5.2	5.2	4.8	5.0	0.80
#5	3.8	4.6	4.6	4.6	0.56	3.4	3.8	3.4	4.2	0.53
#6	3.0	4.0	3.8	4.0	0.50	3.8	4.6	3.8	4.4	0.80
#7	4.8	5.4	4.8	5.8	0.91	5.2	4.6	4.6	6.2	0.30

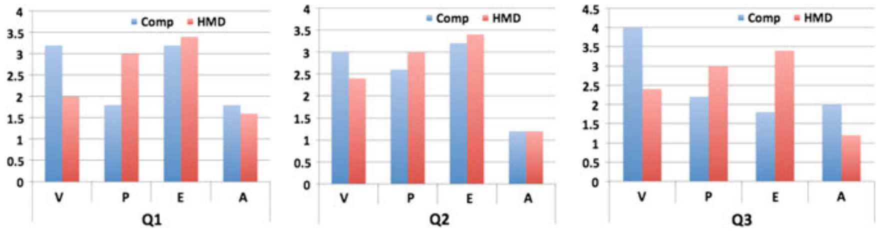


Fig. 9.10 Average rankings for each of the ranking questions (Comp = Remote helper, HMD = Local worker)

Helpers (χ 2(3) = 9.2, $p < 0.05$)). Finally there was a near significant difference in results for Q1 for the Local Workers (χ 2(3) = 6.4, $p = 0.09$)).

After each experiment subjects were interviewed to collect their opinions. They felt that the remote pointing was very useful and so the A and P conditions were their favourites. For the Remote Helper, the pointer made them feel more connected to their partner. One Local Worker said that the remote pointer felt like "*There was a second pair of eyes helping me with the task*". Many users ranked the Expression condition (E) as the worst because, as one user said, "*I couldn't point and didn't know what to do.*" Although some Remote Helpers said that monitoring the facial expression was useful because it enabled them to see if the Local Worker understood what they were saying. Subjects felt that the gaze tracking was helpful for detecting if the Local Worker was looking at the objects being talked about, although some Remote Helpers felt that they could know what blocks the Local Worker was going to manipulate by looking at the Local Worker's hands.

9.5.3 Discussion

In this case hypothesis H1 was partially supported. There was no difference in the subjective Likert scale scores across conditions. In observing how the subjects worked together, the Remote Helper mostly kept their attention focused on the screen showing the live video view from the Local Worker, and was often not aware of the facial expression information shown. The facial expression detection hardware also gave a wide range of responses when the Local Worker was talking and their face moved through various poses. So the face expression was often only occasionally seen, and when it was, the results were often inaccurate. In the ranking results the E condition on average was ranked worst on questions Q1 and Q2, relating to working and communicating with your partner.

Despite this, the ranking scores of the condition A with gaze, pointing and facial expression cues were significantly better than any of the other conditions in terms of how well the subjects felt that they could communicate together (Q2). This may be because of the different communication channels offered by each modality. One Remote Helper stated "*I ranked the A condition best, because I could easily point to communicate, and when I needed it I could check the facial expression to make sure I was being understood*".

Just as with Experiment One, the pointing cue was found to be valuable by the Remote Helpers as this allowed them to clearly indicate objects and feel connected to their partner. One Remote Helper said, "*When I was pointing I felt like there was something for me to do*". Most of the Remote Helpers used the pointing cue a lot, especially when giving deictic commands such as "*Move the block there*".

Although these pilot results are encouraging there are significant areas that could be improved in the future. The current Remote Helper interface has separate screens for the facial expression presentation and the live camera view, forcing them to divide their attention between two locations. In the future this should be combined into a single user interface. The facial expression tracking does not work well when the user is talking, requires careful calibration, and should show a wider range of facial expressions. Finally, the overall system is still bulky and uncomfortable to wear for long periods of time.

9.6 Lessons Learned

Although these results are preliminary, it seems that there are several key lessons that may be useful for others conducting research in this area:

- Eye-tracking can be used to provide valuable attention cues
- Remote Helpers should be given the ability to share pointer cues
- Providing pointing and gaze cues increases feelings of co-presence
- It is important to support for bi-directional visual communication

One of the most interesting implications is that eye-tracking could be used to change the nature of remote collaboration. In earlier systems, (e.g. [12, 25]), the Remote Helper was able to watch the Local Worker's view and see when they were directly interacting with objects in the workspace. The Local Worker often had to explicitly reach for an object for the Remote Helper to know that they were about to interact with it. However, adding gaze-tracking means that the Remote Helper can be aware of the Local Workers implicit intentions before they physically start interacting, because they will look at objects before selecting them. This means that the Remote Helper can now be aware of both implicit and explicit communication cues, and respond accordingly. For example, telling the Local Worker that they are looking at the wrong object before they reach out to pick it up, or drawing their attention to a part of the workspace they many not have seen.

A second implication is that providing gaze cues alone can significantly improve remote collaboration even without supporting pointing from the remote helper. Current eye-tracking hardware is small enough to place inside HMDs and so there could be a great opportunity for developers wanting to create intuitive systems for remote collaboration. The Experiment One results showed that using eye-tracking alone was seen to be just as beneficial as using remote pointing by itself.

Finally, these studies imply that providing rich communication cues like gaze, pointing gesture and non-verbal facial expressions can play a very important role in creating a sense of co-presence and creating deeper understanding between users. However, there is more work that needs to be done with face expression capture and representation before it is as useful as pointing and gaze.

9.7 Conclusions and Future Work

In this research we have explored how eye-tracking can be added to head worn systems for task space remote collaboration. We have developed two prototypes that enabled remote collaborators to share gaze, pointing and facial expressions. Experimental tests with these systems have indeed shown that there is some benefit from sharing gaze cues. Systems with shared pointer and gaze cues enable users to work together more quickly, feel more co-present, communicate together better, and are preferred over other interface combinations without gaze and pointing.

However there are many areas of potential future research. The current user studies are with simple block arrangement tasks that don't require much collaborative problem solving. In the future it would be good to explore tasks that require more negotiation or complex 3D spatial manipulation over a large area. In this case awareness of the user's attention will be more important and so may illustrate an even greater benefit to sharing gaze information. It will also be important to conduct further user studies with a larger number of participants and wider range of subjective and objective measures.

Another area that could be explored is providing symmetric communication cues between both participants. In our prototype we had gaze information being sent from the Local Worker to the Remote Helper, but the Local Worker didn't have any idea of where the Remote Helper was looking. In the future we could provide both users with a similar interface so that we could track the gaze of the Remote Helper and also send this back, making a fully symmetric system.

Gaze information is just one of many non-verbal cues that could be shared between participants. Our efforts at sharing facial expression were not entirely successful due to the limitations of the technology used. It would be interesting to explore the impact of sending other non-verbal cues such as heart rate, galvanic skin response or EEG. These could be used to infer levels of stress, cognitive load or other physiological states.

Finally, one of the most important areas of research will be to explore the current communication models and to extend them to take into account the additional gaze and non-verbal cues being transmitted. The current theoretical models take these cues into account, but there have been few studies that have empirically tested them and modified them to match what happens in real conditions.

References

1. Bauer M, Kortuem G, Segall Z (1999) Where are you pointing at? A study of remote collaboration in a wearable videoconference system. In: Proceedings of ISWC, vol 99, pp 151–159
2. Berlo DK (1960) The process of communication. Holt, Rinehart and Winston Inc, New York
3. Brennan SE, Chen X, Dickinson CA, Neider MB, Zelinsky GJ (2008) Coordinating cognition: the costs and benefits of shared gaze during collaborative search. Cognition 106:1465–1477

4. Brother Air Scouter. http://www.brother.co.uk/ Accessed April 10th 2016
5. Buxton W (1992) Telepresence: Integrating shared task and person spaces. In: Proceedings of graphics interface, vol 92, pp 123–129
6. Carletta J, Hill RL, Nicol C, Taylor T, De Ruiter JP, Bard EG (2010) Eyetracking for two-person tasks with manipulation of a virtual world. Behav Res Methods 42(1):254–265
7. Calvo RA, D'Mello S (2010) Affect detection: an interdisciplinary review of models, methods, and their applications. IEEE Trans Affect Comput 1:18–37
8. Clark HH, Brennan SE (1991) Grounding in communication. In: Perspectives on socially shared cognition, vol 13, pp 127–149
9. Clark HH, Marshall CE (1981) Definite reference and mutual knowledge. In: Joshi AK, Webber BL, Sag IA (eds) Elements of discourse understanding. Cambridge University Press, Cambridge, pp 10–63
10. Daft RL, Lengel RH (1986) Organizational information requirements, media richness and structural design. Manage Sci 32(5):554–571
11. Epson Moverio website. http://www.epson.com/moverio/ Accessed 10 April 2016
12. Fussell SR, Kraut RE Siegel J (2000) Coordination of communication: effects of shared visual context on collaborative work. In: Proceedings of the 2000 ACM conference on Computer supported cooperative work. ACM, pp 21–30
13. Fussell SR, Setlock LD (2003) Using eye-tracking techniques to study collaboration on physical tasks: implications for medical research. Unpublished manuscript, Carnegie Mellon University
14. Fussell SR, Setlock LD, Kraut RE (2003) Effects of head-mounted and scene-oriented video systems on remote collaboration on physical tasks. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, pp 513–520
15. Fussell SR, Setlock LD, Yang J, Ou J, Mauer E, Kramer AD (2004) Gestures over video streams to support remote collaboration on physical tasks. Hum-Comput Interact 19 (3):273–309
16. Gergle D, Kraut RE, Fussell SR (2013) Using visual information for grounding and awareness in collaborative tasks. Hum–Comput Interact 28(1):1–39
17. Gauglitz S, Lee, C., Turk, M., Höllerer, T. (2012). Integrating the physical environment into mobile remote collaboration. In: Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services, pp 241–250
18. Gauglitz S, Nuernberger B, Turk M, Höllerer T (2014) World-stabilized annotations and virtual scene navigation for remote collaboration. In: Proceedings of the 27th annual ACM symposium on user interface software and technology. ACM, pp 449–459
19. Gunawardena CN, Zittle FJ (1997) Social presence as a predictor of satisfaction within a computer-mediated conferencing environment. Am J Dist Educ 11(3):8–26
20. Google Hangout https://hangouts.google.com/ Accessed 10 April 2016
21. Ishii H, Kobayashi M, Arita K (1994) Interactive design of seamless collaboration media. Commun ACM 37(8):83–98
22. Kim S, Lee G, Sakata N, Billinghurst M (2014) Improving co-presence with augmented visual communication cues for sharing experience through video conference. In: 2014 IEEE International symposium on mixed and augmented reality (ISMAR). IEEE, pp. 83–92
23. Kraut RE, Fussell SR, Siegel J (2003) Visual information as a conversational resource in ollaborative physical tasks. Hum-Comput Interact 18(1):13–49
24. Kurata T, Sakata N, Kourogi M, Kuzuoka H, Billinghurst M (2004) Remote collaboration using a shoulder-worn active camera/laser. In: Eighth international symposium on wearable computers, 2004. ISWC 2004, vol 1. IEEE, pp 62–69
25. Kuzuoka H (1992) Spatial workspace collaboration: a Shared View video support system for remote collaboration capability. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 533–540
26. Lanir J, Stone R, Cohen B, Gurevich P (2013) Ownership and control of point of view in remote assistance. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 2243–2252

27. Li J, Manavalan M, D'Angelo S, Gergle D (2016) Designing shared gaze awareness for remote collaboration. In: Proceedings of the 19th ACM conference on computer supported cooperative work and social computing companion (CSCW '16 Companion). ACM, New York, NY, USA, pp 325–328
28. Litchfield D, Ball LJ (2011) Using another's gaze as an explicit aid to insight problem solving. Q J Exp Psychol 64(4):649–656
29. Masai K, Sugiura Y, Ogata M, Kunze K, Inami M, Sugimoto M (2016) Facial expression recognition in daily life by embedded photo reflective sensors on smart eyewear. In: Proceedings of the 21st international conference on intelligent user interfaces. ACM, pp. 317–326
30. Mee A (1898) The pleasure telephone. The Strand Magazine:339–345
31. Müller R, Helmert JR, Pannasch S, Velichkovsky BM (2013) Gaze transfer in remote cooperation: Is it always helpful to see what your partner is attending to? Q J Exp Psychol 66 (7):1302–1316
32. Noll AM (1992) Anatomy of a failure: picturephone revisited. Telecommun Policy 16 (4):307–316
33. Oda O, Sukan M, Feiner S, Tversky B (2013) 3D referencing for remote task assistance in augmented reality. In: 2013 IEEE symposium on 3D user interfaces (3DUI). IEEE, pp 179–180
34. Ou J, Oh LM, Fussell SR, Blum T, Yang J (2008) Predicting visual focus of attention from intention in remote collaborative tasks. IEEE Trans Multim 10(6):1034–1045
35. Picard RW, Picard R (1997) Affective computing, vol 252. MIT press, Cambridge
36. Pupil Labs website: https://pupil-labs.com/. Accessed 10 April 2016
37. Skype website: https://www.skype.com/. Accessed 10 April 2016
38. Tait M, Billinghurst M (2015) The effect of view independence in a collaborative AR system. Comput Support Coop Work (CSCW) 24(6):563–589
39. Tan CSS, Luyten K, Van Den Bergh J, Schöning J, Coninx K (2014) The role of physiological cues during remote collaboration. Presence: Teleoperat Virtual Environ 23 (1):90–107
40. Tecchia F, Alem L, Huang W (2012) 3D helping hands: a gesture based MR system for remote collaboration. In: Proceedings of the 11th ACM SIGGRAPH international conference on virtual-reality continuum and its applications in industry. ACM, pp 323–328
41. Velichkovsky BM (1995) Communicating attention: Gaze position transfer in cooperative problem solving. Pragmat Cogn 3:199–222
42. Whittaker S (2003) Theories and methods in mediated communication. The handbook of discourse processes, pp 243–286
43. Wobbrock JO, Findlater L, Gergle D, Higgins JJ (2011) The aligned rank transform for nonparametric factorial analyses using only ANOVA procedures. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 143–146

Part II
Case Studies and Applications

Chapter 10
Usage of Interactive Event Timelines in Collaborative Digital Tabletops Involving Automation

Y.-L. Betty Chang, Stacey D. Scott and Mark Hancock

Abstract Tabletop computers are increasingly being used for complex, collaborative scenarios, such as emergency response. In such scenarios, maintaining situation awareness of dynamic changes automated by the system is crucial for users to make optimal decisions. If the system does not provide users with appropriate feedback, they can become confused and "out-of-the-loop" about the current system state, leading to suboptimal decisions or actions. To enhance situation awareness of dynamic changes occurring in the collaborative tabletop environment, we designed an interactive event timeline to enable exploration of historical system events. We conducted a user study to understand how various design alternatives of interactive event timelines impacted situation awareness in the context of a cooperative tabletop game. Our initial results showed that, on average, all groups had a high combined level of situation awareness, regardless of the given timeline designs. To better understand what role the timelines played for the groups, we conducted an in-depth video analysis. Participants used the timelines mostly for perceiving new changes by interacting with the detailed information. The analysis also revealed the benefits of the high-level information presented in the timelines for projecting future system states. The information presented in the timelines was considered as the correct historical account and was used to negotiate the knowledge of automated changes. We also report on how other system features, in addition to the timelines, were used for situation awareness maintenance. Finally, we discuss implications for designing interactive event timelines for co-located collaborative systems involving automated events.

Y.-L.B. Chang (✉) · S.D. Scott
Games Institute/Systems Design Engineering, University of Waterloo,
Waterloo, ON, Canada
e-mail: betty.chang@uwaterloo.ca

S.D. Scott
e-mail: stacey.scott@uwaterloo.ca

M. Hancock
Games Institute/Department of Management Sciences,
University of Waterloo, Waterloo, ON, Canada
e-mail: mark.hancock@uwaterloo.ca

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_10

10.1 Introduction

There is a growing interest in using digital tabletops for co-located group activities that involve complex, often dynamically changing data. Given their ability to provide digital functionality for collaborative work while allowing for face-to-face communication, tabletop interfaces have been proposed for many domains, such as crisis and disaster management [6] and commercial maritime operations [5]. In such domains, decision-makers' awareness of the system state is crucial to the quality of decisions made [11]. However, when the system automates changes in system states but does not provide appropriate feedback via the user interface, human operators are left "out-of-the-loop" [22]; that is, they are unable to keep up with system changes. They may be confused, and are unable to make optimal decisions and interact when needed. As digital tabletop applications become more sophisticated and begin to incorporate more automation to manage the type of complex data inherent to many real-world application domains, keeping users in-the-loop becomes an essential design requirement.

Tabletop applications cannot assume that users will attend to and notice all system changes due to a variety of potential distractors, for example, conversing with collaborators at or near the tabletop, attending to devices being used in conjunction with the tabletop (e.g., a smart phone or tablet), or even being called away temporarily. Consequently, a change occurring on the tabletop (automated, or made by another user) can be easily missed. Existing tabletop applications that incorporate dynamic data provide little to no support of situation awareness maintenance, such as displaying historical system data. Instead, they focus on novel interfaces for sharing or collaborating with the current, real-time view of the system state [1, 4].

Interactive event logs and timelines have been previously shown to reduce response time and improve decision accuracy for single-user applications involving automated system changes [21, 36]. We were interested in adapting such timelines to a co-located collaborative context on tabletop systems. We investigated two design factors: *control placement* (number of timelines for a group of users and timeline placements) and *feedback location* (where to display interaction feedback of timelines). We sought to understand how these two design factors impacted situation awareness of dynamic changes in collaborative tabletop applications.

We evaluated the design factors in the context of a popular three- to four-player collaborative tabletop board game, Pandemic.[1] This game requires intense strategy discussions, resource management, and advance planning of actions to prevent epidemic outbreaks. Moreover, Wallace et al. [47] found that their digital tabletop version of the Pandemic game elicited the aforementioned out-of-the-loop automation problem, due to the amount and complexity of changes as well as the fact that players were not constantly paying attention to the tabletop interface.

Our study involved two phases. In Phase 1, a controlled experiment tested the two design factors by asking participants to play three short partial games in which

[1]The Pandemic game was published by Z-Man Games, used with permission.

they used three different timeline alternatives. In Phase 2, participants completed a full game from start to finish using a configurable version of the timeline that allowed them to utilize any combination of the *control placement* and *feedback location* at any time.

The results from a detailed analysis of Phase 1, previously published in Chang et al. [3], revealed that more timeline interactions were encouraged with replicated timelines, where each player had a copy in their personal spaces. More timeline interactions correlated with higher levels of individual situation awareness. Despite individual's differences in situation awareness, groups were overall found to have high combined levels of situation awareness in all conditions. In this article, we expand on this prior work by presenting the results of an in-depth video analysis of players' situation awareness maintenance behaviour in Phase 2. The analysis provides a better understanding of how system features, including the timelines, were used to understand dynamic changes driven by the system. The analysis revealed that the timelines were useful as both static and interactive visualizations, and they were mostly used to investigate recent dynamic changes automated by the system. The timelines were used only occasionally to strategize and prioritize tasks while another system feature, the discard pile, was used primarily for this purpose.

We first contextualize our research by presenting the related work on both situation awareness and workspace awareness. Next, we present the conceptual design of our interactive event timelines and discuss previous work that motivated the timeline design alternatives. We then introduce the Pandemic game case study and describe our timeline designs. Next, we present the study method, research questions, and results, focusing on the video analysis results of timeline usage based on Phase 2 of the study. Finally, we discuss the implications of our findings on timeline designs and the limitations of this work, and we conclude with future work.

10.2 Related Work

There has been substantial research on the concept of awareness and its many forms in the Human Factors and Human-Computer Interaction literature e.g., [32, 37]. In this section, we provide an overview of related research, specifically in the areas of situation awareness, team situation awareness, and workspace awareness. In Sect. 10.3, we discuss related work in digital tabletops to motivate the design factors studied.

10.2.1 Situation Awareness

Situation awareness (SA) describes a person's awareness of the environment, and has been applied to many domains, including military combat [10], aviation [42], and nuclear plant operation [2]. Endsley [11] defined SA as the perception of changes in the system state (level 1), comprehension of the changes (level 2), and projection of future system states (level 3). The second level of SA requires people to connect multiple pieces of knowledge (level 1) to infer their meaning and form an understanding of the perceived changes. The third level describes the ability to predict future states of the system based on the person's understanding (level 2).

The phenomenon of *change blindness* [31] is a key cause of deficient SA in automated systems, as observed in the automation literature [7]. Change blindness refers to a person's inability to recognize changes in the environment after interruption or deviation in attention [31]. The interruption recovery literature has explored the use of persistent, interactive information displays to mitigate change blindness and to rapidly improve SA following an interruption to the task in systems with dynamically changing data [36, 39, 41].

Sasangohar et al. [36] studied interactive event timelines that allowed users to highlight historical events on a main task display (a map) located on a large wall display by interacting with event bookmarks, which were displayed on a graphical timeline located on a secondary handheld display. Their results showed that the timeline allowed people to quickly gain awareness of missed events and helped reduce recovery time while improving decision accuracy after interruptions. They argued that the interactive event timeline provided a "simplified representation of important events [that] facilitated the quick encoding of perceptual information and minimized the visual search" [36:1155]. On a large digital tabletop interface, promoting SA while minimizing visual search across the entire interface is an important design goal. Our work thus applies this interactive event timeline concept to digital tabletops involving automation.

Previous research has largely focused on the design of awareness displays to support individuals' SA [21, 36, 39]; our research expands on this by applying interactive event timelines to multi-user tabletop environments.

10.2.2 Team Situation Awareness

As we aim to support team environments where users have a shared goal, examining *individual* SA of the system may not be sufficient to understand the collaborative process of maintaining SA and strategizing as a group.

Team situation awareness (TSA) is the team members' overlapping knowledge of the situation as well as the full SA required for individuals to successfully coordinate actions and complete the shared goal [35]. Previous work generally agrees that TSA requires high levels of individual SA and communication among team members [8, 13, 48]. Much of the research in TSA has focused on individual tool design (i.e. to facilitate individual SA) and analysis of communication and coordination behaviours to provide design implications and create advanced measurements [8, 13, 29, 35].

Theoretical models of TSA have also been developed [8, 34, 35]. However, they tend to focus on high level processes [8, 35], such as shared goals, communication, team members' background, and teamwork. This project examined how specific system features were leveraged by our participants to maintain and communicate SA.

10.2.3 Workspace Awareness

Extensive research has shown the value and the information richness provided by the objects, people, and environment in co-located collaborative settings [15, 18, 30]. While SA focuses on a person's knowledge of a system's state, workspace awareness describes a person's knowledge of their collaborators and their actions within a shared (physical or virtual) workspace [30].

In distributed settings, workspace awareness has been supported through techniques such as virtual embodiment (e.g., telepointers [15], virtual arms [45], avatars [25]). Although a significant amount of workspace awareness information can be gained "for free" in a co-located tabletop setting through observation [15], the distance between collaborators and the complexity of some tabletop interfaces can hinder people's ability to observe all activities, especially for interactions happening in collaborators' personal spaces [38]. We aimed to preserve users' workspace awareness while providing them with interactive event timelines to maintain SA. Thus, we tested two factors relevant to the design of timelines for supporting workspace awareness in the context of tabletop systems, and we discuss these factors in Sect. 10.3.2.

10.3 Awareness Support for Tabletop Systems Involving Automation

Traditionally, automation has used to reduce manual workload or mechanically change the states of physical materials [28]. Now, automation is also used to reduce mental workload, and may involve changing the state of virtual objects, such as automatically updating data visualizations based on underlying sources. These automated changes can also negatively impact situation awareness, often due to change blindness or state changes not being displayed. As illustrated in Fig. 10.1a, people at a digital tabletop can be unaware of a change occurring in the system interface due to the large size of the display or other competing demands for their attention, such as conversing with a teammate. Moreover, even when a change occurs within a person's field of view, they may still miss the change due to limited attentional capacity.

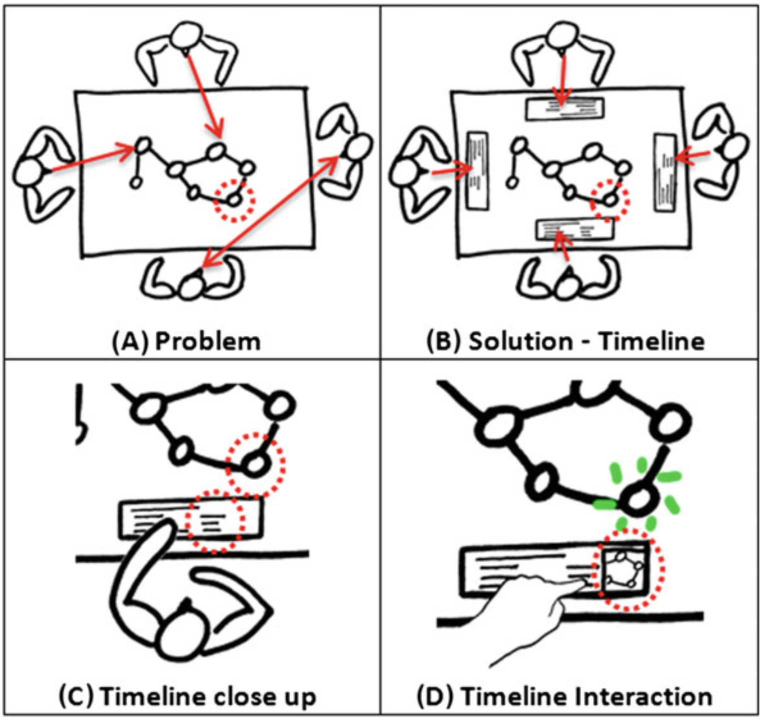


Fig. 10.1 Conceptual design of the interactive event timeline. **a** Problem: users can miss automated changes if their attention is elsewhere (*red arrows* show attentional focus). **b** Solution: timelines provide a way for users to view and explore changes. **c** New changes are appended to the timeline, and **d** users can interact with the timelines to locate the changes on the shared area (highlighted node) and on the timeline (graph cut-out on the right of the timeline)

10.3.1 Situation Awareness Support for Tabletop Systems

Substantial tabletop research has investigated interaction techniques for digital object manipulation, menu invocation, information sharing, and tangible interaction [23, 43, 46, 49]. For collaborative tabletop applications, significant work has also been done on information visualization, coordination and collaboration styles, and control widgets [20, 26, 44]. As more sophisticated tabletop applications are developed to support complex task domains [1, 5, 6], application tools that allow maintenance of awareness of dynamic changes will become essential. To date, no such tools have been tested in collaborative tabletop environments, and existing tabletop applications involving dynamic data focus on supporting current, real-time views of the system state in collaborative work [1, 4]. This work is a first step towards addressing this gap.

10.3.2 Conceptual Design

To address the issues introduced by the use of automation in digital tabletop systems, we explored using interactive event timelines to provide persistent information of historical system events. Such timelines also provide the information in a visual form that can fit within a person's field of view, despite the large size of the table. To gain awareness of the current system state, a person can examine and explore the timeline, which provides an overview of historical events (Fig. 10.1b, c). To get more in-depth information, they can invoke further feedback on the shared display or on their personal areas (Fig. 10.1d). Based on the existing literature, we considered two key factors in designing these timelines: *control placement* and *feedback location*.

10.3.2.1 Control Placement

The event timeline is a visualization of historical events as well as a control for invoking detailed information of the automated changes. It was unclear how to distribute and place the timelines to best support workspace awareness and situation awareness in a group setting.

Morris et al. [26] compared providing individual replicated system controls around the border of a tabletop system with a single, shared control in the centre for a collaborative photo tagging application. They found that while individual controls were preferred, the groups were more collaborative (i.e., more labels per image) when using the shared controls. This result suggests that a shared timeline may contribute to more collaborative work and improved team situation awareness (e.g., joint investigation for all team members). However, it is unclear how well shared timelines support individual situation awareness since users need to coordinate their use of the timelines.

Ha et al. [16] compared direct touch and mouse pointers for a two-player competitive image search game on digital tabletops, and their results show that the direct touch condition allowed for higher levels of workspace awareness and resulted in quicker response to opponents' moves. Nacenta et al. [27] studied five different interaction techniques for selecting, moving, and rotating images for two collaborative tasks: an image sorting game and a storyboarding activity. They similarly found that the interaction technique requiring explicit input in the shared space (i.e., drag-and-drop) allowed for easier tracking of collaborators' actions and helped avoid conflicting actions. While participants may have higher workspace awareness using the shared control, it was unclear how individual versus shared timelines would impact participants' situation awareness. Providing replicated timelines allows each user to view and manipulate the timeline for the purpose of maintaining situation awareness. As the current research still lacks understanding in how the placement of timelines impacts users' situation awareness, we examined the control placement factor.

10.3.2.2 Feedback Location

Another design consideration is where to provide the visual feedback related to a user's exploration of historic system events. Information about the event can be displayed locally (on the timeline) or on the shared area of the tabletop. These design alternatives may better facilitate either individual control or group function, respectively [14]. Displaying feedback on the timeline provides a consistent location to look for the information, and it fits into a person's field of view. On the other hand, feedback in the shared area provides more contextual information of the overall situation to the individual. This feedback location also better facilitates feedthrough—the observation of shared artifacts in the workspace to gain awareness of collaborators' actions and work progress [30]—by making collaborators' actions more visible to the whole team. However, the size of the display may still necessitate searching for the feedback in the shared workspace, making situation awareness maintenance more difficult for individuals. Moreover, other users' feedback on the shared area may make searching more difficult and distract users.

Existing work that explored the impact of specific input methods and interaction techniques on workspace awareness [16, 27] provides insights that helped us hypothesize how the different feedback locations may impact workspace awareness. However, our timelines were designed for situation awareness maintenance, which is a different goal from the previous work. Thus, the timeline's impacts on situation awareness and the trade-off between providing awareness and reducing distraction need further investigation.

10.3.3 Research Questions

We sought to understand the utility of different design factors for adapting interactive event timelines to collaborative tabletop applications. Specifically, we are interested in the following research questions:

- For Phase 1, how do *control placement* and *feedback location* affect situation awareness?
- For Phase 2, what are the usage patterns of the configurable timeline in a collaborative tabletop application?
- For Phase 2, how are the different system features, including the timelines, used in a group setting to maintain the three levels of situation awareness?

10.4 Studying Interactive Event Timelines in the Pandemic Game

Our literature review revealed a gap in how interactive event timelines can be adapted to a collaborative tabletop context for situation awareness support. Thus, we designed an interactive event timeline and chose a cooperative board game as our study context. We describe the study context and timeline design in this section, and we present the study design in the Sect. 10.5.

The cooperative board game, *Pandemic*, was selected for several reasons. Games allow for a more rapid, human-centred prototyping process, since it is easy to recruit "subject matter experts" of popular games. We can have more control in manipulating parameters, such as degree of difficulty. Moreover, the digital tabletop version of the Pandemic game by Wallace et al. [47] was shown to elicit situation awareness deficiencies due to automation.

Pandemic is a commercial board game for three–four players, requiring intense collaborative activities, such as forecasting game states, advance planning of actions, and managing resources. Players work together as a team, with distinct roles and abilities, to save the world from epidemic outbreaks. Players win by curing all the diseases, and lose if they run out of time (not having enough cards to draw from) or if the game state is out of control (too many outbreaks or diseases). During each turn, a player completes four actions through careful planning and strategizing, such as treat diseases, move, and exchange player cards. Then, they draw player cards, which are collected to trade for the cure (i.e., to win the game). At the end of a player's turn, the team acts as the game board (opponent) and draws infection cards that determine which cities are infected with new diseases (in the original board game, players place wooden cubes (diseases) onto the game map based on the cards drawn). *Outbreaks* and *epidemics* are critical events that increase the difficulty of the game, and players have to stay aware of them to effectively strategize.

Our digital tabletop adaptation version of Pandemic (Fig. 10.2) provided automation to help reduce manual workload and to enforce rules. For example, the system automated game board (the opponent) actions by placing disease cubes based on cards drawn, or outbreak and epidemic events. The game provided feedback of the automated changes through the following three system features:

Board. The changes were reflected on the game board, including displaying disease cubes on the map and counters around the map (e.g., cards left, epidemic counters, and cubes left). Moreover, after automated system events, three seconds of system animations appeared on the map to highlight the changes on the relevant cities (e.g., Fig. 10.3l). Different animated graphics were used to represent infection, outbreak, and epidemic events.

Infection Discard Pile. The system provided a limited history of previous infected cities in a textual log format, contained in the infection *discard* pile (see Fig. 10.3m). The pile was periodically emptied into the infection *draw* pile when an epidemic event occurred so it only contained limited history since the last epidemic.

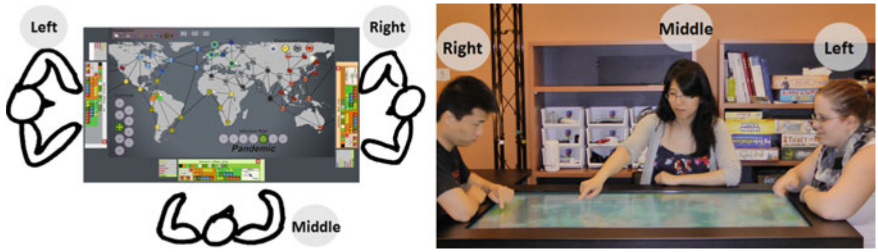


Fig. 10.2 The *Pandemic* digital game was used as the study context. (*Left*) A screenshot of the game interface, labeled with participant seating locations, based on the orientation of the game map. (*Right*) A group was playing the game

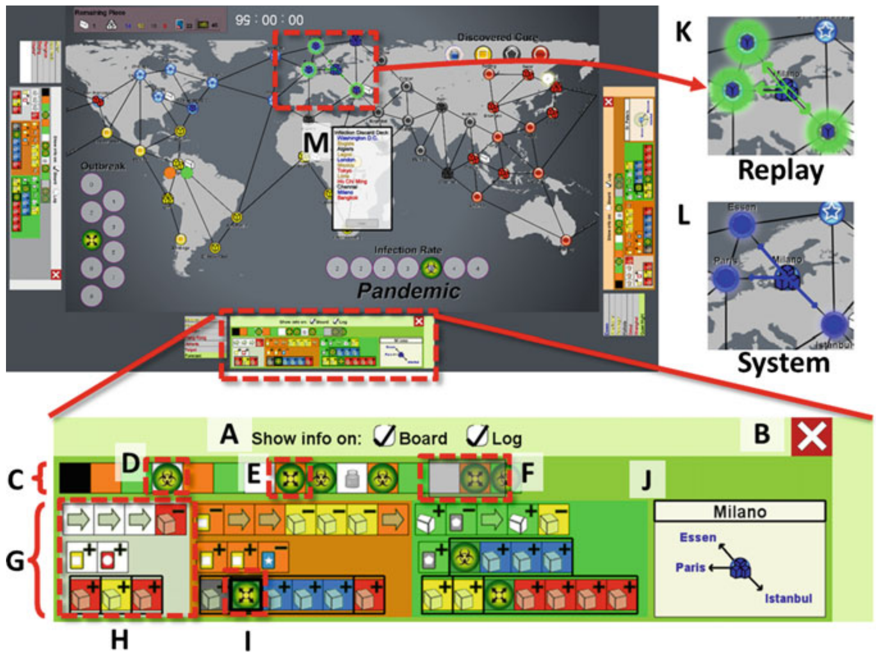


Fig. 10.3 Design of the interactive event timeline (configurable version). Users could (*A*) toggle the feedback location on the board and on the timeline as well as (*B*) close and open it at any time. *C* The overview bar showed all game turns so far with symbols denoting important game events, such as (*D*) epidemics and (*E*) outbreaks. *F* A viewport was used for selecting a timeframe to show in (*G*) the detail view. *H* A player's turn contained three rows, corresponding to the three game phases. Each block represented an action carried out by either the player or the system, and black bounding boxes grouped related game events. *I* Selected event had a thick black bounding box. Location details of the selected event was (*J*) shown on the timeline as a map cut-out and (*K*) highlighted through a replay animation on the map (in contrast to the (*L*) system animation). *M* The infection discard pile is shown at the centre of the map by default. For full details, see Chang et al. [3]

Players could open the discard pile via a button on the top left of the interface. It initially opened at the centre of the game map, and can be moved by dragging the pile.

Interactive Event Timeline. The interactive event timelines provided a complete record of events that happened throughout the game (see Fig. 10.3), and it consisted of automated game events and other players' actions.

The design was based on a task analysis of experienced to expert players playing the Pandemic game, and it was designed to fit into a player's personal territory on the tabletop, based on prior research on tabletop territoriality [38]. Moreover, it persisted on the game board, allowing players to explore prior game events at any time. The timeline showed history for one game session. When new automated events happened, they were appended to all timelines in the game. Once users started a new turn by executing new actions, the timelines automatically scrolled to show the current game turn.

The timeline consisted of two main components: an *overview* (Fig. 10.3c) and a *detail* view (Fig. 10.3g). The *overview* showed an overview of all game turns so far in chronological order, colour-coded by the in-game player colour (orange, green, and white). Critical events were marked with symbols on turns in which they occurred. To navigate through the game history, players could drag the viewport (Fig. 10.3f) or tap on a given turn to reveal details of the turns of interest in real-time.

The *detail* view (Fig. 10.3g) contained the currently selected player turns. Each turn consisted of three rows corresponding to the three phases in the game (Fig. 10.3h): (1) player actions, (2) cards drawn for players by the system, and (3) cities infected by the system. Each block represented one game event with a symbol denoting the type of event and colour derived from the colour coding scheme in the Pandemic board game. Selecting a game event could invoke two types of feedback. First, detailed information was shown next to the timeline (see Fig. 10.3j), if feedback on the timeline was enabled by toggling Fig. 10.3a. Second, a replay animation that persisted for three seconds was triggered on the shared game map (see Fig. 10.3j), if feedback on the board was enabled by toggling Fig. 10.3a. The colour of the replay animation corresponded to the player's in-game colour and the timeline colour. The colour coding was based on the original Pandemic game design.

10.5 Study

We conducted a laboratory-based study to understand how the two design factors, *feedback location* and *control placement*, impacted users' situation awareness and timeline usage. Participants played the Pandemic game with different design alternatives of the interactive event timeline, and answered questionnaires for us to evaluate their situation awareness and experience.

10.5.1 Participants

Experienced Pandemic players were recruited from the local community, and they signed up in groups of three. Thirty-six paid participants (twenty-three males, thirteen females, ages twenty-two to thirty-six) were recruited, with all team members having experiences playing Pandemic prior to the study. For this book chapter, the participants are denoted as $P_{\text{group number, seating position}}$. For example, $P_{1,\text{ right}}$ denotes the right player in Group 1 (based on the orientation of the game map).

10.5.2 Equipment and Setting

Each group of participants was seated in the lab around a 148 × 95 cm digital table (3840 × 2160 pixel, 121 × 67 cm for screen size) with an embedded PQ Labs frame to detect touch input. Two participants sat at the short edge, and one participant at the long edge, to avoid the situation of one participant seeing the game board upside down (see Fig. 10.2). The computer was running 64-bit Windows 7 using an Intel® Xeon® CPU E5-1603 @ 2.80 GHz with 4GB of RAM. Two digital camcorders were placed at different angles to capture the game sessions.

10.5.3 Study Design

There were two study phases. Phase 1 (Pandemic Challenges) was conducted using a mixed design, testing two factors:

- Control placement (between-subjects): 2 levels (shared, individual)
- Feedback location (within-subjects): 3 levels (next-to-timeline, on-board, both)

For *control placement*, half of the groups used the shared controls and the other half used the replicated individual controls. The order of the three feedback locations was counterbalanced. The widgets for toggling feedback locations and opening and minimizing the timeline (Fig. 10.3a, b) were removed in this Phase. The shared timeline was movable. This phase sought to provide empirical data to understand these two factors' impacts on participants' situation awareness.

In Phase 2 (Full Game), participants played a full game with a configurable version of the timeline, in which a group could open up to three timelines that could be moved anywhere on the tabletop. They could minimize and reopen these timelines at any time (Fig. 10.3b). Each timeline allowed players to indicate where the feedback generated by that specific timeline was displayed (next-to-timeline, on-board, or both) via toggle widgets at the top of the timeline (Fig. 10.3a). All groups had the same game setup. Phase 2 provided a more realistic usage data of the timeline to inform further improvements and to understand how it was used to facilitate the situation awareness maintenance.

10.5.4 Procedure

The study sessions lasted approximately two and a half to three hours. Participants completed consent forms and background questionnaires, and then completed the two study phases: (1) Pandemic challenges and (2) a play through of a full game.

Phase 1—Pandemic Challenges. After the researcher explained the game interface, participants played with the Pandemic game without using timelines for 10 min and completed the gameplay questionnaire (study questionnaires are discussed in Sect. 10.5.5). Then, with the same procedure, participants practiced on the same timeline variant they would see in the first Pandemic challenge. For each trial, participants started in the middle of an ongoing Pandemic game and played for two rounds (two turns for each player). We constructed three initial game states (scenarios) from real gameplay with some controlled parameters, such as the number of critical events that happened and the number of cures discovered. The order of the initial game states was randomly selected. Players individually completed post-condition questionnaires, which consisted of both the gameplay and the situation awareness (SA) questionnaires. The order of the three SA questionnaires was randomly selected. Participants were asked to rank their preferences of the timeline alternatives at the end of this phase and to provide free-form feedback.

Phase 2—Full Game. After the researcher explained the configurable timeline, participants played a full game. Then, they completed the gameplay questionnaire with a free form area for any additional comments. Finally, the researcher debriefed the participants with the goal and details of the study, and conducted an unstructured interview to receive any additional feedback.

10.5.5 Data Collection

During gameplay, we collected various data, including video recordings from two different angles, screen recordings, computer logs, audio recordings, and questionnaires. The computer logs captured all touch interactions on the timelines, e.g., tap, rotate, open, and close timelines as well as toggle feedback locations.

Two questionnaires were used to evaluate participants' gameplay experience and situation awareness. The gameplay questionnaire consisted of the Player Experience of Need Satisfaction (PENS) survey [33] and the NASA Task Load Index (NASA-TLX) survey [17]. We developed the situation awareness questionnaire by following the steps outlined in the SAGAT methodology [9, 12]. Questions were in the form of "name one city/colour/player that…", or "estimate the number of turns away from…". More details on the questionnaire can be found in Chang et al. [3].

10.5.6 Quantitative Analysis

Each individual's situation awareness questionnaire results (from Phase 1) were scored as correct (1), partially-correct (0.5), and incorrect (0) for each question. We analyzed the situation awareness questionnaires using a 2 (control placement) $\times$ 3 (feedback location) repeated measures analysis of variance (RM-ANOVA). See full details of the quantitative analysis in Chang et al. [3].

10.5.7 Qualitative Analysis

For Phase 2 (full game), two researchers analyzed eight full game sessions with an open video coding process. One researcher watched the videos and took notes of participants' discussions and activities related to timeline usage and situation awareness maintenance. An initial set of codes was then established, and two researchers coded for players' interactions with the features in the system and their discussion with teammates, e.g., interacting with timeline and discard pile, pointing at the game board, using deictic references for game cities, as well as announcing, narrating and discussing of automated events. The codes were revised until an acceptable inter-rater reliability was reached (79.39 %), and then the rest of the videos were coded.

Next, we focused on codes most relevant to participants' situation awareness maintenance, including (1) looked at or touched the timelines, (2) opened and closed the timelines, (3) toggled feedback locations, (4) opened the infection discard pile, (5) discussed automated game events, and (6) corrected each other's knowledge of the automated events. For all instances, we classified the purpose behind the observed actions and discussions as follows:

- Investigation of automation results: the coded instance was conducted for the purpose of finding out the type and location of an automated event that took place as well as connections between automated events.
- Prioritization: the coded instance was conducted for the purpose of gathering information to predict future game states and prioritize player actions.
- Other: the coded instance was for any other purposes, such as played with the timelines, rotated the timelines, and toggled feedback locations at the start of the game for learning. This category also included instances that could not be classified by the researchers (i.e., insufficient information).

The instances were also classified based on whether the participants achieved their goals, such as correct (successfully obtained correct information), incorrect (successfully obtained, but information incorrect), incomplete (failed to obtain information), and unknown (researchers were unable to classify its successfulness).

Next, to understand how players made use of various system features for situation awareness maintenance, we sequenced the codes based on game events investigated by participants. We also examined whether players' investigation of particular game events led to game commands to address them, for example, a player asked about the new infections, another player checked the timeline and found that Moscow had an infection, and the infection was treated in the same turn. We classified each sequence based on its purpose and whether it led to a success in achieving the goal. Through the video analysis, it became apparent to us that the codes classified under automation results were most closely related to the perception and comprehension levels of situation awareness. The prioritization actions were most relevant to the projection of future game states as participants gather information to determine their urgency.

10.6 Results

In this section, we first summarize the quantitative results from Phase 1 on the *control placement* and *feedback location* factors (details can be found in our previous publication [3]) to motivate our video analysis. Next, we present our video analysis results on timeline configurations and describe how timelines and other system features were used for the three levels of situation awareness.

10.6.1 Summary of Phase 1 Findings

The analysis of Phase 1 data revealed that participants who used individual timeline controls had higher situation awareness scores than participants who shared a timeline within the group ($M_{individual} = 0.71$, $M_{shared} = 0.67$, $F_{1,28} = 4.7$, $p = 0.04$). Participants using individual timelines also interacted with their timelines more on average in each condition ($M_{individual} = 12.5$, $M_{shared} = 5.6$, $F_{1,10} = 6.2$, $p = 0.03$), including navigating and tapping on game events. This finding showed the benefits of the timelines for improved situation awareness, and a partial correlation (control for group) confirmed that there was a positive correlation between interaction with timelines and situation awareness scores ($r_{105} = 0.20$, $p = 0.04$).

Although no main effect was found across feedback locations, feedback both on the game board and on timeline was most preferred as it provided the best of both worlds. For feedback on timeline, it allowed for quick navigation of game events while participants could fixate on the same area. However, participants reported that timelines felt disconnected from the game. Feedback on the board provided contextual information on the map, but participants also reported distraction and confusion associated with board feedback; this problem is further discussed in Sect. 10.6.4.

Next, we investigated the groups' interactions with the timelines. We calculated a group situation awareness score by taking the best situation awareness score achieved by any one member for each question and then taking the average of these best scores. Overall, groups had high aggregated group situation awareness scores ($M = 0.87$, $SD = 0.06$). However, there were no main effects for *control placement* ($F_{1,10} < 0.1$, $p = 0.94$) nor for *feedback location* $F_{2,20} = 1.2$, $p = 0.33$). In the shared condition, the participant who interacted with the timeline most frequently was defined as the primary user, and the rest were secondary users. However, there was no difference in the individual situation awareness score between the primary and secondary users.

We hypothesized that the information participants gathered from the timelines was shared with the group; thus, the secondary users benefited from the primary users' interactions. Moreover, participants might have gathered situation awareness information through other components in the tabletop interface. Thus, we decided to follow-up with a video analysis of Phase 2 to better understand how various system features was used, including the timelines, to gain situation awareness.

10.6.2 Timeline Configurations

To understand the usage patterns of the interactive event timelines, we examined the percent of time each feedback mode was kept for individuals and groups in Phase 2. Our data analysis revealed that participants made use of the configurable timelines, and kept it open for most of the time. Next, we investigated how much time each feedback mode was kept. When participants first started the gameplay, the timelines were set to show no feedback and were closed. However, if the game crashed and restarted (happened to two groups), the timelines were opened with both feedback locations enabled by default. The time during game crash was excluded from the analysis. For one group, the participants' timeline configurations before and after the game crash were different, and they did not all reconfigure their timelines. Thus, for this group, the time after the game crash was excluded.

As depicted in Fig. 10.4 Left, *Both* feedback was the most popular mode ($M = 60.82$ %, $SD = 42.02$ %), followed by *Timeline Only* ($M = 30.05$ %, $SD = 40.32$ %) and *Closed* ($M = 6.38$ %, $SD = 17.93$ %). *Board Only* feedback ($M = 1.37$ %, $SD = 7.31$ %) and *None* (while the timeline was open) ($M = 1.37$ %, $SD = 1.59$ %) were the least kept mode. This distribution was consistent with participants' feedback and our observations, since the *Both* configuration was also rated as most preferred in Phase 1. Some participants reported noticing interference between their own feedback and others' feedback on the board, which was likely why the second-most frequent configuration was *Timeline Only*. While one player used *Board Only* more extensively ($P_{4,\mathrm{right}}$: 14.35 $_{\mathrm{min}}$), the rest of the participants almost never kept their timelines in this mode ($M = 2.6$ s, $SD = 3.69$ s). This was likely due to the need to search for the replay animation on the map as well as the interference problem. Although the percent of time in the *None* configuration might be a result of intermediate time between toggling

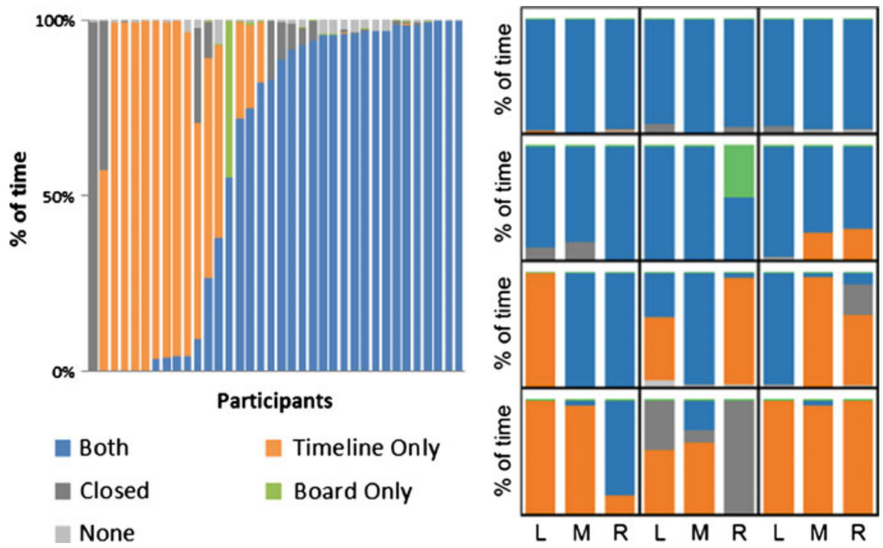


Fig. 10.4 Percent of time each timeline configuration was kept by each participant. (*Left*) Sorted by percent of time a participant kept the *Both* configuration. (*Right*) Sorted by the average percent of time a group kept the *Both* configuration. Each cell shows a group (12 groups in total) and each *bar* shows one participant, arranged by their seats (*L* Left, *M* Middle; *R* right)

feedback locations, the video analysis presented in Sect. 10.6.4 showed some benefits of the timeline as a static visualization.

Participants occasionally switched to different timeline alternatives throughout the game, but it was difficult to determine their intention based on the observable actions as there was no verbal explanation in most cases (only 6/31 cases could be clearly identified as related to understanding automated events).

We further examined participants' usage of timeline configurations as groups, and found that most groups had at least one player keeping the *Both* feedback mode on for most of the gameplay (see Fig. 10.4 Right). The last three groups (Group 2, 5, and 8 on the last row of Fig. 10.4 Right) all explicitly discussed the potential interference of displaying feedback on the map, while participants in Group 2 specifically agreed that only one player would be displaying feedback on the map. The configurable timelines allowed the groups to decide on their own strategies of maintaining awareness of each other's interactions with the timelines.

10.6.3 System Feature Usage for Maintaining Situation Awareness

As the timeline was designed to improve users' situation awareness of dynamic changes, we examined the usage of the timelines in supporting the three levels of situation awareness (i.e., perception, comprehension, and projection) as defined by Endsley [11]. The first level of situation awareness, perception, refers to the knowledge of the changes that happened. In the context of the Pandemic game, the perception level refers to knowing what the dynamic changes were, as well as whether the new changes were casual. The comprehension level refers to participants' understanding of the overall situation and of the changes that they just learned about to know their significance. Finally, the projection level refers to making predictions about future game states.

The three levels of situation awareness are internal cognitive processes. Thus, they are not directly observable without participants' verbal communication, physical interaction with the application interface, and visible body language. For example, participants may be exploring the timeline and thinking about the automated game events' impact on the overall game state. However, without verbal communication, it was impossible to definitely determine whether the interaction facilitated participants' comprehension. For this reason, few observable actions occurred for the comprehension level. Moreover, we incorporated decision making into the third level, projection, although it was originally modeled as a separate process by Endsley [11]. Participants' strategizing and prioritization behaviour represented participants' decisions in response to their projection of future game states. Since our data only recorded participants' visible and audible behaviours, we were constrained to determining how the timelines supported situation awareness based on observable actions.

We were also interested in how other system features were used for maintaining situation awareness. The video analysis revealed that the game map and the discard pile were the most relevant features used by participants. The game map refers to the connected cities as well as all information contained within it, e.g., the disease cubes on cities, locations of player pawns, and system animations that highlight particular cities. The discard pile contained a log of limited history of cities infected by automated events, periodically emptied after epidemic events. It could be opened by tapping a button, as described in Sect. 10.4.

To answer the research question on how each situation awareness level was supported by various system features, we investigated how the timelines, the game map, and the discard pile were used by our participants. In the following sections, we present data pertinent to the system features' usages for each level of situation awareness: perception, comprehension, and projection (as depicted in Figs. 10.5, 10.6 and 10.7).

10.6.4 Perception

At the end of each game turn, the system automated the drawing of new player cards (i.e., shared resources) and placing new disease infections on the game map (i.e., changes in the system state). The new changes were reflected on the associated cities and were highlighted on the map by a brief system animation. Moreover, they

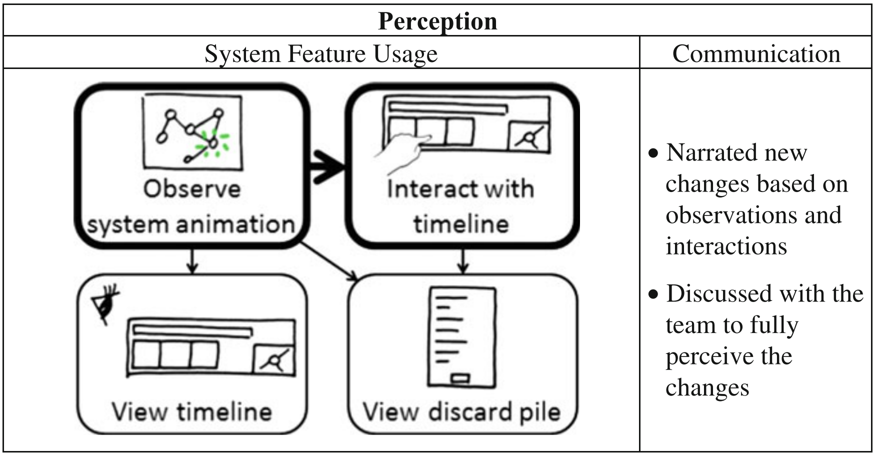


Fig. 10.5 At the perception level, participants typically first observed system animations and then interacted with their timelines to verify or further investigate changes. Changes were often narrated, and participants also discussed changes based on information gathered to negotiate their knowledge

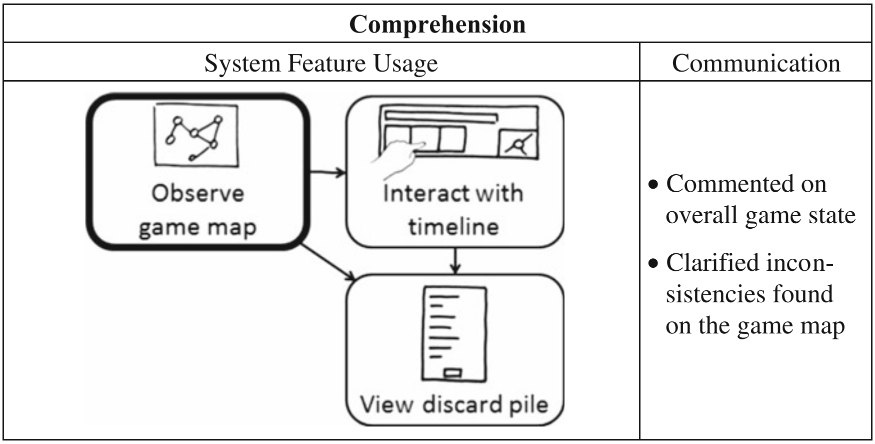


Fig. 10.6 At the comprehension level, the game map was used most frequently to discover inconsistencies between participants' understanding of the game state and the actual game state. The timeline and the discard piles were then used to gather information, which allowed them to collaboratively understand the game state

were appended to the timelines (players had to tap on the new changes to see the associated locations). Participants should aim to find out the types of events that took place, their locations and quantity, and if the events were causal.

The analysis revealed both static and interactive use of the timelines. For simple changes, participants observed both the system animations and the timelines to gain

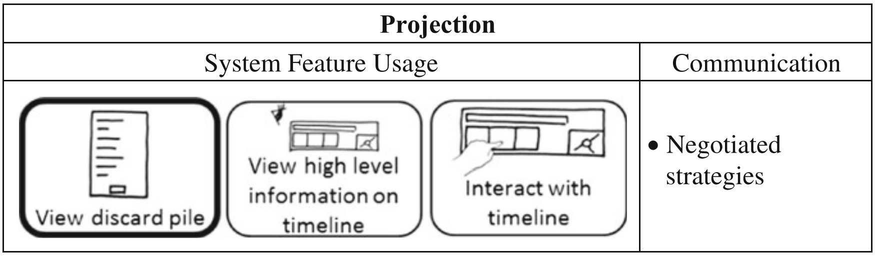


Fig. 10.7 At the projection level, the discard pile was used most frequently to prioritize actions, and participants negotiated their strategies with each other based on the information gathered. The timeline was beneficial for participants to view high level information, but it was used less frequently

awareness of dynamic changes. However, participants sometimes only caught parts of complex changes or completely missed the changes, and the timelines were then used to investigate the changes. The timelines were considered as the correct historical account, and used to negotiate participants' knowledge especially for complex changes. This section describes strategies for perceiving simple and complex changes, as depicted in Fig. 10.5.

10.6.4.1 Observation First then Investigation

Participants often narrated new changes as the system animations appeared on the game map. Due to the large size of the tabletop display and the fact that players were not constantly attending to the interface during gameplay, players sometimes missed seeing the system animations in time or only noticed that some changes took place without knowing the details (e.g., they noticed an animation occurring in their peripheral view). Complex changes that involved chained events could also be difficult to follow. Thus, participants typically first observe the available feedback then interact with timelines. For example, after the system finished animating on the map, $P_{1,middle}$ noticed it but did not know the exact associated cities from observing the map. He said to the group: "hmm… Something went pop!" He then used his timeline to locate the changes by tapping on the newly automated event and narrated the result to the group: "San Francisco." As he had *Both* feedback on, $P_{1,left}$ then pointed at the replay animation triggered and said: "right here."

The timeline was useful as a static visualization for perceiving new changes, especially because it automatically scrolled to show the current turn and placed the changes in a readily accessible location for users. There were only 9/333 cases of such usage that we coded. The actual usage could be much higher due to the constraints in precisely determining the eye gazes of participants (see Table 10.1).

Although the static timeline design did not provide detailed location information of the game events, the colour-coding of game events provided a general sense of regions. The icons indicated the types of events (i.e., infection, outbreak, or

Table 10.1 Summary of timeline usage counts. Interacting with current turn game events was the most common observed interactions

Usage categories	Counts
Tap on events in the current turn	272
Navigate to and interact with historical events	22
Learn to use timeline	14
Interact with the timeline for fun	10
View timeline (static usage without interaction)	9
Count critical events on the overview bar of timelines	6
Total	333

epidemic), and the amount of game events provided a hint to the complexity of changes. Moreover, there were a few cases where participants opened and closed their timelines only to view the changes without interacting with any specific game events (evident from their narrations). Participants sometimes narrated the colour of the player cards (shared resources) received by collaborators, showing the value of providing awareness of the changes in shared resources automated by the system. As users tended to first observe without any interactions, making changes apparent is important. While this strategy was effective for simple changes, more complex changes often required interactions with various system features.

10.6.4.2 Interactions to Resolve Complex Events

For complex disease spread, participants' process of learning the changes was often a joint, iterative effort among the team members. While the game map provided a reference to the current system state and allowed participants to notice changes, the timelines were the main tools for participants to understand how the system automation arrived at the new state, see the observe to interact states in Fig. 10.5.

Participants interacted with the timelines to verify what they observed on the game map or what they overheard from collaborators. The timeline was also used to investigate new changes. This was the most common type of timeline usage (272/333 cases). It was also common for multiple players to investigate their timelines and announce the results at the same time. We hypothesize that players did so to make sure they, as a group, had the correct knowledge of the automation that took place.

When a group was confused with complex changes reflected on the game map, they used the timelines to investigate and verbalized their perception for negotiation to reach a common ground of the events that happened. In this process, the timeline was considered as the "correct" history and was used to correct each other's "theory" of the changes. For example, while the system animation was still playing, $P_{3,\ \text{right}}$ noticed that two outbreak events just took place by viewing his timeline, and he announced this to the group. As there were two outbreak events, participants tried to determine if one caused the other. It was a complex event as three types of events happened during the same turn: an epidemic event at Chennai, two independent outbreak events at Bangkok and New York, and an infection event at Moscow. As $P_{3,\ right}$ was investigating on his timeline, the rest of the team looked at the game map on which they could see the new disease-spread system animation that was still playing, in addition to the replay animations triggered by $P_{3,\ right}$. After $P_{3,\ right}$ identified that the first outbreak event occurred at Bangkok, by checking his timeline, $P_{3,\ left}$ mistakenly thought that it caused a chained outbreak event in Chennai. $P_{3,\ middle}$ then jumped in and tapped on the second game event with an outbreak icon, and this triggered a replay animation on the game map at New York. $P_{3,\ right}$ then continued to check game events on the timeline but provided an incorrect reasoning to why the two outbreaks were not chained. As $P_{3,\ right}$ had an incorrect reasoning, $P_{3,\ left}$ finally started interacting with the game events on his timeline and announced the correct set of events that took place: “It’s with Bangkok and then New York. Those are the two outbreaks.” This observation showed the importance of the timelines for the correct perception of changes.

Our results showed that some participants appreciated the replay animation on the game map and commented that it was beneficial for keeping track of others’ exploration on their timelines. However, some also noted that it distracted and confused them. Although the system animation and replay animation looked different (see Fig. 10.3k, l for an example), participants had difficulties distinguishing these two types of animations quickly. For example, $P_{5,\ left}$ mistook the animation triggered by $P_{5,\ middle}$ as new outbreaks by the system, and announced “Bogota just outbroke!” He then quickly realized that it was a replay animation triggered by $P_{5,\ middle}$, and said “oh no, you are just smashing things. I hate you! I hate the board thing! Turn your board off, please!” $P_{5,\ middle}$ then turned off the feedback on the map. This confusion resulted in a negative response to the replay animation feature. Participants continued their discussion and pointed out that the key issue was the lack of awareness of collaborators’ actions.

$P_{5,\mathtt{left}}$: `Inform us when you are going to turn it on; otherwise, I go, ‘oh no Bogota just outbroke!’`
$P_{5,\mathtt{right}}$: `It’s kinda funny, but I also found it distracting when people do it.`
$P_{5,\mathtt{left}}$: `It’s okay as long as you tell people you are doing it.`

Due to the potential interference, some players manually toggled the feedback locations. However, this resulted in mode errors [40] where participants forgot about the current timeline mode and were confused when the replay animation was not triggered on the game map. Such observation showed a need to provide further support for workspace awareness of collaborators’ timeline interactions.

The discard pile system feature was used for perceiving new changes as well, although infrequently (8 cases vs. 281 cases for timelines). In 3 of these 8 cases, the discard pile was used in conjunction with the timelines to verify the changes. For example, after new changes took place during one group’s gameplay, $P_{9,\ middle}$ was confused about why there was an additional disease cube on Moscow. He first navigated through the game history on his timeline to find out when it first happened. $P_{9,\ left}$ then opened the discard pile to check. Then, $P_{9,\ middle}$ and $P_{9,\ left}$ found that the Moscow card was drawn and thus had a new disease in the most recent turn through the timeline and the discard pile, respectively. The discard pile acted as an alternative information source.

Overall, players reached the correct perception of the automated events most of the time (293/397 cases, 68.26 %) even though participants had to correct themselves or each other in 22/293 cases, 7.51 %. There were 99/397, 24.94 %, cases in which we were unable to determine whether their perception was correct and 5/397 cases, 1.26 %, where participants gained incorrect information or could not find the information needed.

The analysis also revealed the flexible work patterns employed by our participants. Participants sometimes ignored the system animations and continued to discuss strategy. Moreover, since advance planning of actions was common and necessary in the game, the current player sometimes focused on executing the actions agreed upon by the entire group beforehand, and relied on team members to observe and report the new changes. This finding showed the importance of providing persistent timelines for individuals to enable such flexible work patterns.

10.6.5 Comprehension

With the new changes explored, the comprehension level refers to making sense of the new changes and the overall game state. The players should seek to determine how the new changes impacted the overall game state. As participants were all experienced Pandemic game players, they generally understood the meaning of the changes. In some cases, the new changes did not have urgent impact on the game state, while in other cases participants started strategizing about how to address changes right away. We based our analysis on observable behaviours, and our data showed that the game map was used as a reference point for the group to comprehend the overall game state.

The game map was the most frequently used feature in the comprehension level to understand the overall state as well as to spot inconsistency in their understanding of the game state, as depicted in Fig. 10.6. After new changes took place, participants commented on overall game state based on the game map. For example, in one session, $P_{5,\ middle}$ commented on the overall spread of the blue colour diseases on the game map: “Oh my goodness, there’s a lot of blue going on!”.

The game map was sometimes used in conjunction with the discard pile and the timelines for players to correct their understanding of the system state. For example, $P_{1,right}$ noticed that on the map Bogota had more disease cubes on it than expected, and she asked “have we been noticing that Bogota is a problem?” Then, $P_{1,left}$ opened the discard pile for the entire group to view and $P_{1,middle}$ looked at the discard pile and clarified that “no, it’s just out [in the last turn].”

In another example, after new changes took place, $P_{3,right}$ first checked his timeline. Later on, while inspecting the game map, he found that the narrated events were inconsistent with the number of disease cubes on the map. This prompted $P_{3,right}$ to further investigate using his timeline to correct the group's knowledge, and he announced the correction to the group. Overall, the game map provided an overview of the situation for the comprehension of changes and understanding of the system state.

By the end of the comprehension stage, participants had usually reached agreement about the changes that took place and their meanings to the game. Next, they negotiated with each other on the strategies and on which actions to prioritize.

10.6.6 Projection

The projection level refers to predicting the future game states, and participants strategized, prioritized actions, and managed resources, based on their predictions. Generally, in Pandemic, players need to strategize based on when critical events would happen at which locations. This information can be estimated based on the current and past disease spread as well as when previous critical events took place. While the timeline and the discard pile were both the key system features used to help remember historical events and forecast future game states, the discard pile was the primary feature used, as depicted in Fig. 10.7.

We found that the timeline provided high-level information that was beneficial for forecasting game states. For example, players counted the number of turns since the last epidemic event on the overview bar at the top of the timelines (6/333 cases of timeline interactions). As a fix number of epidemic events was roughly evenly distributed in the game, counting the number of turns since the last epidemic was a good predictor of the next epidemic event. Epidemic events signaled that previous infected cities may be infected again soon to create wider disease spread, so it was important for players to accurately predict when the next epidemic event might occur and adapt their strategies accordingly. Players also navigated through historical events to determine if any cities might be potential problems in the future. Since all other system features only provided limited amounts of historical information (e.g., the discard pile only listed the infected cities since the last epidemic event), players had to rely on the timeline for much older events.

The analysis showed that very few timeline interactions were conducted for the purpose of strategizing (used only 6 times in 88 cases of prioritization). The amount of effort required to navigate through many game turns to locate the target game event likely contributed to the limited use of the timeline for this purpose. More often, participants opened and read the content of the discard pile to prioritize their actions (used 82 times). The discard pile provided a quick way to access recent cities that were affected by disease spreads by providing all information in a single textual log with minimal interaction required (other than to open, and potentially reposition the widget). This information allowed players to decide which cities needed more attention by comparing cities in the pile and the current game state on the map. This showed that the design of such a textual log was more beneficial for the projection level of situation awareness.

The following example illustrates a usage of the discard pile for the purpose of prioritizing actions. $P_{1,middle}$ went through a list of cities that could potentially create outbreaks based on the current game state (i.e., cities that needed more urgent attention). $P_{1,middle}$ first named Moscow and $P_{1,left}$ opened the discard pile for the entire group to see (default location is at the centre of the map). After confirming that it was not in the pile and thus was potentially high in priority, $P_{1,middle}$ continued to inquire the group about the status of Mumbai and Bangkok. $P_{1,left}$ opened the discard pile again, and $P_{1,right}$ viewed the pile and confirmed that they were in the pile, meaning that players only needed to attend to them when the next epidemic event was near. $P_{1,middle}$ thus concluded that Moscow was the only problematic city: "which is to say, Moscow is the only thing [to be concerned about]." $P_{1,left}$ agreed and reiterated on the urgency of Moscow: "that [Moscow] really needs to be dealt with right now." Players then continued to discuss how to spend actions to move to Moscow, and eventually treated diseases in Moscow in the same turn.

The discard pile was sometimes used as a tool to suggest potential actions to consider. However, this sometimes failed because there was too much information to parse through (i.e., too many cards in the discard pile), or it was simply not helpful due to the game state at the time.

10.7 Discussion

While the data analysis in Phase 1 revealed no difference across timeline alternatives in group situation awareness, the follow-up analysis showed that there was frequent communication among players to narrate information, discuss changes, and negotiate strategies. This sharing of information facilitated the group situation awareness. The game map, the timelines, and the discard pile acted as information sources for situation awareness maintenance. In this section, we discuss specific timeline designs that were beneficial for participants' situation awareness and lessons learned for potential improvements.

10.7.1 Perception: Make Changes Readily Available

The timelines were mainly used for perceiving new automated changes, and several aspects of the timelines helped participants to gather this information. The timelines appended new changes and automatically scrolled to the current turn, making the most recent information readily available for exploration. The visual design of the timeline structured the game events based on their types into three rows (i.e., player action vs. system automations) to facilitate locating game events. The colour-coding and icons provided overview information. Moreover, each timeline was placed at individual's personal area to provide quick access to new changes, visually and physically. In contrast, the discard pile was used less frequently for perceiving new changes, and this may be explained by the fact that reaching out to open the discard pile required more physical effort or more coordination to ask the player on the left (the position closest to the discard pile button) to open it. This observation also helps to explain why the shared timelines were used less frequently and resulted in lower situation awareness in Phase 1 of the study. In light of the benefits of timelines, potential redesign may consider how to further streamline the perception of new changes, such as reducing the interaction required by including more detailed information for the most recent events.

10.7.2 Projection: Provide Critical Event Overview and Summary View

While the overview of critical events on the timeline helped participants determine the overall strategy, the discard pile was used much more frequently for forecasting events and prioritizing actions. The interactivity of the timeline was beneficial for reducing clutter. However, it required a high level of cognitive and physical effort for the participants to gain an overview of the historical events to project the relative urgency of problems. Moreover, the design that the discard pile appeared at the centre of the game map by default might have better facilitated information sharing and strategizing for a tightly coupled collaboration [44], such as in our context. Future designs of tabletop applications involving dynamic changes should consider providing a different summary view of recent events in the timeline or as a different system feature to support the projection of future system states.

10.7.3 Timelines for Supporting Group Work

The timeline was designed to support situation awareness for collaborative work. Our analysis revealed that the timeline was often used in conjunction with the game map. While the game map reflected the current system state and helped participants to notice new changes, the timeline was used primarily as the correct historical account to negotiate users' perception of the changes.

We designed the replay animation, invoked by tapping game events, as a way for users to gain more detailed information of new changes and as a way to virtually point on the map for information sharing. While both use cases were found in the data, there were only a few clearly observable instances. Participants mostly physically pointed at the game map to aid their conversations, and we believe that this was due to the turn-based nature of the game and the difficulties in searching for the replay animation due to the current design and large size of tabletop displays.

Moreover, the replay animation sometimes confused the participants and they mistook the replay animations as new system automated events. Since having both replay animation on the map and map cut-out on the timeline was the most popular configuration, future designs should consider more salient workspace awareness cues for the replay animation to facilitate feedthrough [30]. Considering participants' feedback in Phase 1 that the timeline felt disconnected from the game, we may consider a design where the timeline is visually associated with the replay animation to create a redundant encoding of invoker identity and allow for quicker association. Furthermore, as participants tried to manually toggle the feedback locations, they sometimes forgot about their current setting. A potential solution would be to use a user-maintained mode [40] for the replay animation, where replay animation is only displayed on the game map when users dwell on a game event on the timeline. This design eliminates the need of toggling different modes of feedback locations.

10.7.4 Support Flexible Work Patterns

Our data analysis revealed several work patterns. Although groups' collaboration styles were mostly tightly coupled, the participants often investigated their timelines concurrently to investigate changes and verify information observed from the game map or overheard from others. Moreover, they sometimes split the workload. One participant carried out strategies previously agreed upon, and the rest of the team investigated new changes. Phase 1 and Phase 2 data analysis revealed the benefits of individual timelines on improved situation awareness. The configurable timelines also allowed participants to investigate changes at their own paces and allowed groups to use different strategies for maintaining workspace awareness.

10.8 Limitations

The Pandemic game provided a platform for rapid prototyping, and it was effective in eliciting complex planning and decision making behaviours. Moreover, its turn-based mechanics simulated the long down time and short spurs of urgent discussions that are similar to other contexts, such as emergency response and military training. However, when applying the interactive event timeline to other domains, it would need to be adapted to represent real-time data, which may impact its effectiveness. Nevertheless, the Pandemic game provided a context for quick iterations and resulted in design lessons for other co-located collaborative tools.

10.9 Conclusion and Future Work

This project contributed in further understanding timeline usage for situation awareness maintenance in the context of collaborative tabletop applications. Our video analysis showed the benefits of the interactive event timeline design for users to maintain situation awareness, especially for investigating complex, automated system events. It was used as both static and interactive visualization, and was primarily used for the perception level of situation awareness. While the summary of critical events on the timelines was useful for the projection level of situation awareness, the results showed a need to provide summarized historical information optimized for strategizing.

In the future, we would like to investigate the following timeline redesign: streamlining the perception of new changes, providing a summary view to facilitate strategizing, and enhancing the replay animation to better facilitate observation of collaborators' actions. Furthermore, we would like to deploy the system in the field, e.g., a home or a game shop. Such environments have more interruptions and will help us understand the necessity of providing different types of information on the timelines. We would also like to apply our findings to other domains that require real-time awareness displays, such as emergency response, to understand how to adapt timelines for real-time data. Finally, timelines can also be applied to other co-located contexts, where a person in a monitoring role needs to keep track of the activities happening in the workspace to provide timely assistance, such as in classrooms [24] and in workshops [19].

Acknowledgements We would like to thank the Games Institute at the University of Waterloo, the Leif exchange program, NSERC, NSERC SurfNet, GRAND NCE, and SSHRC IMMERSe for funding this project. We also thank Matt Leacock and Z-Man Games for giving us permission to use the Pandemic board game.

References

1. Bortolaso C, Oskamp M, Graham TCN, Brown D (2013) Ormis: a tabletop interface for simulation-based training. ITS 145–154
2. Carvalho PVR, dos Santos IL, Vidal MCR (2006) Safety implications of cultural and cognitive issues in nuclear power plant operation. Appl Ergon 37(2):211–223
3. Chang Y-LB, Scott SD, Hancock M (2014) Supporting situation awareness in collaborative tabletop systems with automation. ITS 185–194
4. Conversy S, Gaspard-Boulinc H, Chatty S, Valès S, Dupré C, Ollagnon C (2011) Supporting air traffic control collaboration with a tabletop system. CSCW 425–434
5. Domova V, Vartiainen E, Azhar S, Ralph M (2013) An interactive surface solution to support collaborative work onboard ships. ITS 265–272
6. Döweling S, Tahiri T, Sowinski P, Schmidt B, Khalilbeigi M (2013) Support for collaborative situation analysis and planning in crisis management teams using interactive tabletops. ITS 273–282

7. Durlach PJ (2009) Change blindness and its implications for complex monitoring and control systems design and operator training. Hum Comput Interact 19(4):423–451
8. Endsley MR, Jones WM (2001) A model of inter- and intra-team situation awareness: implications for design, training and measurement. In: McNeese M, Salas E, Endsley M (eds) New trends in cooperative activities: understanding system dynamics in complex environments. Human Factors and Ergonomics Society, Santa Monica, CA, pp 46–67
9. Endsley MR (1988) Situation awareness global assessment technique (sagat). In: IEEE national aerospace and electronics conference, pp 789–795
10. Endsley MR (1993) A survey of situation awareness requirements in air-to-air combat fighters. Int J Aviat Psychol 3(2):157–168
11. Endsley MR (1995) Toward a theory of situation awareness in dynamic systems. Hum Factors 37(1):32–64
12. Endsley MR (2000) Direct measurement of situation awareness: validity and use of sagat. In: Endsley Mica R, Garland Daniel J (eds) Situation awareness analysis and measurement. Lawrence Erlbaum, Mahwah, NJ, pp 131–157
13. Gorman JC, Cooke NJ, Winner JL (2006) Measuring team situation awareness in decentralized command and control environments. Ergonomics 49(12–13):1312–1325
14. Gutwin C, Greenberg S (1998) Design for individuals, design for groups: tradeoffs between power and workspace awareness. CSCW 207–216
15. Gutwin C, Greenberg S (2002) A descriptive framework of workspace awareness for real-time groupware. Comput Support Coop Work 11:411–446
16. Ha V, Inkpen KM, Mandryk RL, Whalen T (2006) Direct intentions: the effects of input devices on collaboration around a tabletop display. TABLETOP 8–15
17. Hart SG, California MF, Staveland LE (1988) Development of nasa-tlx (task load index): results of empirical and theoretical research. In: Hancock P, Meshkati N (eds) Human mental workload. North Holland, Amsterdam, pp 139–183
18. Heath C, Luff P (1992) Collaboration and control: crisis management and multimedia technology in london underground line control rooms. Comput Support Coop Work 1(1–2):69–94
19. Herrmann T, Nolte A, Prilla M (2013) Awareness support for combining individual and collaborative process design in co-located meetings. Comput Support Coop Work 22(2–3):241–270
20. Isenberg P, Fisher D (2009) Collaborative brushing and linking for co-located visual analytics of document collections. Comput Graph Forum 28(3):1031–1038
21. John M, Smallman HS, Manes DI (2005) Recovery from interruptions to a dynamic monitoring task: the beguiling utility of instant replay. Hum Factors Ergon Soc Annu Meet 49 (3):473–477
22. Kaber DB, Endsley MR (1997) Out-of-the-loop performance problems and the use of intermediate levels of automation for improved control system functioning and safety. Process Saf Prog 16(3):126–131
23. Kruger R, Carpendale S, Scott SD, Tang A (2005) Fluid integration of rotation and translation. CHI 601–610
24. Martinez-Maldonado R, Clayphan A, Yacef K, Kay J (2015) Mtfeedback: providing notifications to enhance teacher awareness of small group work in the classroom. IEEE Trans Learn Technol 8(2):187–200
25. Moore RJ, Gathman ECH, Ducheneaut N, Nickell E (2007) Coordinating joint activity in avatar-mediated interaction. CHI 21–30
26. Morris MR, Paepcke A, Winograd T, Stamberger J (2006) Teamtag: exploring centralized versus replicated controls for co-located tabletop groupware. CHI 1273–1282
27. Nacenta MA, Pinelle D, Stuckel D, Gutwin C (2007) The effects of interaction technique on coordination in tabletop groupware. Graphics Interface 191–198 (ACM Press)
28. Parasuraman R, Sheridan TB, Wickens CD (2000) A model for types and levels of human interaction with automation. IEEE Trans Syst Man Cybern Part A: Syst Hum 30(3):286–297

29. Parush A, Kramer C, Foster-Hunt T, Momtahan K, Hunter A, Sohmer B (2011) Communication and team situation awareness in the or: implications for augmentative information display. J Biomed Inform 44(3):477–485
30. Pinelle D, Gutwin C, Greenberg S (2003) Task analysis for groupware usability evaluation: modeling shared-workspace tasks with the mechanics of collaboration. ToCHI 10(4):281–311
31. Rensink RA (2005) Change blindness. McGraw-Hill yearbook of science and technology. McGraw-Hill, New York, pp 44–46
32. Rittenbruch M, McEwan G (2009) An historical reflection of awareness in collaboration. In: Markopoulos P, De Ruyter B, Mackay W (eds) Awareness systems. Springer, London, pp 3–48
33. Ryan RM, Rigby CS, Przybylski A (2006) The motivational pull of video games: a self-determination theory approach. Motiv Emotion 30(4):344–360
34. Salas E, Dickinson TL, Converse SA, Tannenbaum SI (1992) Toward an understanding of team performance and training. In: Swezey RW, Salas E (eds) Teams: their training and performance. Ablex Publishing, Westport, CT, US, pp 3–29
35. Salas E, Prince C, Baker DP, Shrestha L (1995) Situation awareness in team performance: implications for measurement and training. Hum Factors 37(1):123–136
36. Sasangohar F, Scott SD, Cummings Ml (2014) Supervisory-level interruption recovery in time-critical control tasks. Appl Ergon 45(4):1148–1156
37. Schmidt K (2002) The problem with "awareness." CSCW, 285–298
38. Scott SD, Carpendale S (2010) Theory of tabletop territoriality. In: Müller-Tomfelde Christian (ed) Tabletops—horizontal interactive displays. Springer, London, pp 357–385
39. Scott SD, Mercier S, Cummings ML, Wang E (2006) Assisting interruption recovery in supervisory control of multiple uavs. Hum Factors Ergon Soc Annu Meet 50(5):699–703
40. Sellen A, Kurtenbach G, Buxton W (1992) The prevention of mode errors through sensory feedback. Human Comput Interact 7(2):141–164
41. Smallman HS, John M St (2003) Chex (change history explicit): new hci concepts for change awareness. Hum Factors Ergon Soc Annu Meet 528–532
42. Smith K, Hancock PA (1995) Situation awareness is adaptive, externally directed consciousness. Hum Factors 37(1):137–148
43. Spindler M, Tominski C, Schumann H, Dachselt R (2010) Tangible views for information visualization. ITS 157–166
44. Tang A, Tory M, Po B, Neumann P, Carpendale S (2006) Collaborative coupling over tabletop displays. CHI 1181–1190
45. Tuddenham P, Robinson P (2010) Coordination and awareness in remote tabletop collaboration. In: Müller-Tomfelde C (ed) Tabletops—horizontal interactive displays. Springer, London, pp 407–434
46. Voelker S, Weiss M, Wacharamanotham C, Borchers J (2011) Dynamic portals : a lightweight metaphor for fast object transfer on interactive surfaces. ITS 158–161
47. Wallace JR, Pape J, Chang Y-LB et al (2012) Exploring automation in digital tabletop board games. In: Proceedings of computer supported cooperative work companion, pp 231–234
48. Wellens AR (1993) Group situation awareness and distributed decision making: from military to civilian applications. In: Castellan J (ed) Individual and group decision making: current Issues. Lawrence Erlbaum, Hillsdale, NJ, pp 267–291
49. Yoshikawa T, Shizuki B, Tanaka J (2012) Handywidgets : local widgets pulled-out from hands. ITS 197–200

Chapter 11
Activity-Based Collaboration for Interactive Spaces

Jakob E. Bardram, Morten Esbensen and Aurélien Tabard

Abstract Activity-based computing (ABC) is a conceptual and technological framework for designing interactive systems that offers a better mapping between the activities people conduct and the digital entities they use. In ABC, rather than interacting directly with lower-level technical entities like files, folder, documents, etc., users are able to interact with 'activities' which encapsulate files and other low-level resources. In ABC an 'activity' can be shared between collaborating users and can be accessed on different devices. As such, ABC is a framework that suits the requirements of designing interactive spaces. This chapter provides an overview of ABC with a special focus on its support for collaboration ('Activity Sharing') and multiple devices ('Activity Roaming'). These ABC concepts are illustrated as implemented in two different interactive spaces technologies; ReticularSpaces [1] and the eLabBench [2, 3]. The chapter discusses the benefits of activity-based collaboration support for these interactive spaces, while also discussing limitations and challenges to be addressed in further research.

J.E. Bardram (✉)
Department of Applied Mathematics and Computer Science,
Technical University of Denmark, Lyngby, Denmark
e-mail: jakba@dtu.dk

M. Esbensen
Pervasive Interaction Laboratory, IT University of Copenhagen, Copenhagen, Denmark
e-mail: mortenq@itu.dk

A. Tabard
Université de Lyon, CNRS Université Lyon 1, LIRIS, UMR5205, 69622 Lyon, France
e-mail: aurelien.tabard@liris.cnrs.fr

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_11

11.1 Introduction

Since the pioneering research on ubiquitous computing environments done at Xerox PARC in the early 1990s [4], there has been a growing scientific and commercial interest in Interactive Spaces. In contrast to a single interactive device—such as laptops, tablet computers, interactive displays, smartphones, or tabletop displays—an interactive space is *comprised of several collaborating interactive devices* of many form factors that work together to form a unified and shared interactive experience for several users. Figure 11.1 shows a sketch of an interactive space in which several users inside and outside a room can work together by interacting across several devices with different form factors, including wall-based displays (low and high resolution), tabletop displays, and portable laptops and tablet computers.

An early example of systems support for interactive spaces is the *i-LAND* system [5]. i-LAND supported interactive surfaces integrated in the architectural space and furniture of a 'smart room', including walls, tables, and chairs. The system allowed users to transfer documents and windows between different surfaces, as well as replicating documents and windows across several surfaces. This allowed users to interact simultaneously on multiple displays: users can make remote annotations at the wall display from one of the interactive chairs or manipulate an artifact at the table. As

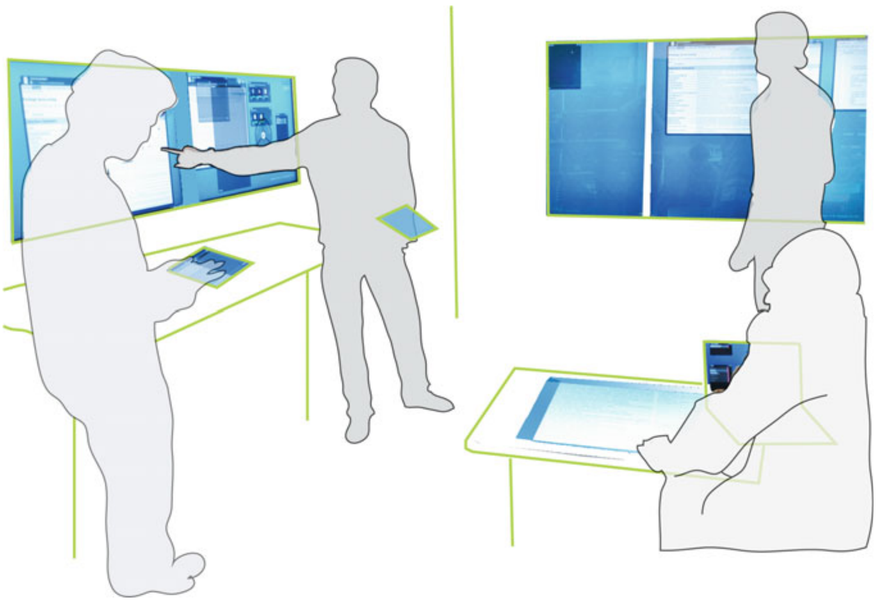


Fig. 11.1 An *Interactive Space* is comprised of a set of interactive wall, tabletop, desktop, laptop, and tablet displays that work together in unison. Nomadic users can bring devices to the space, which are then included into the interactive space setup. Collaboration across two or more interactive spaces in different locations can be initiated

such, the i-LAND system provided a *unified* interface enabling collocated and synchronous group collaboration. Similarly, *iRos* was a suite of systems components that was designed to help create applications for multiple devices with the ability to integrate portable devices in an interactive space [6]. It supported redirection of input via the *PointRight* component [7]; replication of content with the *Multibrowse* component [6]; and asynchronous exchange of documents with the *DataHeap* component. In iRos, information could be accessed across multiple displays and by mobile devices dynamically added or removed as they join or leave the interactive space.

Interactive space technologies have also been designed for specific application areas. For example, the *Impromptu* system was designed specifically to support collocated collaborative interaction in software engineering [8]. Software developers could exchange application windows by replicating them, e.g. for problem solving, or by sharing them on public displays, e.g. for discussion or reflection. To improve a collaborative interactive space experiences, Impromptu integrated special collaboratives tools, such as tele-pointers, screen sharing and instant messaging, into the interactive room technology. Interactive space technology have also been designed for clinical work in hospitals. For example, Clinical Surfaces [9] allowed clinicians to manage, access, and move patient data across a distributed multi-display environment covering an entire hospital. The system aggregated medical information relevant for a patient case and allowed clinicians to easy access this patient data across large wall-based displays situated in e.g. the patient wards, the nurses' offices, and inside operating rooms.

The promise of interactive spaces is that users will be able to fluently and flexibly utilize many different interactive devices inside a room according to their need and work activity. By being able to transfer work and use any device—both fixed and portable devices—in an interactive space, users become independent of the limitations of devices and should be able to work more efficiently together and share work. However, with the introduction of multiple devices with different form factors, used by multiple users, for multiple activities, the concept of interactive spaces introduces a new level of complexity both at the architecture level and the interaction level. It is by no means a trivial task to design the advanced interaction techniques needed for sharing, moving, and interacting with files, folders, documents, etc. across multiple devices, users, and locations as part of a collaborative work activity.

To address this challenge of complexity, our group has been researching the concept of *Activity-based Computing* (ABC) [10, 11] and how ABC can be applied in the design of distributed user interfaces for interactive spaces [1, 12–14]. The core idea in ABC is to explicitly represent the human, collaborative activity as a first class object in the computer and in the user interfaces on the interactive devices making up the interactive space. Hence, just like the desktop windowing systems of a personal operating system uses folders and files to organize computational objects, an activity-based computing system uses '*activities*' and '*resources*' for this purpose. The main idea is then, that the computational activity reflects the real-world activity, which a group of users are involved in.

In this chapter, we will outline the principles of ABC and show how this has been applied to the design of interactive space technology. In particular, we shall focus on how ABC provides support for moving digital resources across multiple devices inside and between interactive spaces—which is called '*activity roaming*' in ABC—as well as how ABC provides support for collaboration between multiple users inside and outside an interactive space—which is called '*activity sharing*' in ABC. We illustrate the use of these ABC principles in the design and implementation of two specific interactive spaces; the *ReticularSpaces* system which is a general-purpose interactive space technology similar to e.g., i-LAND and iRos, and the *eLabBench* which is a application-specific interactive space for biology experiments in a wet laboratory.

11.2 Activity-Based Computing

Early research in the 1980s pointed out that activities performed by users of computer systems show complex patterns of interleaved activities, and that contemporary human-computer interfaces provided little support for the kinds of problems users encounter when attempting to accomplish several different tasks [15, 16]. Since then, a large number of observational studies have shown that users often try to structure their work within the context of higher-level activities in desktop environments [17–19]. Users not only reason within the context of activities, but sometimes also actively tweak available tools to organize their files and folders according to these activities.

These observations have lead to a research agenda on *Activity-based Computing* (ABC) that seeks to make *activities* first-class computational objects. One of the earliest implementations of this idea was the Rooms system [20]. For a historical overview of recent ABC systems, see Bardram et al. [11]. Our group has been researching ABC for more than 10 years with a special focus on providing ABC support for ubiquitous computing [21]. The central goal is to provide a computing platform which allows the user to focus on higher-level collaborative activities rather than low-level application and data management.

The core idea of ABC is the principle of *Activity-centric Resource Aggregation*. This principle states that all documents, files, resources, services, etc. that are relevant for a human activity, should be organized into a corresponding *Computational Activity*, or just *Activity*. Each activity contains a set of participants that are associated with that activity as part of a their collaborative work, and each participant can *resume* and *suspend* the activity as part of this work. An activity can be suspended on one device and resumed on another device, and hence activities *roam* between devices in an interactive space and between different spaces. When resumed on heterogeneous devices in different work context, an activity needs to be able to *adapt* to different computational settings in which it is being resumed and used. Finally, multiple activities are *interlinked* and specific relationships may exist between them. For example in a workflow system, one activity needs to precede another activity.

ABC can be implemented in a variety of ways. Figure 11.2 illustrates a minimalist approach to an ABC framework used in the context of a biology laboratory. Here

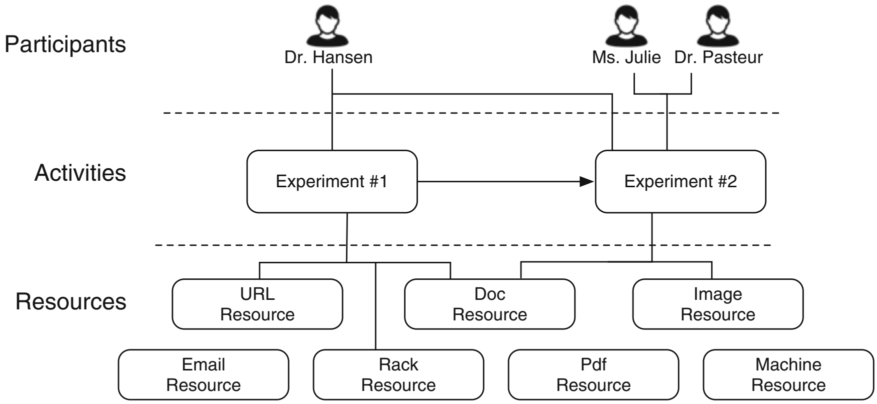


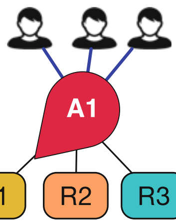
Fig. 11.2 A *Computational Activity* encapsulates *resources* and *participants* (users) in one coherent first-class object. This ABC model illustrates biology experiments as used in the eLabBench project. The model contains two activities (Experiment #1 and #2), which each aggregates a set of resources such as web pages (URLs), emails, and documents (such as PDF and Word files). Resources can be shared between activities (such as the *doc Resource*). Each activity has a set of participants: Experiment #1 has one participant (Dr. Hansen), whereas Experiment #2 has all three users as participants. Finally, the arrow between Experiment #1 and #2 illustrates a simple workflow relationship meaning that experiment #1 has to be done before #2

activities are mostly used as a means to aggregate resources and roam them across different devices. In the section below we outline the general principles of ABC and in Sect. 11.3 we present how these principles were incorporated in the two different interactive spaces: ReticularSpaces and eLabBench.

11.2.1 ABC Principles

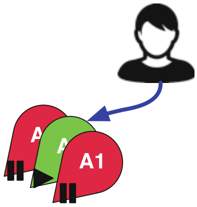
The core ABC concepts as outlined above have been crystallized into five ABC principles [10]. These principles are grounded both in theoretical models of human cognition and activity, as well as in empirical research involving the design and evaluation of ABC technologies and applications. Although different types ABC technology have evolved over time, these principles and the core concepts of ABC have remained stable.

Activity-centric Resource Aggregation



As already discussed, users organize files, folders, and other digital resources into appropriate bundles according to key activities. These resources may come from many different sources and applications (such as files, email, project management tools, editors, etc.) but all are part of an activity, which is what gives them meaning to the user. In ABC all digital resources are organized into *activities*, which are higher-level computational constructs that encapsulate all resources, tools, services, and communication mechanisms into one goal-oriented interaction model. By moving away from classic application-oriented interfaces, users are presented with logical units of work rather than the tools required to perform that work. This is especially relevant in multi-device interactive workspaces in which tools will change depending on the platform.

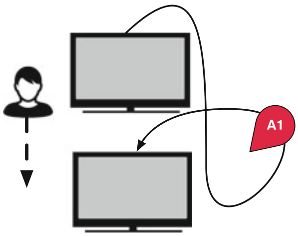
Activity Suspension and Resumption



In all modern work settings, users attend to multiple parallel work activities and often need to switch seamlessly from one activity to another. The biologist, for example, may switch between different lab experiments, teaching classes, and administrative work during a work day. Since users are involved in several collaborative activities, users often interrupt each other. ABC seeks to support the management of many parallel activities, each of which is subject to interruption, by enabling an activity to be suspended and resumed later on. When an activity is suspended, a snapshot of its current state is persistently saved. When the activity is resumed, this state information is used to enable the user to continue where s/he left off, giving immediate access to the services and resources s/he was previously using. The specific application domain and the nature of the activities and workflow determine what state information that is relevant to save when suspending an activity. In the biology case, state information about the current experiment and what resources are being used is saved, since this allow a biologist to resume an experiment later and access all resources. In practice this means that on the eLabBench, when resuming an experiment all resources (like pdf documents, spreadsheets, documents, and web pages) are shown in the exact same location and showing the same data, as when suspended earlier.

By supporting activity suspension and resumption, users can easily switch between different activity contexts. Suspending an activity means its state is stored and removed from the active workspace, while resuming an activity restores it. This feature supports parallel activities (multitasking) and interruptions in work.

Activity Roaming

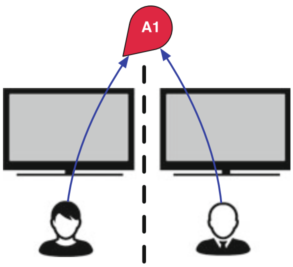


Most of earlier work on activity-centric computing have modeled a task as a collection of applications on the local computer [11]. For example, when a user refers to a particular task, the Kimura system automatically brings up all the applications and files associated with that task. This mechanism relieves the user from finding files and starting applications individually [22]. In interactive spaces, a core design challenges arise from the need to support nomadic and mobile computing in which users' context is changing as users move between different interactive devices.

For example, in biology work, a post doc researcher would conduct the same activity in a variety of contexts: working in his office to plan an experiment, checking material in the wet laboratory, in meetings to discuss the experiment, and finally tutoring students on how to conduct it. While roaming these different physical places s/he will access different devices including desktops, laptops, interactive lab benches (such as the eLabBench), and wall-based displays. It should be possible to continue the activities that s/he is engaged in within all these different physical and computational settings. These examples illustrate a core concept in activity-based computing, namely Activity Roaming. This term refers to the migration of activities from one computing environment (e.g., a desktop PC) to another (e.g., the wall-sized display in the classroom).

Moreover, by *combining* activity roaming with activity suspension/resumption, ABC enables a user to pause an activity on one device and resume it on another, with its previous state fully restored. The system will automatically bring up all the resources and services associated with the activity, thereby relieving users from manually restoring all the resources and views associated with the ongoing activity: in other words, the tools and materials for executing the operations involved in particular actions within the activity are always ready at hand [23].

Activity Sharing

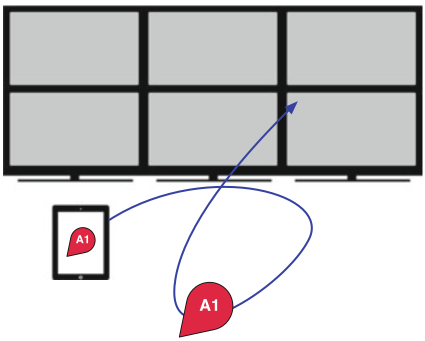


Collaboration is central to all work; yet task- or activity-centric computing approaches are mainly targeted personal information management [11]. In ABC, each activity is *shared* amongst a set of participants as illustrated in Fig. 11.2. This means that each participant can resume and access the activity. Depending on the timing of each participant's resumption, two types of activity sharing can take place; asynchronously and synchronously.

Asynchronous activity sharing happens when an activity paused by one user is resumed by another. Because the exact state of the activity was recorded when the first user suspended it, the second user will be able to re-establish the activity where his/her predecessor left off. For example, if the biology student resumes an activity that she shares with the post doc, she will see all the material and resources that the post doc might have prepared for her.

Synchronous activity sharing happens when two or more participants work on the same activity simultaneously. This can happen collocated on the same device, such as when two biologist work in front on an eLabBench; it can happen collocated on different devices, such as shown in Fig. 11.3 where multiple users are engaged in the same activity on multiple devices inside an interactive space; and it can happen when users are not collocated, but working remotely from two different devices, such as when a participant outside the interactive space in Fig. 11.3 accesses the activity unfolding inside the space. In the case where the same activity is resumed on multiple devices—both collocated and remotely—participants are collaborating within the activity and will see a synchronized view of what is going on. While a student is working on an experiment in the lab, a post-doc can add some explanatory resources from his office. In the other direction, the post-doc or professor can follow notes and pictures captured by the student from the eLabBench, as they are produced. An important aspect of synchronous activity sharing is that collaboration session management [24] is incorporated into the activity concept, since the activity functions as a collaborative session manager.

Activity Adaptation



The principle of activity adaptation supports *multi-device configuration*. When an activity is resumed in an interactive space, the different resources and service may be resumed on different devices, which then is synchronized by the overall activity. For example, in the *ActivitySpace* system, an activity resumed on a tabletop would allow users to include portable devices like smartphones and tablet computers as auxiliary displays showing some of the resources (e.g. images) in the activity [14].

An interactive space is comprised of many different types of devices with many different form factors and capabilities in terms of hardware resources, connectivity, screen size and resolution, interaction modalities (e.g. touch-based), and operating systems. In order for activities to be able to roam between devices they need to adapt to the capabilities of these devices. This can be *technical adaptation* to the resources, connectivity, and sensors available on a device, as well as *user interface adaptation* to the different interaction and displays capabilities of an interactive device inside a interactive space. Hence, an activity might have different (technical) resources available and may look quite different, depending on whether it is resumed on a wall-sized display, a tabletop, or on a smartphone.

11.3 Interactive Space Technology Cases

The ABC principles have been applied in the design of several systems since they were originally outlined in 2002. In this chapter we shall present and discuss two cases where ABC have been applied in the design of Interactive Space technology: the *ReticularSpaces* and the *eLabBench* systems. These two systems have been documented in detail elsewhere and in this chapter we shall only discuss the role of ABC in their design. The two systems are quite different in how they use ABC; ReticularSpaces is a general purpose platform and user interface for interactive spaces which seeks to implement all of the ABC principles, whereas the eLabBench is special-purpose system for wet laboratory research in biology, which focuses primarily on activity-centric resource aggregation and activity roaming. Both technologies provides support for collaboration, but in two very different ways.

11.3.1 ReticularSpaces—Multi-device Collaborative Interactive Space

In personal computing one user is typically using one device in one location. A core challenge in the design of interactive space technology is to support work which is distributed across multiple devices, involving multiple users in multiple locations. ReticularSpaces [1, 13] was designed to address this challenge. Our approach was to use the 'activity' of a set of collaborating users as the core mechanism for coordination across multiple devices, users, and locations. Figure 11.3 shows ReticularSpaces in use. By implementing the ABC principles, ReticularSpaces introduces a novel infrastructure and user interface for interactive spaces focusing specifically on activity-based support for device management, information management, mobility, and collaboration.

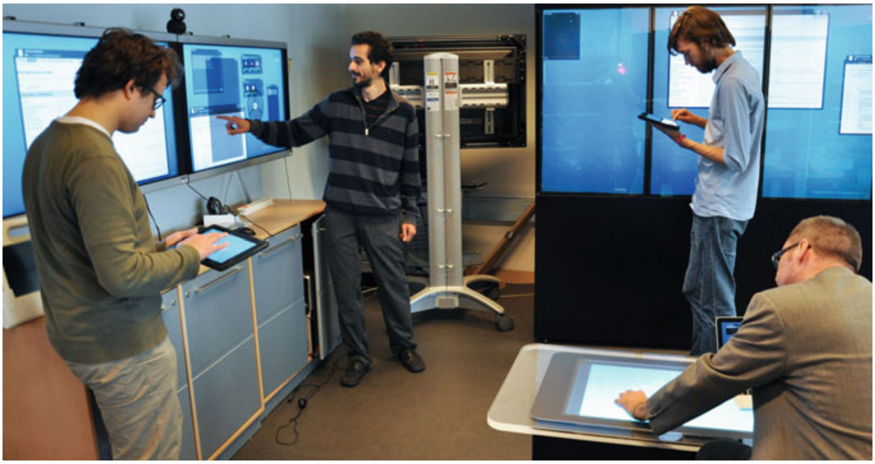


Fig. 11.3 ReticularSpaces in use by four users using six devices; two wall-based interactive displays, two mobile tablet computers, one tabletop display, and a laptop

Following the ABC principle of **activity-centric resource aggregation**, the ActivityManager organizes all documents, resources, services, etc. into a set of activities. Each *activity* is composed of a set of *actions*, each again holding a set of *operations*. Each operation points to a *resource*, such as a document, a picture, html page, etc. Resources can also be external *services*, such as a device, like a printer, which can be accessed through an operation. Each activity has a list of *participants*, and only participants can access (resume/suspend) the activity, and its actions and operations. *Relationships* allows users to organize activities, actions, and operations in different workflow structures. Such structures could be simple association links showing which activities are related, as well as more complex workflow constraints specifying which activities has to be completed before another activity can be resumed. Figure 11.4(6) shows two related activities represented as a white line with text describing the type of relationship ('Belong to the same project').

The two main user interfaces of ReticularSpaces are the *Activity View* (Fig. 11.4) and the *Action View* (Fig. 11.5). The *Activity View* provides an overview of all relevant activities from mounted activity managers, as well as contextual information about location, collocated users, and available activity managers. Each activity (the white box) can be expanded to show its list of actions and participants. Workflow relationships between activities are shown as lines with a text label. **Activity suspend and resume** in ReticularSpaces happens when a user clicks an action in the Activity View (e.g. Fig. 11.4(4)) thereby resuming this action and is taken to the Action View. The *Action View* (Fig. 11.5) shows the action's operations and the resource each operation links to, such as a source code document (Fig. 11.5(1)) or a web page (Fig. 11.5(2)). The action view can show various overview panels as shown at the bottom of the view (Fig. 11.5(3–6)). These are overviews of: all actions in the overall activity (Fig. 11.5(3)); operations in this action (Fig. 11.5(4)); available resources

Fig. 11.4 The *Activity View* showing a list of available activity managers (*1*), a list of users in this location (*2*), and the relevant set of activities from all mounted activity managers (*3*). Each activity (the *white boxes*) can be expanded to show its list of actions (*4*) and participants (*5*). Relationships between activities are shown as *lines* with a text label (*6*)

Fig. 11.5 The *Action View* is displayed when a user resumes an action by clicking on it in the activity view. The action view shows the action's operations and the resource each operation links to; in this case a source code document (*1*) and a web page showing Java documentation (*2*). The action view can show various overview panels as shown at the *bottom* of the view. From *left* to *right* these are overviews of; all actions in the overall activity (*3*); all operations in this action (*4*); available resources (*5*); and the participants (*6*). Users can communicate using the action log (*7*) and the remote video feeds (*8*)

(Fig. 11.5(5)); and the participants (Fig. 11.5(6)). Users switch between the Activity and Action Views by suspending and resuming an action. When clicking an action inside an activity in the Activity View, the action is resumed and the user interface

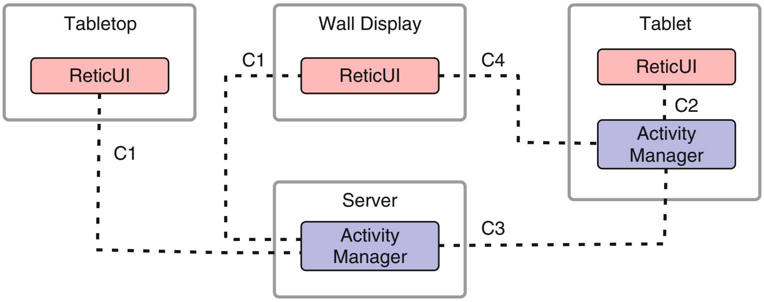


Fig. 11.6 The software architecture deployment diagram for the setup in Fig. 11.3

shifts to show the Action View. When a 'suspend' button (not shown) is clicked in the Action View, the user interface shifts back to the Activity View.

In ReticularSpaces, the ABC principle of **activity roaming** is supported via a peer-to-peer (P2P) infrastructure that enables clients to manage their own activities or to discover and mount distributed activity managers. As illustrated in Fig. 11.6, the ReticularSpaces software architecture consists of two main components; the *ReticUI*, which is the user-interface component, and the *ActivityManager*, which stores, manages and distributes all data. Devices can run either ReticUI, the ActivityManager or both. Figure 11.6 shows the deployment diagram reflecting the setup of devices and displays in the interactive space shown in Fig. 11.3. In this deployment setup, the interactive space runs a dedicated ActivityManager on a separate *Server*. Each of the fixed displays in the interactive space—i.e. the *Wall Display* and the *Tabletop*—runs a ReticUI client that connects to this central ActivityManager (the *C1* connections). When a mobile device—in this case a *Tablet* running its own ActivityManager connected to its ReticUI (the *C2* connection)—enters the interactive space, the two activity managers will discover and connect to each other (the *C3* connection). The ReticUI on the Wall Display can now mount the newly discovered ActivityManager to get access its activities and resources (the *C4* connection).

This architecture allows activities to be shared via a central ActivityManager (such as the *Server* in Fig. 11.6), thereby enabling users to access activities and their associated resources and data from distributed ReticUI clients. Moreover, the infrastructure supports a mobile ActivityManager (such as the *Tablet* in Fig. 11.6) to enter and be discovered by the interactive space. This allows displays inside the space to mount this newly discovered activity manager (as shown in Fig. 11.4(1)) and access the activities and data on the mobile device enabling a user to access and e.g. present data from this portable device on a display in the space. Vice versa, this also allows the ReticUI on the mobile device to discover and access data from the server in the space, thereby enabling mobile users to access and use local data. As such, the P2P architecture of ReticularSpaces has very flexible support for different mobility scenarios.

Data is managed as resources in the ReticularSpaces architecture, as also illustrated in Fig. 11.2. An ABC resource entity either contains the data, or points to a piece of data outside the ReticularSpaces architecture. For example, data are typically referenced using existing Internet protocols using URIs and a standardized protocol like HTTP, IMAP, or FTP, and are rendered based on their MIME type. Assuming that the ReticularSpaces runtime architecture have access to the Internet and thus online data resources, data will always be available during roaming between different devices and interactive spaces.

ReticularSpaces supports **activity sharing** and collaboration in multiple ways. Since an activity (and actions) has multiple participants, the activity and its data are shared and can be accessed by all participants. This enables a participant to resume an action, work on it in the Action View, and hence supports *asynchronous activity sharing* in which users can take turns in accessing and working with data in an activity and action. The P2P infrastructure allows participants to access data on shared and personal devices. Hence, users in the interactive space in Fig. 11.3 can access the activities and data in all activity managers inside the room, including the data on the tablet computer carried by the user entering the room. This allow for collocated collaboration and data sharing. Moreover, ReticularSpaces supports *synchronous activity sharing* where two (or more) participants resume the same action simultaneously on different devices. Synchronous collaboration is shown in the Action View in Fig. 11.5 in which two participants have resumed—and is hence engaged in—the same action. Synchronous action sharing have the effect that the user interface elements such as window positioning and size are synchronized in real time across the two device displays. This means that users working on different devices on the same action will see a synchronized view similar to the WYSIWIS principle.[1] ReticularSpaces also provides a live video feed between the two participating devices, as shown in Fig. 11.5. Finally, beside logging events (like resumption and suspension of actions), the *Action Log* allow users to type messages to the activity, which can be read by its participants. During asynchronous activity sharing, users can use the log to leave messages to each other, whereas during synchronous activity sharing, the log works as a instant messaging system.

11.3.2 eLabBench—Activity-Based Collaboration for the Biology Laboratory

The eLabBench is part of an activity-based computing infrastructure for biology research [2, 3]. It is designed to support the transition between the planning, execution, and analysis phases of biologists' experimental research, by connecting desktop computers to interactive tabletops located in the wet-lab (see Fig. 11.7 for an example of use). This work is highly collaborative. In the planning phase, graduate

[1] Acronym for "What I See Is What You See", used for groupware that guarantee that users see the same thing at all times.

Fig. 11.7 The eLabBench in use by a biologist conducting a lab experiment organized as an activity. Both physical resources, such as the test tube racks, as well as digital resources, such as research articles and a web-based lab notebook, are included in the activity

students will meet with post-docs or professors to discuss experiments, and plan their experimental protocols based on past experiments from colleagues. During the execution phase, multiple researchers can be involved in the same experiment either for teaching/learning purposes, or based on unique expertise in a tool or method. Finally, during the analysis phase other participants may be involved, either to process material or run specific analyses. In practice, experiments are often iterated upon, until conclusive results are reached.

Activity-centric computing fits well with this kind of experiment-centric laboratory work, which on top of being collaborative, is highly distributed. Distribution can be global with research teams spread across the globe, and local at the scale of a laboratory building. Typically an experiment will be planned in meeting rooms and offices, conducted in a wet lab, with back and forth to specialized lab rooms with unique equipment. Once the data is gathered from several machines and servers, it will be analyzed in the office.

The eLabBench infrastructure allows biologists to conduct activities across multiple devices and location while supporting collaboration. Figure 11.8 shows the overall architecture of the eLabBench system. The *activityDock* is a desktop application running on personal computers (see also Fig. 11.10) and the *ABC Server* is a distributed data management infrastructure responsible for collecting and distributing digital data between personal computers and the eLabBenches. This architecture

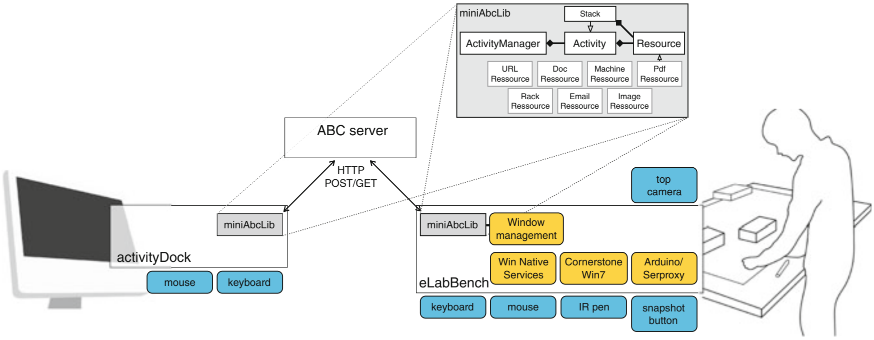


Fig. 11.8 The overall architecture of the eLabBench system, with its three main components: The activityDock running on Desktop computers, the ABC server in charge of roaming activities and the eLabBench running on tabletops. The activities are managed locally on each client with the miniAbcLib

supports activity-centered resource aggregation, activity suspend/resume, activity roaming, and activity sharing.

Biologists typically organize their work and information around experiments, and an *experiment* is often the chosen unit for an activity. An activity is a collection or **aggregation of resources** that maps the digital information a biologist uses during the experimental cycle, and serves as a placeholder for all captured data. Biologists are able to create, delete and archive activities and their associated resources. Figure 11.9 shows a closeup picture of a biologist working at the eLabBench where he has resumed an activity containing relevant resources for this biology experiment. Examples of resources include a set of lab notes handled in a digital lab notebook (in this case a wiki-based system); article and videos explaining a specific protocol; online resources such as instructions on the use of biohazard material (in this case accessed through a web browser); or RNA and DNA sequences which are accessed, stored, and analyzed in a bioinformatics tool (in this case the CLC Bio Workbench[2]). More generally, the eLabBench supports the visualization of different kinds of digital content like PDF files, text documents, spreadsheets, pictures, web pages, emails, etc. By allowing biologists to access this broad range of digital content, the eLabBench aims at covering the most common information needs of a biologist.

By aggregating the relevant resources in a versatile structure, while leaving originals in their respective tools or system, i.e. email, bio-informatics suite, etc., the eLabBench enables the creation of reusable activities. The eLabBench also enables biologists to capture data while working at the bench during an experiment and adds it directly into the unfolding activity. This includes adding files to the activity, such as pictures, documents, and spreadsheets, as well as handwritten notes, picture from the top-mounted camera, and digital annotations on physical racks of test tubes.

[2]http://www.clcbio.com/products/clc-main-workbench/.

Fig. 11.9 Activity-based Resource Aggregation—When the biologist conducts his experiment, he has access to all relevant resource and these are automatically shown on the eLabBench when resuming an activity. In this case, resources include an annotated testtube rack, the wiki-based lab notebook, and a set of other web sites. The menubar at the lower right allows the biologist to add additional resources to this activity, including websites, a calculator, and a video that records an experiment through the top-mounted camera

Reusable activities coupled to capture tools enable long-term collaboration and reuse or activities.

Activity suspension and resumption is supported by an *Activity Bar* on the eLabBench allowing the user to access his or her list of activities and to resume these. The user is identified by a simple username/password login to the eLabBench. Only one activity at the time can be resumed on the eLabBench. When resuming an activity, the state of this activity is restored thereby bringing up the digital resources in the same state and UI position as when paused, just like a virtual desktop manager. By suspending and resuming activities from the activity bar, the user is able to alternate between many concurrently running experiments. It can also be a way to hand-off experiments to colleagues.

The ABC Server supports **activity roaming** between an activityDock on a PC and the eLabBenches in the lab. This means that activities and resources can be moved between offices and laboratories, and in-between laboratories. Figure 11.10 shows the use of the the *activityDock* in the office. The activityDock lists all the activities that the current user participate in. Using the activityDock, a biologist can prepare an experiment in the office by creating an activity. Then, he can add resources to it, such as a protocol from the wiki lab book, PDF files of research papers, an email

Fig. 11.10 Activity Roaming—The *activityDock* in use by the biologist in his office while preparing for an experiment. He adds resources to the activityDock, such as the DNA structure shown on the right, which then later is accessed by resuming this activity on the eLabBench in the lab

from a sample provider, etc. The biologist can also prepare racks from their offices by describing the layout and content of each tube, and thereby prepare for the physical execution of the experiment.

When moving to the laboratory, the biologist can access the relevant activities from any eLabBench in the lab. This allows him or her to load the necessary resources, access the experimental protocol, and record relevant information during the experiment. Documentation can be done with annotations to the protocol, by adding text notes to the activity, or directly in the wiki lab book. Activity roaming also enables the biologist to move an experiment between different eLabBenches, if need be. Back in the office, the biologist can resume the activity and thereby continue working on the activity and its resources. For example, checking notes and documenting more precisely the results of the experiments in the wiki lab book.

This basic roaming mechanism supports the iterative nature of biology work where biologists go back and forth between analytical work on the PC in the office and experimental work in the wet lab. It also supports distributed collaboration: while someone works in the laboratory on the experiment, a colleague can follow changes to the experiment by monitoring changes to its activity doc, but also provide support for instance by clarifying a shared protocol.

11.4 Activity-Based Collaboration for Interactive Spaces

Support for collaboration is core to interactive spaces; one of the main design rationales is to move beyond 'personal computing' towards 'collaborative computing'. Opening up the collaborative design space, we can identify different types of collaboration that we would like to support in interactive spaces technologies. A simple taxonomy of such collaboration types is shown in Table 11.1. This taxonomy is divided across the *locality* dimension—are users collocated or remote in one or more interactive spaces—and the *working* dimension—are users working independently, engaged or closely together. By '*engaged*' we mean when a group of users are engaged in the same activity but most of them are not actively working to change any resources. An example would be one person making a presentation for a group of people in the interactive space. In contrast, by '*together*' we mean when a group of users simultaneously work on the same resources and changing them. For example, pair-programming in software engineering or when two biologists sit in front of the same eLabBench and work on the same experiment. In the following, we will detail these six types of collaboration and how activity-based computing supports these, exemplified by the ReticularSpaces and eLabBench systems.

11.4.1 Independent Collaboration

The first type of collaboration is when users work collocated in an interactive space on independent work tasks. For example, two biologists working side-by-side in the laboratory on two separate eLabBenches. In this case, support for sharing of resources, handing over tasks, and collocated workspace awareness becomes relevant. In ABC, resources can be part of multiple activities, which allows for sharing of resources. For example, the two biologists can share both physical resources like a test tube rack, as well as digital resources, like a lab protocol (a 'URL Resource'

Table 11.1 Simple taxonomy of collaboration in Interactive Spaces. The vertical *locality* dimension differentiates between users who are either collocated in one interactive space or located remotely in different interactive spaces. The horizontal *working* dimension differentiates between users working independently, focused, or closely together

	Independently	Engaged	Together
Collocated	• Sharing resources • Handing-over tasks • Workspace awareness	• Moving and displaying resources • Engaging in tasks • Personalized views	• Concurrent modification of resources • Simultaneous task management • Shared views
Remote	• Turn-taking • Remote awareness	• Remote presentation • Communication	• Synchronized views

in Fig. 11.2), between the two eLabBenches. If the two biologists are using the same protocol for different variations of an experiment, they can access, use, and update the information about the protocol and the description of a shared rack of test tubes that holds test material, from each of their eLabBenches.

Handing over tasks between users working inside an interactive space is a core feature. For example, different tasks are typically handed over and allocated during stand-up meetings between a group of software engineers. In ABC, task allocation is supported by adding (and removing) users from the list of participants of an activity. ReticularSpaces was evaluated according to a software engineering scenario and this kind of task allocation was common for the participants to do. As for creating workspace awareness—i.e., enabling users inside the interactive space to monitor and see what others are doing and what is going on—the physical layout and design of the space and its interactive surfaces and devices plays a core role. The use of large-scale interactive displays on walls, tables, and tablet computers is instrumental in providing collocated users with a shared workspace awareness. For example, the large surface of the eLabBench allowed collocated biologists to monitor what was going on in the laboratory, including seeing the physical content and layout of the test tube racks on neighbor eLabBenches. Moreover, support for virtual workspace awareness is also needed in a collocated setup. For example, in ReticularSpaces the Activity View (Fig. 11.4) would continuously update the lists of available activity managers (i.e. devices) and users inside the interactive space, and the list of users would show what activity each user was engaged in. This provided users with a rudimentary workspace awareness about available devices (with activities and resources) and users inside the room.

The second type of collaboration is when users works remotely (e.g., in two different rooms) on independent tasks. For example, two software engineers working in separate offices. In this case, support or turn-taking and remote awareness become important. In ABC, turn-taking is supported by asynchronous collaboration, i.e., the ability that one participant can resume and continue working an activity, which has previously been suspended by another participant. In the eLabBench, this allows the supervisor to prepare an experiment for a group of students. This kind of turn-taking requires some sort of workflow support that enables the suspending participant to signal to the resuming participant that s/he can now take over. This kind of support for 'signaling' was, however, not designed as part of neither the eLabBench or ReticularSpaces, which was a shortcoming also discussed during the evaluation of them. Just like in collocated collaboration, workspace awareness is essential in remote situations. Remote awareness includes being aware of the location, activity, work load, and working context of collaborators, even when not directly collaborating with them on a task (right now). In ReticularSpaces remote awareness was supported through the same mechanisms as collocated awareness by showing the location and resumed activity for devices (activity managers) and users (see Fig. 11.4), whereas the eLabBench did not have any support for remote awareness.

11.4.2 Engaged Collaboration

The third type of collaboration is when users are engaged in the same activity in the same interactive space. For example, when a user enters an interactive space, gets access to a presentation on her portable device, and presents this on an interactive wall-display. In ABC this kind of engaged collocated collaboration is supported via activity roaming, i.e., the ability to move an activity with its associated resources between devices, and activity adaptation, i.e., the adaptation of resources to the devices on which it is resumed. In ReticularSpaces, a user would be able to mount the activity manager on her portable device on the wall-display inside the room, and directly from the wall-display resume the relevant 'presentation' activity. This would then display the presentation adapted to the display size of the wall-display. Users inside the room can be added as participants to the presentation activity and hence get access to the presentation resources. Once participants are engaged in an activity, they would need support for personalized views, i.e., the ability to display, render, annotate, and change the resources of the activity. For example, users listening to a presentation might want to see a relevant video (which is a resource in the activity) on their own laptop or they might want to make personal notes to the presentation. Neither ReticularSpaces or the eLabBench supports this personalized view, which we found during our evaluations to be a limitation. For example, the annotations that can be made during an lab experiment in the eLabBench is stored as part of the activity and is hence available and editable for all participants of the activity; hence these annotation are not personal. Similarly, since the display and rendering of all resource are tightly synchronized in ReticularSpaces, if a participant would launch a video on her laptop, then this video would also be displayed on the wall-display for everyone to see. Our studies showed that better support for moving between personal and synchronized modes of collaboration needs to be investigated.

The fourth type of collaboration is when users are engaged in the same activity remotely. For example, when doing a presentation, which also involves remote participation. From an ABC point-of-view, remote collaboration is supported by the same means of activity roaming and activity adaptation, combined with synchronous activity sharing that allow for simultaneous access to resources on remotely located devices. In ReticularSpaces, remote participants could join the 'presentation' activity and the presentation would then be displayed on their local devices and the video-link and the action log chat window would be established to support communication across the two interactive spaces. Note that in this type of remotely engaged collaboration, the activity (i.e., the presentation) would still be driven from the presenter in one interactive space, having the remote participants joining as listeners. If the remote participants would start working on the activity (or more specifically on its resources), they would be working together instead (right-hand column in Table 11.1).

11.4.3 Collaborating Together

The fifth type of collaboration is when users collaborate actively together inside an interactive space. For example, when two biologist work together in front of the same eLabBench. In this type of collaboration, concurrent modification of shared resources is essential. For example, allowing the biologist to access, view, and modify the resources in the shared activity including updating the experimental protocol, note down results, adjust content to the test tubes, and to add annotations to a pdf document. To a large extent, this was supported by the eLabBench system which was implemented using a multi-touch user interface that allowed several users (at least two) to work in front of it. However, the eLabBench system was limited by its underlying operating system (OS)—in this case Windows—and the applications running inside this OS. Hence, concurrent editing on the same device in the same application was not possible. For example, both biologists could not update information in the bioinformatics application on the eLabBench, and even though two browser windows each showing the web-based lab notebook could be accessed on the same eLabBench display by the two biologists, the limitation of single-input focus in the Windows OS was a limiting factor to true concurrent editing and browsing. Hence, to support true concurrent modification of resource in ABC, the underlying OS and applications used to access the resources need to support this concurrency—something which yet only exists in to very limited degree. The ReticularSpaces system was subject to the same limitations.

Collocated collaboration inside an interactive space also requires support for simultaneous task management allowing collaborating users to access, modify, and update task information. This would include updating basic task information, but in particular to update the list of participants who can work together on the task and the workflow relationships between tasks. In ReticularSpaces this was supported by having a very open access control mechanism; basically all participants of an activity could add participants (but only the user him or herself or the owner could remove participants). Moreover, workflow modeling was accessible for all participants of an activity. This allowed users to model simple workflows across existing activities while working on them inside the interactive space.

Finally, a shared view involving one or several interactive displays is a core requirement for collocated collaboration. Many interactive workspace technologies are designed to allow users to easily move and distribute resources across multiple shared displays inside the space (e.g., the i-LAND and iROS technologies allowed for this). In ABC there is less explicit support for this. Instead the more generic principles of activity roaming, activity adaptation, and activity sharing support creating a shared view across multiple devices and displays. In ReticularSpaces, for example, the same activity can be resumed on several displays inside an interactive space simultaneously, which will result in a synchronized view of all the resource on all displays. The benefit of this approach is that the same concepts and user interface mechanisms support both collocated and remote collaboration, since the system keeps a synchronized view on all resources, also when the displays are distributed in physically separate rooms. The drawback to this approach is that it is not possible to 'split up' an activity and display its resources on different displays. Hence, you would not be able to show e.g. a code section and an UML diagram from the same activity on two different displays, since all displays are kept synchronized.

In the sixth and final type of collaboration, users work closely together from two (or more) remote interactive spaces in separate locations. In this case, being able to share a synchronized view on the work task is essential. This is the principles of synchronous activity sharing in ABC as also implemented in ReticularSpaces. The clear benefit to this solution is that collaborating users share the same view as they work closely together. For example, when two software engineers engage in pair-programming, they see the same code segment and the same viewport and zoom level of the UML diagram, which makes it easy to point to (with telepointers) and talk about the code. The drawback to the current implementation of the ReticularSpaces is the lack of more personalized views, as also discussed above. Hence, one of the engineers cannot open e.g., another UML diagram to consult without this diagram also being shown on the remote display of the other engineer.

11.5 Conclusion

This chapter offered an overview of the core concepts of Activity-Based Computing (ABC) with a special focus on the use of this approach for the design of interactive space technology. ABC builds on five core principles:

- *activity-centric aggregation* of computational resources which makes them easily accessible;
- support for *suspending and resuming* activities across multiple work context which supports multitasking and interruption;
- support for *roaming* activities and their associated resources between multiple devices, which supports mobility;
- *sharing* of activities and resources amongst multiple collaborating users; and
- enabling the activity to *adapt* to available resources and devices on which it is resumed and hence executed.

These five ABC principles were first outlined in 2002 and have proved to remain very stable over time when forming the basis for the implementation of different ABC technologies for different application areas, including the design of the interactive space technologies presented in this chapter. We presented two different types of technologies. *eLabBench* is an interactive space for wet lab biology work, including an interactive multitouch laboratory bench. In this setup, the eLabBench infrastructure implemented support for activity-centric resource aggregation, roaming, and sharing, which allow users inside and outside the biology lab to work together across multiple devices—the core requirements of an interactive space. The *ReticularSpaces* system was a much more elaborate infrastructure for interactive spaces, which implemented all of the ABC principles. This design case showed that the ABC principles are a very solid basis for the design of interactive spaces with a coherent computational and user interaction metaphor. As a general-purpose infrastructure for interactive spaces, ReticularSpaces have proved to be very flexible while providing many features in a simple and coherent manner.

The research on ABC have, however, also revealed a set of challenges still to be addressed. In particular, the core challenge to activity-centric computing is that most existing computer operating systems and applications do not have a notion of 'activity', which means that it is notoriously difficult to implement an activity-centric computing model in contemporary operating systems and applications. A simple example is mail; a mail message is a typical resource in an activity where a set of related emails should be referenced by an activity. Technically this is rather straightforward using IMAP. However, there are—to our knowledge—no IMAP email client that allow rendering of the email message alone without the entire email client with all other emails visible—emails that are completely irrelevant to the current activity. Hence, a common challenge in the design and implementation of ABC technology have been that many applications had to be re-implemented in order to make the 'activity-aware'. Data management has been another recurrent challenge to the implementation of ABC technologies. Some data types which typically reside on shared servers and can be accessed through standardized protocols is a perfect match for activity-centric data management since these can be accessed during activity roaming and suspend/resume. However, files that reside on personal computers have turned out to be a particular challenging to handle in ABC since such files (and folders) are hard to migrate or replicate across multiple devices. One approach to pursue here is adopt file replication mechanisms as implemented in Dropbox and similar file synchronization protocols. But again, these protocols are agnostic to the concept of activity, and activity-centric data management and replication thus has to be implemented in addition to the basic file synchronization.

Based on this, we can conclude that Activity-Based Computing and its principles seems to be a technological approach very well suited for the design and implementation of interactive spaces. However, ABC and interactive space technology have proven to be particular hard to implement on top on exiting personal computing operating systems, applications and file systems, which have no notion of the activities users are engaged in. Therefore, we expect that a real well-functioning operating system for interactive spaces cannot be built from existing operating systems and therefore a new generation of operating systems are needed. We would argue that such an operating system could benefit from incorporating support for activity-based computing.

References

1. Bardram JE, Gueddana S, Houben S, Nielsen S (2012) Reticularspaces: activity-based computing support for physically distributed and collaborative smart spaces. In: Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems, CHI '12, pp 2845–2854. ACM, New York, NY, USA (2012). doi:10.1145/2207676.2208689
2. Tabard A, Hincapi Ramos JD, Bardram JE (2012) The elabbench in the wild: supporting exploration in a molecular biology lab. In: Proceedings of the 2012 ACM annual conference on human factors in computing systems, CHI '12, pp 3051–3060. ACM, New York, NY, USA. doi:10.1145/2207676.2208718
3. Tabard A, Hincapi-Ramos JD, Esbensen M, Bardram JE (2011) The elabbench: An interactive tabletop system for the biology laboratory. Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces 2011., ITS2011ACM, New York, NY, USA, pp 301–310
4. Weiser M, Gold R, Brown JS (1999) The origins of ubiquitous computing research at parc in the late 1980s. IBM Syst J 38(4):693
5. Streitz NA, Geißler J, Holmer T, Konomi S, Müller-Tomfelde C, Reischl W, Rexroth P, Seitz P, Steinmetz R (1999) i-LAND: an interactive landscape for creativity and innovation. In: CHI'99: proceedings of the SIGCHI conference on human factors in computing systems, pp 120–127. ACM, New York, NY, USA
6. Johanson B, Fox A, Winograd T (2002) The interactive workspaces project: experiences with ubiquitous computing rooms. IEEE Pervasive Comput 1:67–74
7. Johanson B, Hutchins G, Winograd T, Stone M (2002) Pointright: experience with flexible input redirection in interactive workspaces. In: UIST'02: proceedings of the SIGCHI conference on user interface software technology, pp 227–234. ACM, New York, NY, USA
8. Biehl JT, Baker WT, Bailey BP, Tan DS, Inkpen KM, Czerwinski M (2008) Impromptu: a new interaction framework for supporting collaboration in multiple display environments and its field evaluation for co-located software development. In: CHI'08: proceedings of the SIGCHI conference on human factors in computing systems, pp 939–948. ACM, New York, NY, USA
9. Bardram JE, Bunde-Pedersen J, Doryab A, Sørensen S (2009) CLINICAL SURFACES: activity-based computing for distributed multi-display environments in hospitals. In: INTERACT'09: proceedings of the conference on human computer interaction, Lecture Notes in Computer Science, vol 5727, pp 704–717. Springer
10. Bardram JE (2009) Activity-based computing for medical work in hospitals. ACM Trans Comput-Hum Interact 16(2):1–36
11. Bardram JE, Jeuris S, Houben S (2015) Activity-based computing: computational management of activities reflecting human intention. AI Magazine 36(2):63–72
12. Bardram JE, Doryab A, Gueddana S (2011) Activity-based computing–metaphors and technologies for distributed user interfaces. In: Distributed user interfaces, pp 67–74. Springer (2011)
13. Bardram JE, Houben S, Nielsen S, Gueddana S (2012) The design and architecture of reticularspaces: an activity-based computing framework for distributed and collaborative smartspaces. In: Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems, pp 269–274. ACM
14. Houben S, Tell P, Bardram JE (2014) ActivitySpace: managing device ecologies in an activity-centric configuration space. In: Proceedings of the ninth ACM international conference on interactive tabletops and surfaces. ITS '14ACM, New York, NY, USA, pp 119–128
15. Bannon L, Cypher A, Greenspan S, Monty ML (1983) Evaluation and Analysis of Users' Activity Organization. In: CHI'83: proceedings of the SIGCHI conference on human factors in computing systems, pp 54–57. ACM (1983)
16. Miyata Y, Norman DA (1986) Psychological issues in support of multiple activities. In: Norman DA, Draper SW (eds) User centered system design: new perspectives on human-computer interaction, chap. 13, pp 265–284. Lawrence Erlbaum Associates, Hillsdale, NJ

17. Bergman O, Beyth-Marom R, Nachmias R (2006) The project fragmentation problem in personal information management. In: CHI'06: proceedings of the SIGCHI conference on human factors in computing systems, pp 271–274. ACM
18. Boardman R, Sasse MA (2004) Stuff goes into the computer and doesn't come out: a cross-tool study of personal information management. In: CHI'04: proceedings of the SIGCHI conference on human factors in computing systems, pp 583–590. ACM
19. Gonzalez VM, Mark G (2004) Constant, constant, multi-tasking craziness: managing multiple working spheres. In: CHI'04: proceedings of the SIGCHI conference on human factors in computing systems, pp 113–120. ACM
20. Henderson DA Jr, Card S (1986) Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. ACM Trans Graph 5(3):211–243
21. Christensen HB, Bardram JE (2002) Supporting human activities—exploring activity-centered computing. In: UBICOMP'02: proceedings of the international conference on ubiquitous computing, Lecture Notes in Computer Science, vol 2498, pp 107–116. Springer
22. MacIntyre B, Mynatt ED, Voida S, Hansen KM, Tullio J, Corso GM (2001) Support for multitasking and background awareness using interactive peripheral displays. In: Proceedings of the 14th annual ACM symposium on User interface software and technology, pp 41–50. ACM
23. Winograd T, Flores F (1986) Understanding computers and cognition: a new foundation for design. Intellect Books
24. Roseman M, Greenberg S (1996) Building real-time groupware with groupkit, a groupware toolkit. ACM Trans Comput-Hum Interact 3(1):66–106

Chapter 12
Collaborative Business Process Modeling in Multi-surface Environments

Alexander Nolte, Ross Brown, Craig Anslow, Moritz Wiechers, Artem Polyvyanyy and Thomas Herrmann

Abstract Analyzing and redesigning business processes is a complex task which requires the collaboration of multiple actors such as process stakeholders, domain experts and others. Current collaborative modeling approaches mainly focus on modeling workshops where participants verbally contribute their perspective on a process along with ideas on how to improve it. These workshops are supported by modeling experts who facilitate the workshop and translate participants' verbal contributions into a process model. Being limited to verbal contributions however might negatively affect the motivation of participants to actively contribute. Interactive technology such as smartphones, tablets, digital tabletops and interactive walls can provide opportunities for participants to directly interact with process models. Multi surface environments where different interactive technologies (e.g. display walls, tabletops, tablets, and mobiles) are combined also allow for orchestrating different modes of collaboration. In this chapter we describe an approach that combines different styles of collaboration using various interactive surfaces in a multi surface environment. Testing this approach in three different

A. Nolte (✉) · M. Wiechers · T. Herrmann
Information and Technology Management, Ruhr-University Bochum, Bochum, Germany
e-mail: nolte@iaw.rub.de

M. Wiechers
e-mail: wiechers@iaw.rub.de

T. Herrmann
e-mail: herrmann@iaw.rub.de

R. Brown · A. Polyvyanyy
Information Systems School, Queensland University of Technology, Brisbane, Australia
e-mail: r.brown@qut.edu.au

A. Polyvyanyy
e-mail: artem.polyvyanyy@qut.edu.au

C. Anslow
Department of Computer Science, Middlesex University, London, UK
e-mail: c.anslow@mdx.ac.uk

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_12

settings we found indications that interactive technology not only improves involvement by participants but also speeds up workshops and improves the quality of collaboration outcomes. The studies also revealed means for improving the proposed approach.

12.1 Introduction

Business process management (BPM) can be considered a relevant practice for most organizations. BPM is among the main drivers for organizational development and innovation and organizations commit ongoing and substantial investments in BPM projects which range from 500,000 to 50,000,000 USD per organization according to Harmon [1]. The basis for most BPM projects are graphical representations of processes in process models. They are used to document processes and to analyze and improve them. Process models are also used as training material or as a basis for software development [2]. Creating process models is a complex task because real world phenomena have to be depicted which might include a mesh of activities that are conducted by a number of different actors (c.f. *Goods receipt officer* and *Booking Clerk* in Fig. 12.1 left). In order to depict such processes in a model it is also necessary to translate real life phenomena into elements of a modeling notation and integrate them into a process model which adds to the overall complexity. Modeling notations consist of a set of graphical symbols such as rectangles and ellipses, which represent process parts such as actions, and actors and they also provide rules for how symbols may be combined. Figure 12.1 shows an example for a model of a goods receipt process where the actors are depicted as lanes (booking clerk and goods receipt offices) and tasks as yellow boxes.

Knowledge about a process is usually distributed between different groups of stakeholders and domain experts with each of them potentially having a different perspective on a process. Most stakeholders though are not capable of analyzing and visualizing processes on their own because they lack methodological education and practice both with respect to gathering information about a process and translating that information into constructs of a modeling notation. The latter might

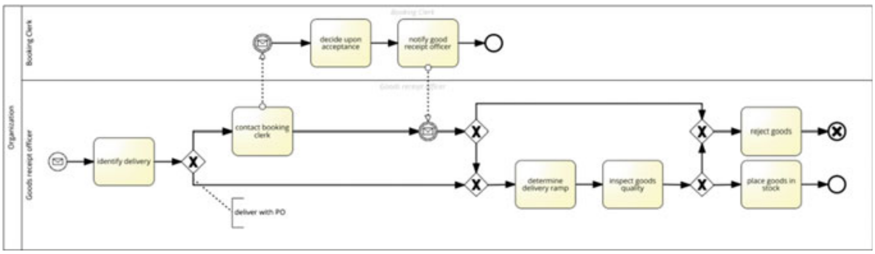


Fig. 12.1 Part of a model for goods receipt process based on the business process model and notation (BPMN [24])

sound surprising at first since modeling languages such as the Business Process Model and Notation (BPMN) were specifically created to be easy to use and understand [3]. Studies on the understandability of process models however show that stakeholders generally are not capable of depicting complex phenomena using a modeling notation without being trained to do so [4, 5]. In order to analyze and visualize processes they consequently require the support of modeling experts who are knowledgeable about a modeling notation and about approaches to analyze processes and improve them. Modeling experts usually come from outside of an organization and are thus hardly knowledgeable about one particular process that has to be visualized in a process model. In order to acquire the information required for process modeling, modeling experts rely on a number of different approaches such as document analysis, interviews, observations, workshops and more. During the course of this chapter we will focus on collaborative approaches since misunderstandings and diverging perspectives about processes become more obvious in a mode of discursive collaboration. This not only leads to a better understanding of a process but also improves the quality of business process models [6–9].

Collaboration in this context usually happens in facilitated workshops where stakeholders and domain experts are supported by modeling experts to analyze and visualize processes in process models and subsequently derive means for improving these processes [6, 9, 10]. During those workshops modeling experts serve as facilitators who organize workshops, guide the communication during the course of these workshops and translate verbal contributions of participants into elements of a modeling notation (c.f. Fig. 12.1). It is common that more than one modeling expert supports a workshop since it is not possible for a single person to guide the communication, translate contributions into a modeling notation and operate a process modeling tool to integrate contributions into a process model [9, 11].

Current workshop approaches are often criticized as being inefficient [12, 13] since they suffer from a number of inherent limitations. Some limitations stem from that all communication has to be channeled through the facilitator since she has to process all contributions before they are integrated into a process model. This effect is commonly referred to as the facilitator bottleneck [13]. Furthermore, limiting participants to verbal contributions potentially leads to a missing sense of participation and a missing sense of ownership of a process model. This in turn might lead to a lack of motivation to participate during a workshop, a reduced buy-in for process changes and a subsequent missing motivation to apply process changes into everyday work practice [13]. Finally, most approaches in collaborative modeling solely focus on participants working together in a single group, while there are indications that collaboration in varying constellations during the course of a workshop cannot only positively influence collaboration outcomes but also the perception of collaboration itself [14–16].

The wide spread of touch enabled devices such as smartphones and tablets alongside the emergence of multi surface environments [17, 18] provides an opportunity to overcome some of the aforementioned limitations. Multiple studies have already shown the feasibility of using interactive technology in the context of process modeling [19–21]. They indicate that the possibility for multiple users to collaborate on large touch devices such as large interactive touch display walls and digital tabletops positively affects collaboration and collaboration outcomes [19, 21]. Alongside these findings there are also indications that personal mobile devices can positively influence collaboration outcomes [15, 22]. Mobile devices can also serve as a means of tying phases of collaboration together by allowing for a seamless switch between phases of collaborating in large groups and phases of working in smaller subgroups [16, 23]. Taking these approaches as a background we propose a concept which aims at **creating a space where interactive surfaces such as smartphones, tablets, digital tabletops and large interactive touch display walls support and facilitate the orchestration of collaboration on business process models**.

The remainder of the chapter is structured as follows. In the following section we describe current BPM approaches highlighting the necessity for collaboration especially during process analysis and re-design using process models (Sect. 12.2.1). Afterwards we provide an overview of how interactive technology can be used in multi surface environments (Sect. 12.2.2). Based upon these reviews we describe a setting for collaborative process modeling in a multi surface environment (Sect. 12.3). Based upon this setting we propose three distinct collaboration styles as well as means of fluid transitions between them before showing three case studies during which different collaboration styles were tested (Sect. 12.4). In what follows we discuss results from these case studies (Sect. 12.5) before providing an outlook on future research (Sect. 12.6).

12.2 Background

During the course of this section we describe BPM approaches and highlight potentials for interactive technology in the context of collaborative modeling. These potentials then serve as a basis for the collaboration styles that are described in Sect. 12.3.

12.2.1 BPM and Collaborative Modeling—Potentials and Pitfalls

BPM is a body of principles, methods and tools to design, analyze, execute and monitor business processes, with the ultimate goal of improving these processes [2]. BPM affects efficiency, effectiveness, and thus competitiveness, of organizations. Companies invest millions of dollars into BPM initiatives and in return obtain the increase in productivity, improvement in quality of service, reduction of operating costs, and faster process cycle times.

BPM initiatives often deal with business process models. A business process model is a specification of business activities, or tasks, and constraints between them that an organization commits to follow to reach its objectives. There are several commonly accepted notations for capturing business process models including BPMN [24], EPC [25], and UML activity diagrams [26]. All of these modeling notations are similar in the sense that they all provide a set of graphical symbols that can be combined with textual labels in order to visualize processes. These symbols cover all aspects of a business process such as actions, actors, resources, decisions and relations [2]. Modeling notations also provide a set of rules of how elements can be combined. Process models are usually created within computer systems that are specifically tailored to support one or multiple process modeling notations (e.g. Signavio[1]).

In order to succeed in volatile business environments, organizations perpetually design new business process models and improve existing ones by re-evaluating customer needs and analyzing real world executions of the deployed business processes. Designing a new or improving an existing business process model is a complex task that often requires different expertise from several domains. Note that business processes usually involve multiple departments within an organization or capture business procedures that involve multiple organizations. Hence, a business process modeling exercise often takes place in a highly collaborative setting such as the ones described in the introduction. The success of collaborative business process modeling largely depends on the quality of methodologies and tools used to guide and support the collaboration as well as the skill of the facilitator [12, 27, 28]. Through a joint creation of BPMN models, EPC models, or activity diagrams, stakeholders acquire shared understanding of operational procedures within their organizations.

The state of the art of collaborative modeling focuses on studies of facilitated workshops [6, 9, 12, 29]. There are a number of different approaches to facilitate such workshops including structured walkthroughs [6], scripts [30], or flexible collaboration patterns [31]. In facilitated workshops, a dedicated person acting in a special role of a facilitator translates individual verbal contributions of process stakeholders into a modeling notation. Workshops are usually divided into phases. During a first phase, aspects of a current as-is process are collected. These parts are then consolidated and aligned to each other in order to form a representation of the current as-is process. Afterwards this visualization is used as a basis to identify means for improving the process and discuss how the process could be altered (e.g. make it more efficient). Once a consensus is reached, the discussed changes are integrated into the process model to form a visualization of a future to-be process [2, 10, 32].

Collaboration support should subsequently fit each of the aforementioned phases. Most of the approaches focus on a single style of collaboration as described above. This leads to the perception that participants perceive workshops as

[1]http://www.signavio.com.

ineffective. They stay mostly limited in their ability to actively interact with and directly contribute to the model that is being created. This may subsequently lead to a missing sense of participation [33] and a lack in the sense of ownership over the artifact developed in the course of the workshop by its participants, which in turn may lead to a lack of motivation to participate during the course of a workshop and may later translate into a weak “buy-in” and reuse of the model. Moreover, facilitated workshops may suppress spontaneous creativity of its participants as all the changes to the model are incremental and are administered centrally by the facilitator [14].

Luebbe and Weske [34] study the use of tangible media in business process modelling workshops. For that they used glass cut outs of BPMN elements which could be labeled using felt pens. They conclude that the use of tangibles by all workshop participants allows them to actively contribute to the model creation process, which leads to more effective process elicitation. In particular, participants of the experiments reported that they get better insights into process modeling. However, this approach solely focusses on single participants eliciting process models while our focus is on collaborative modeling. The approached proposed by Luebbe and Weske also focusses on a single constellation while we aim at supporting multiple collaboration styles in order to address the different phases of collaborative modeling.

12.2.2 Multi Surface Environments in Collaborative Modeling

A major component of engagement on the part of process modeling with stakeholders is the need for tools that provide an appropriate visual to aid in both the cognition of stakeholders using the tool, but also their ability to then communicate their concepts, and to relate information presented to their colleagues in an intuitive and cognitively low overhead manner.

It is all well and good to e.g. provide a large interactive touch display wall which allows users to interact with process models. The possibility to interact with materials on a touch display will not improve engagement of user by itself since they require appropriate visualizations, and support. This is a current topic of research still requiring refinement [35]. There is also research suggesting a need for more flexibility in collaborative modeling workshops thus supporting different means of collaboration [14, 16]. We cover these following issues briefly and focus on how they contribute to the collaboration tasks at hand.

Representations—People understand their domain using particular visual forms that are amenable to the cognitive and work models of the stakeholders (c.f. different visualizations of the same process on the large screen in the top of Fig. 12.2), shown by evidence from cognitive fit experimentation [36]. Not only is this effect evident from a theoretical analysis of representational approaches [37] but also from

Fig. 12.2 Multi surface environment: variation of different visualizations on different devices such as large interactive touch display walls (*top*), digital tabletops (*bottom left*) and personal mobile devices (*bottom right*) in collaborative modeling

user habituation, which has formed trained constructs that are easily understood using the visual language of that stakeholder's domain [38].

Relationships—Just placing items of visuals besides each other in temporal sequences is not necessarily useful to the process of collaboration; explicit relationships between the domain information must be added to aid in communicating these concepts between the stakeholders [35]. In previous work, we have analyzed multi-domain visualization in a 3D sense for manufacturing, juxtaposing process information with other engineering data, providing relationship disambiguation as part of the design [39]. We propose that touchscreen process modeling frameworks will allow other data in the form managed by diverse stakeholders in management (e.g. Bill Of Materials (BOM), accounts, IT operations) and physical operational representations (e.g. 3D workplace representations [40]) to be displayed, and related to each other, side by side, in order to assist in discussion and collaboration.

Scale—The use of large display walls provides room to show both relationships and context of information presented to stakeholders, allowing people to gather around the representations for analysis. As well as collaborating on one representation, large interactive touch display walls allow people to move easily between representations, without the cognitive overhead of multiple displays on machines causing loss of context via excessive eye movements [41]. As well, the size of the representations has an immersion effect by filling the visual field and engaging the viewer's senses more strongly [42, 43].

Flexibility—there are situations in collaborative modeling where a single large display visualization is not sufficient since participants have different interests with respect to different parts of a process [44]. It is thus necessary to provide a setting, which supports different constellations with respect to collaboration. These constellations have to cover working in solitude as well as working in smaller subgroups and working in a single group together [14, 16]. In these settings smaller touch enabled devices such as tablets or smartphones appear to be more reasonable (c.f. Fig. 12.2 bottom right).

Styles of collaboration—several further dimensions influence the collaboration between the participants. Aspects of time matter: the usage of the tools within a multi surface environment depends on the length of a meeting. The shorter the meeting the less effort can be invested to switch between various tools or to organize several cycles of collecting ideas and refining ideas. It might be the case that collaboration within the whole group is put into the foreground of individual work, which can even include work results, which have been helped outside of the meeting room. The choice of how participants collaborate depends on the decision whether a workshop will focus on divergence or convergence of ideas and contributions [14]. In the latter case it is important that all participants are aware [45] of which decisions have been taken and how they are represented in a model. Furthermore, the size of the collaborating group decisively influences its decision of how to use the media of a multi surface environment and of how to switch between them. Another important difference refers to the question whether the participants' individual work and contributions always take place in a public space and are immediately visible to others, or whether they are prepared in a private space and an explicit decision is needed to make them available to others [22]. Figure 12.2 shows an example of how different views on different devices can be combined in a multi surface environment. Similar settings have explored oil and gas exploration [17], emergency management [46], geospatial interaction [47], software visualization [48], and software development team meetings [49].

Such issues bring up a rich set of research questions with regards to both the social and technical aspects, that need to be addressed in order to fully utilize the novel affordances of such constellations in process modeling. Example questions can be:

1. What is the optimal combination of representations to use in such a process modeling scenario?
2. What is the optimal relationship representation that can be developed to ease interactions between team members of different domains?
3. What is the optimal combination of collaboration styles for each phase of the modeling process?

12.3 Collaborative Modeling in a Multi Surface Environment

Based on the previously described review we will propose a concept for collaborative modeling in a multi surface environment. The concept includes three styles of collaboration based on an environment that combines different interactive surfaces such as large interactive touch display walls, digital tabletops and personal mobile devices. For each of these styles we will describe how they work in the proposed setting and for which specific aspects of collaborative modeling they may be useful. We will describe how these styles can be intertwined and how the described setting as well as the different collaboration styles affects facilitators and participants.

12.3.1 Environment

Supporting collaborative process modeling we propose a setting where different interactive devices are placed within a single room thus creating large multi surface environment that allows multiple users to simultaneously interact with different representations of a process model using different devices or interactive surfaces [50–52]. These devices include large interactive touch display walls as well as digital tabletops and personal mobile devices such as tablets and smartphones. Participants can interact with process models using touch gestures which were derived from previous research into the use of touch gestures for process modeling [20]. The underlying design rationale was to create interfaces that are easy to use and fast to learn. A comprehensive overview of gestures for business process modeling used can be found in Nolte et al. [53]. Furthermore, we made sure that the appearance of the interface as well as the ways on interacting with the displayed materials is identical for all devices.

In order to allow simultaneous interaction with models using different devices we created an application where the model itself resides on a server, which handles all changes to a process model. The software provides means of concurrency control to ensure that conflicting actions by different participants at the same time cannot result in corrupted models [54].

12.3.2 Styles of Collaboration

In what follows we will describe three distinct styles of collaboration for process modeling in a multi surface environment. These styles are based on limitations of current approaches and serve as an example for how collaborative modeling can benefit from interactive technology in multi surface environments. In addition, we will describe an approach to support switching between different collaboration styles.

Collaboration Style 1: Parallel contributing by individual work

In this collaboration style, participants contribute to a process model in parallel using a personal mobile device (e.g. smart phone or tablet). Contributions are integrated into a process model which can be—but need not—immediately displayed on a large interactive touch display wall and are thus visible for all participants during the whole course of a workshop. Succeeding, contributions can be collaboratively altered or combined using the large interactive touch display wall (c.f. Fig. 12.3). It is not possible to alter or delete elements using personal mobile devices since we perceived it as valuable to be able to collaboratively discuss all contributions by all participants.

Collaboration style 1 is especially suitable for the early phases of process documentation where parts of a current as-is process are collected. During this phase it is common practice to document a process from its start to its end before identifying means of altering or improving the process. This activity can be very time consuming in a workshop setting where participants can only contribute verbally since all contributions have to be picked up by the facilitator, translated into elements of a modeling notation and integrated into the process model. During this phase of divergence, only a few participants are active at the same time since and not every participant is knowledgeable about or interested in all aspects of a process.

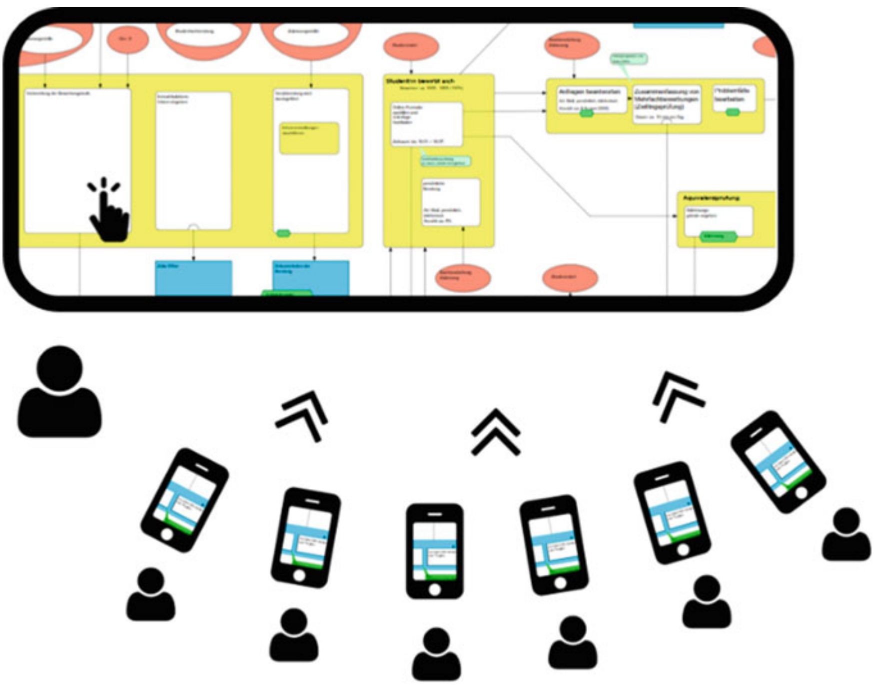


Fig. 12.3 Participants contributing to a process models using personal mobile devices

Allowing participants to contribute in parallel using personal mobile devices can increase the efficiency of collecting relevant process parts significantly [15].

Collaboration style 1 is not only suitable for phases where a process is documented. It can also be feasible to allow for parallel contributions during phases where ideas have to be developed on how to improve a process. In a classical workshop setting participants would have to wait for other participants to state their respective ideas which can for example lead to production blocking [55]. Production blocking describes an effect that occurs when someone cannot express an idea directly but has to wait for her turn to speak. This can result in that person forgetting the respective idea or altering it in a way that it fits the contributions of others. Parallel contributions via mobile devices can potentially overcome this effect. Contributions via mobile devices can also reduce the fear of being evaluated by others (evaluation apprehension [56]) since ideas do not have to be expressed verbally but can instead be contributed anonymously through a personal mobile device.

Collaboration Style 2: Collaboration in smaller sub-groups

In this collaboration style, the whole group of participants is split into smaller sub-groups which collaborate using larger interactive devices such as digital tabletops or large interactive touch display walls (c.f. Fig. 12.4). This style is similar to

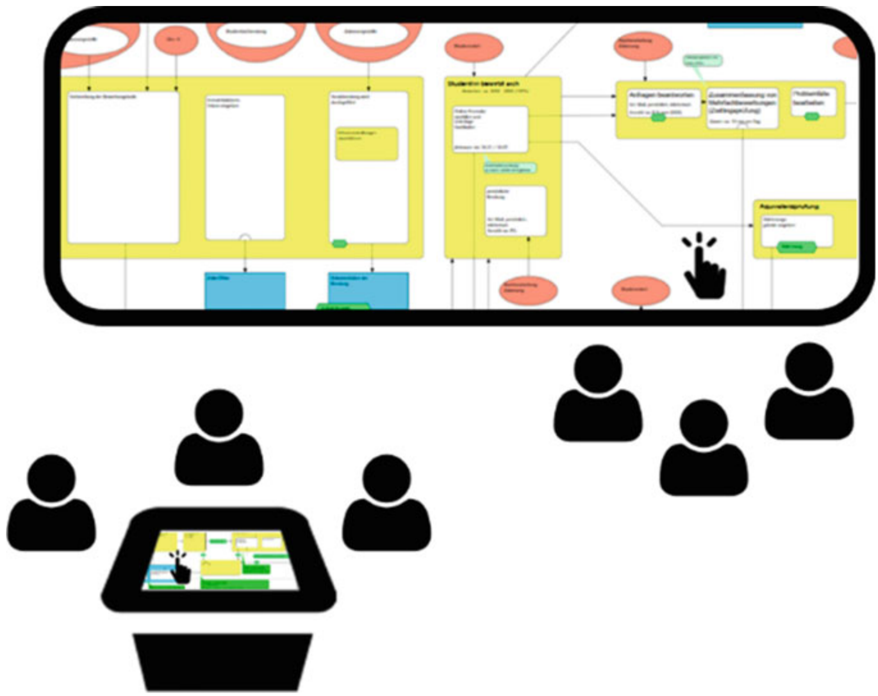


Fig. 12.4 Participants collaborating in smaller sub-groups using digital tabletops (*bottom left*) or a large interactive touch display wall (*top*)

the previous one since all participants can contribute in parallel but instead of using one input device each, they now share an input device within a small group of e.g. two to four participants. In this style it is also possible to alter existing model elements, combine them or delete them using the respective touch interfaces.

This collaboration style supports situations in which it is possible to build on a prepared process model, or in situations where process parts have already been collected. This style allows for sub-groups of participants to discuss aspects of a process they are interested in. Discussions can focus on identifying means of supporting certain process parts by IT or on discussing details about how collaboration within the process could be improved. The main reason for dividing one large group into smaller subgroups in a workshop setting is that—as discussed earlier—not all participants are knowledgeable about or interested in the same aspects of a process. This style allows participants to form interest groups that can focus on certain aspects of a process in parallel thus potentially increasing workshop efficiency. A facilitator in this context can serve as an initiator for those phases and she can serve as a modeling expert if certain participants struggle in expressing their ideas using a modeling notation.

Collaboration Style 3: Collaboration in a group together

This collaboration style is similar to a typical workshop setting where the participants collaborate in the group together. This style is suitable for phases of convergence where e.g. previously gathered process aspects are combined into one large process or where different ideas on how to alter a process are discussed. However, while in other settings, the participants are limited to verbal contributions, this setting allows them to modify the process model at any point in time using a touch interface on a large interactive touch display wall (c.f. Fig. 12.5). Similar to the previously described collaboration style, all participants can contribute in parallel but this time they all have to share the same device which is a large interactive touch display wall instead of a digital tabletop or tablet in order to support larger group sizes. Here it is again possible for all participants to alter the process model in various ways using the touch interface on the display wall. This includes adding elements, altering them, putting them into relation with each other and deleting them.

Similar to the previous collaboration style, this style aims at phases during the course of a workshop where parts of a process model already exist that have to be consolidated. This style supports exchanging perspectives of all participants, discussing different views and ultimately reaching a common understanding about the process as a whole. The latter is especially important since the previously described collaboration styles did not allow for participants to reach a common understanding throughout the whole group since they were either working in solitude (c.f. collaboration style 1) or in smaller sub-groups (c.f. collaboration style 2). Reaching a common understanding about a process is a prerequisite for reaching a consensus [57] about future process changes or at least an acceptance for compromises. Allowing participants to alter the process model at any point in time using large interactive surfaces potentially improves the motivation of participants to actively

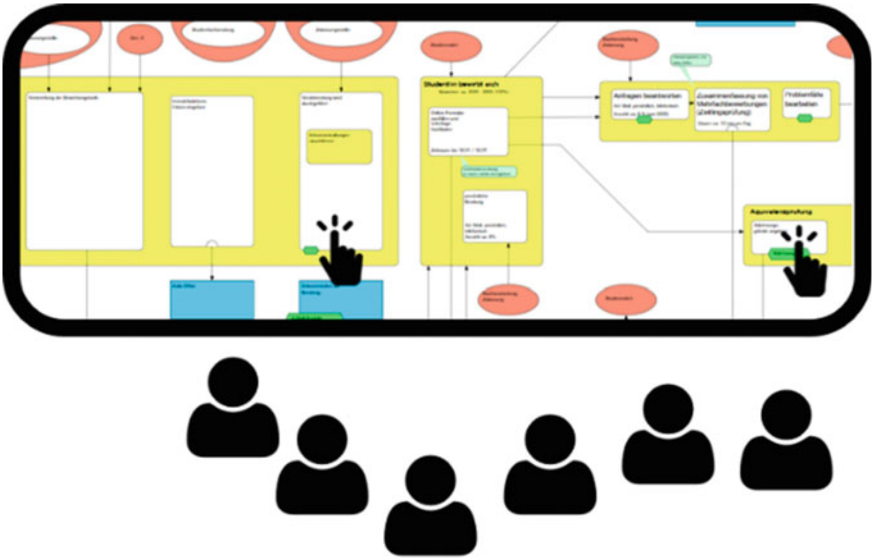


Fig. 12.5 Participants collaborating in a single group using a large interactive touch display wall

participate during the course of a workshop. Allowing participants to alter process models themselves also potentially increases their motivation to implement changes to a process that were discussed during a workshop [13, 28].

Intertwining different styles of collaboration

It is not sufficient to work with one of the previously described collaboration styles alone. It is rather necessary to intertwine them on demand. Since all devices used in our setting are connected to each other, it is easily possible to switch between different styles on demand [16]. The only requirement is to distribute an URL among the participants alongside their respective user credentials. Changing between styles can even be simplified by for example using personal mobile devices that are equipped with a camera. With support of a special app, a participant can then simply take a picture of a part of a model that they are interested in and the system could open the corresponding model and navigate to the part that was photographed [23].

The possibility to change between different collaboration styles aims at improving the flexibility of collaborative modeling workshops while providing participants with means to actively influence the content of process models.

The styles described before should not be considered as the complete spectrum of possibilities. It would also be possible to for example allow participants to continue contributing process parts (collaboration style 1) while others start consolidating the already existing elements (collaboration style 2).

12.3.3 The Role of the Facilitator and the Role of Participants in Collaborative Modeling

Current facilitation concepts in collaborative modeling focus on the facilitator being in charge of running a workshop, keeping track of its goals and subsequently managing the communication throughout a workshop. The facilitator is the central point of interaction with the process model throughout the entire workshop. The facilitator picks up verbal contributions by workshop participants, translates these contributions into elements of a modeling notation and integrates them into a process model.

The previously described collaboration styles (Sect. 12.3.2) still require the facilitator to be in charge of running a workshop and keeping track of its goals. The facilitator though will not have to continuously keep track of all communication and is no longer the only person interacting with a process model. Instead the facilitator will have to focus on guiding a workshop thus orchestrating different collaboration styles. This includes deploying different means of collaboration and deciding when participants should come back together after phases of parallel contributions and collaboration in sub-groups. The facilitator will still be required to serve as a modeling expert in certain cases. She will however not be required to make all changes to a process model since the participants can alter the process models on their own. The facilitator will rather serve as a modeling expert in cases where the participants cannot decide on how to depict certain aspects of a process in a model.

The role of the participants of a workshop also has to change. Since they are no longer limited to verbal contributions they have to learn how to use the interfaces proposed for the different collaboration styles. They also have to become proactive as it is necessary for them to choose a means of interacting with a process model that reflects their modeling expertise. They have to be able to determine when they require additional information by other participants (e.g. while working in sub-groups) thus supporting the facilitator in orchestrating collaboration.

Taking the aforementioned aspects together there has to be a shift of responsibilities. While the participants have to take more responsibility with respect to actively shaping a process model, the facilitator has to focus more on becoming a guide rather than being responsible for all changes to a process model throughout the whole workshop. Interactive surfaces provide opportunities for these changes to happen.

12.4 Case Studies: Collaborative Modeling in Multi Surface Environments

In the previous section we proposed three distinct collaboration styles alongside means to switch between them. In this section we will now describe examples of how we applied these styles in practice. We will report on the setting and the procedure as well as effects on the role of the facilitator. The examples serve as a proof of concept and a basis for deriving ideas on how to potentially improve collaborative business process modeling in multi surface environments.

12.4.1 Integrated Brainstorming

During the course of a project that aimed at supporting elderly people to live in their own homes for as long as possible we were faced with the task of designing a service where elderly people are accompanied during their weekly shopping. The service should be ordered using paper based forms that allowed for the ordering person to suggest other elderly people that would participate in a shopping trip. The service and the underlying process had to be designed from scratch since there was no process to build on in the first place.

We conducted multiple workshops where future stakeholders and domain experts jointly developed a model of a process that would then be used to establish the respective service. During these workshops we conducted multiple brainstorming phases and combined them with phases during which brainstorming contributions were clustered, discussed and aligned with respect to a process sequence [15, 22]. During the brainstorming phases each participant was given a tablet which they could then use to access an interface that the participants can use to contribute text (c.f. Fig. 12.6 top). The contributions were then automatically transferred into elements of a modeling notation and integrated into the process model (c.f. Fig. 12.6 bottom right).

After each brainstorming phase the facilitator of the workshop clustered the contributions by asking the participants whether or not an element fit at a certain

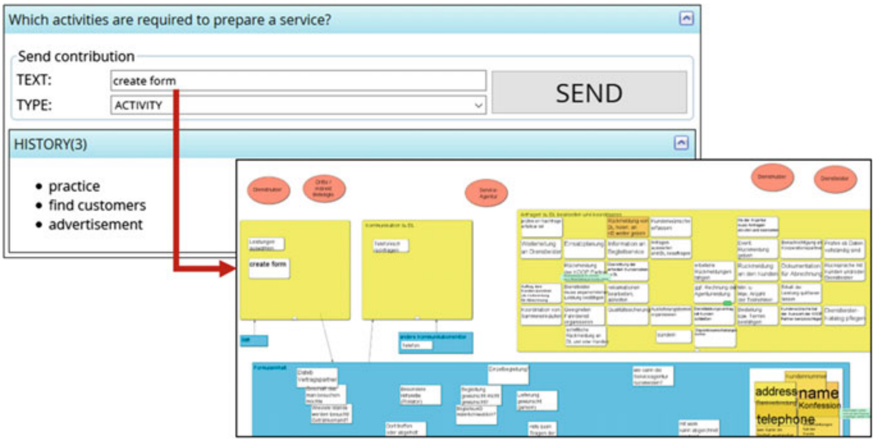


Fig. 12.6 Textual contributions (*top*) are transferred into elements of a modeling notation (*bottom right*)

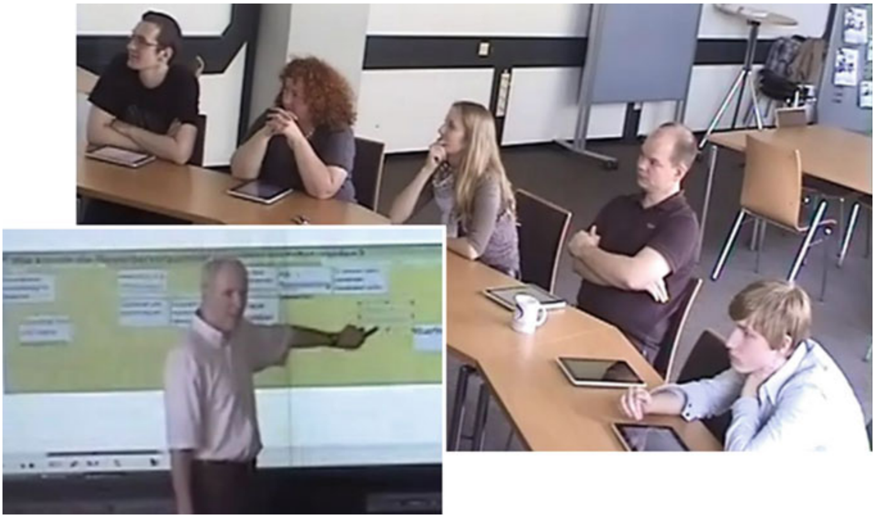


Fig. 12.7 Facilitator (*bottom left*) and participants (*top right*) during the course of a workshop

position (c.f. Fig. 12.7). If needed the facilitator changed the type of an element created new ones or altered existing ones to create clusters or created relations between the elements. During this phase the facilitator used an interactive touch display wall, which allowed them to move elements around, delete them or create new ones using touch gestures.

The workshops lasted about 2 h each. During those workshops we conducted 3 brainstorming phases of about 7 min. After each of these brainstorming phases we had a clustering phase, which lasted around 30 min each. In total we invited 11 participants. Their heterogeneity covered aspects such as gender (5 female, 6 male), age (range: 26 to 57 years), status (students, postdocs, research assistants, full professors, practitioners) and professional background. Some of them were involved as academics in the research on process design while others worked in nursing homes or as service providers. The participants were guided by a facilitator who was supported by another modeling expert who could operate the modeling tool on demand if something went wrong during the session for example with respect to the responsiveness of the touch interface. The workshops were videotaped and we tracked contributions by participants and interactions of the facilitator with the interface. After the workshops we conducted interviews with selected participants as well as the facilitator aiming at getting an insight into their experiences during the course of the workshops.

This setting thus combines collaboration style 1 with a phase where the facilitator asked the participants about their contributions and alters the model herself. During the brainstorming phases the participants could contribute in parallel while they were limited to verbal contributions during the following clustering phase. The facilitator was the only person that used the interactive touch display wall to move elements around, delete them, create new ones or put them into relation with each other. Our subsequent evaluation of the workshops provided indications that the possibility to contribute in parallel helped in overcoming some of the negative effects that are associated with typical workshop settings such as production blocking and evaluation apprehension [56]. Participants reported that the setting did not only allow them to develop their own flow of ideas. They also reported a strong sense of participation and mentioned that being able to contribute at any point in time fostered motivation. We also found indications that the participants developed a sense of ownership for the contributions since all of them were discussed, considered and integrated into the final process.

The setting also had some inherent limitations. Despite allowing the participants to directly contribute to a process model and thus become more active during parts of a workshop, it was not possible for them to alter or enhance their contributions in any way with their personal device. They were still dependent on the facilitator to carry out these tasks. Furthermore, participants could only contribute directly for short periods of time during the course of a workshop (about 20 min). Most of the time they were still limited to verbal contributions, which limited their flexibility of the participants to contribute at any point in time. The facilitator also reported some limitations with respect to the setting with the major one being that they found it hard to switch between phases. It always took some time for the participants to realize that they should stop contributing ideas and refocus on the facilitator.

All in all, we have to conclude that allowing parallel contributions by participants had a positive effect on collaboration mainly with respect to the participants feeling more involved and being more motivated to contribute. There are some limitations to this setting especially with respect to contributions only being possible at certain times. Furthermore, switching between phases should be improved.

12.4.2 Selecting Sections of Process Models by Taking Pictures

We developed a system that facilitates a seamless transition from working in one group to working in smaller breakout groups within the context of collaborative modeling workshops [16]. The system allows participants to alter a process model using a browser on a mobile device. The interface on the devices is coupled with the interface on a large interactive touch display wall which shows the same process model. In order to access a certain position within that model, the participants can use the camera on their mobile device, take a picture of the desired part of the process model. The system then automatically detects the correct process model, opens it on the mobile device and navigates to the detected position. Afterwards the participants can alter the process model using the interface displayed on their mobile device (c.f. Fig. 12.8). The system provides awareness features as it indicates the view port of other users that are currently connected with the model

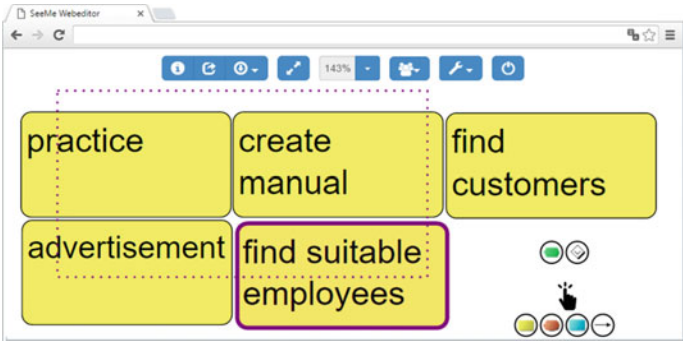


Fig. 12.8 Web based interface with flexible onscreen menu (*bottom right*). The *dashed area* shows the viewport of another user who also has selected an element (c.f. element labeled "find suitable employees")

(c.f. dashed rectangle in Fig. 12.8). The system also includes means of concurrency control as it locks elements that are selected by one user for all other users (c.f. element labeled "find suitable employees" in Fig. 12.8).

The system is operated via a touch interface that is based on gesture recognition. The system supports pinch gestures to zoom and stroke gestures to move the viewport of the model. Altering and creating elements is done by selecting the desired action within a location based menu that is activated by a single touch (c.f. Fig. 12.8 bottom right). In order to create an element a user can tap at any position on the screen and drag the respective element out of the menu that appears on the first tap. In order to use the system, the actors thus have to be knowledgeable about the modeling notation used.

In order to test the usability of the system and to identify means for improvement we conducted a study. The study was based on a workshop where 6 participants (5 male, 1 female) acted as process participants. They were asked to improve a prepared model that showed the process of how a renter should deal with a broken water line. All of the participants were knowledgeable about the modeling notation used and they were familiar with the process in question. The model of the process was purposely vague and contained errors both with respect to the spelling of certain model elements as well as with respect to process related aspects such as a wrong sequence.

The workshop lasted about 2 h and was divided into three phases. During the first phase a facilitator explained the model to the participants and they jointly decided on tasks that should be performed during the course of the following phase. Some tasks were very simple tasks (e.g. correcting spelling errors) while others were more complex (e.g. extending certain aspects of the process). During the following phase—which lasted for about 70 min—the participants split up into 3 groups and started working with the model. Using one tablet per group they took pictures of the areas of the model in which the respective task should be carried out and then started working on it using the interface displayed in Fig. 12.8. After each group had finished their respective tasks they came back together and discussed the respective changes each group had made to the model. This phase lasted for about 18 min and was guided by the facilitator. The setting thus covers the aspect of intertwining different collaboration styles while including collaboration style 2.

After the workshop we conducted a group interview and both participants and the facilitator were handed questionnaires after the workshop. The questionnaires aimed at assessing the perception of the participants on the system and the setting. They covered aspects such as whether or not the system allowed the participants to be more active during the course of the workshop, whether or not the system fostered discussion among them and whether or not the system increased the efficiency of the workshop. The participants had to rate each aspect on a scale of 1 (“strongly disagree”) to 5 (“strongly agree”). The questionnaire for the participants also contained questions based on the heuristics described by Nielsen [58] in order to assess the usability of the system.

Evaluating the questionnaires and the group interview we found that the participants as well as the facilitator perceived the system to support workshop participants to be more active during the course of a workshop (likert scale: median 4 out of 5) thus increasing their sense of participation. With respect to whether or not the system fostered discussion among the participants the verdict was not so clear (median 3). We thus assume that not all participants perceived the system to foster discussion. We found a similar situation with respect to whether or not the system increases the efficiency of a workshop. A median of 3 out of 5 indicates that some participants perceived the system to increase the efficiency of the workshop while others did not. Furthermore, the participants positively rated the usability of the system (median 4 out of 5) with the exception of how the system handled errors (median 2).

During the subsequent group interview the participants positively mentioned the possibility to use pictures as a means for navigation within a process model. They said that they found it “surprisingly useful” and that it allows for a “seamless transition” between working on the large display wall and on a tablet. They also positively mentioned the previously described awareness features. They said that it was “easy to follow what others are doing” and that the features were “not distracting”. The participants however mentioned that keeping track of the tasks was difficult since there was no indication in the model itself what the task was. They thus had to keep track of the tasks themselves. The participants also mentioned that operating the interface on the small tablet display was hard sometimes especially when operations had to be conducted that require a certain precision such as connecting elements through arrows.

The facilitator mentioned during the subsequent group interview that different group speeds could potentially be hard to handle during the course of a workshop. She thus suggested for the system to allow groups to hand over tasks to others.

All in all, we can conclude that while the system did not bring a considerable advantage with respect to speed, the participants as well as the facilitator perceived it to allow them to become more active during the course of a workshop. Furthermore,

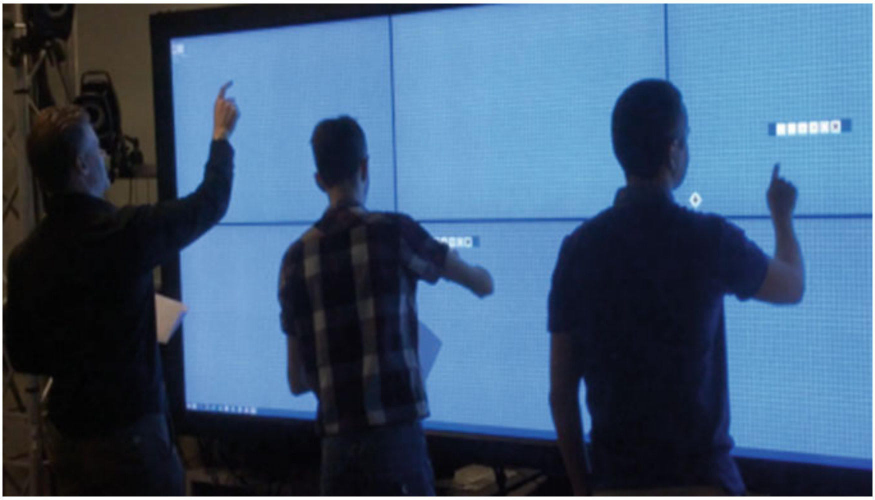


Fig. 12.9 Three workshop participants collaboratively using the CubeBPM system

the photos allowed for a seamless transition between working in one large group and working in smaller subgroups. The system still requires some improvements with respect to usability (error correction as well as handling small elements on a touch display) as well as with respect to supporting group dynamics (e.g. tying tasks and model changes together as well as handing over tasks between groups).

12.4.3 CubeBPM—Collaborative Modeling on Interactive Large Display Walls

Aiming at assessing how large interactive touch display walls can influence collaborative process modeling, we developed the CubeBPM system [53, 59]. CubeBPM allows multiple actors to draft models collaboratively using a large interactive touch display wall (c.f. Fig. 12.9) thus providing them with direct access to process models and the possibility to directly manipulate them. The system runs on a large single integrated touch display,[2] which consists of 6 almost seamlessly, connected panels (2 rows by 3 columns, c.f. Fig. 12.9). The tool can run on large segmented displays via synchronized and networked hardware, to produce a highly scalable solution to cover large wall display systems (e.g. QUT Cube [60]).

[2]See CubeBPM demo video: https://youtu.be/OuEHsL9vCR8.

CubeBPM implements the majority of the control perspective BPMN[3] grammar including: swim lanes to represent actors in processes, gateways to represent decision points, activities, and event types [24]. CubeBPM is operated via an interface that is based on touch gesture recognition. The gestures used were devised from previous research into the use of touch gestures on digital tabletops for process modeling [20]. The underlying design rationale was to create an interface that was easy to use, fast to learn and that could be used by multiple actors who work at the same model in parallel. We thus focus on simple touch gestures (e.g. crossing an element out to delete it or drawing a line between two elements using two fingers to connect them to one another). The system also offers location-based flexible menus that provide actors with basic modeling functions at disparate locations. These menus are accessible via double tapping (c.f. Fig. 12.9 top right). In order to create an element, actors have to select the elements they want to create and drag the element out of the menu to the screen (c.f. [53] for more information on the system and the gestures used). Using CubeBPM requires actors to be knowledgeable about the modeling notation used (BPMN) since there are no functions implemented that relieve actors from the necessity to translate their contributions into elements of a modeling notation.

In order to test the feasibility of the CubeBPM we conducted preliminary studies during which 3 groups of 4 participants were asked to create a process model based on a textual description. All of the participants were graduate students that were attending a class on BPMN and they were thus knowledgeable about the modeling notation used. The process in question is the procedure of shopping in a retail store. Each experiment was set to last for about 30 min with an additional preparation time of roughly 10 min. During this preparation the facilitator showed the participants how to operate CubeBPM and gave them some time on their own to familiarize themselves with the system. Then the facilitator opened a predefined model that contained all elements necessary to model the described process and asked the participants to assemble them so that the model fits the description, which required them to alter the sequence of elements by moving them around and connecting them to one another. The participants were allowed to add elements when they feel it is necessary. We provided the participants with a predefined set of elements, as typing in text is time consuming on vertical display walls. The facilitator only served as a guide who made sure that the participants followed the pre-planned procedure of the workshop. The facilitator also supported the participants when they had questions relating to the usage of BPMN as a process modeling notation (e.g. how to visualize a certain process step within the model). Each workshop was videotaped and we tracked interactions with the CubeBPM interface. Afterwards we coded the videos using the free tool ELAN.[4]

The previously described setting can be considered an incarnation of collaboration style 3 and can thus serve as an example for a convergence phase. All

[3]http://www.bpmn.org/.

[4]https://tla.mpi.nl/tools/tla-tools/elan/.

participants could alter the process model at any time in any way they saw fit using a touch gesture based interface on a large interactive touch display wall. Since all participants were knowledgeable about the modeling notation used the facilitator only had to guide them through the course of the workshop. The facilitator did not have to assist them with respect to using the modeling notation.

Analyzing the material gathered during the course of the workshop we found that almost all participants used the touch interface provided by CubeBPM in order to alter the model. Changes to the model mainly focused on altering the sequence of elements which included moving them around on the screen and connecting them. Sometimes participants also created new elements.

The extent to which single actors used the touch interface expectably differed hugely between individual participants. Some participants used CubeBPM extensively on their own while others only rarely altered the model. We also found occasions during which participants asked others to carry out changes to the model rather than doing it themselves. The participants that carried out the changes thus took over duties that are normally associated with the role of a modeling expert. How the participants used CubeBPM and whether or not they used it at all was entirely left to them. Considering the aforementioned observations that almost all participants did use CubeBPM themselves together with the fact that they were not obliged to do so consequently leads us to the assumption that CubeBPM positively influenced the motivation of the participants to actively alter a process model and thus to participate in process model development.

With respect to collaboration we found all possible kinds of different constellations among the groups. Sometimes all participants worked together while it also happened that they split up in groups of two or that a single participant left the group to work at a different part of the model while the other participants stayed together. There even were occasions where all four participants worked individually on different parts of the process model. Changes between different group constellations happened on demand without explicit coordination.

Regularly different participants altered the process model at the same time. These changes however were all independent to one another. It never occurred that participants interacted between different groups (e.g. handed over elements to another group or another participant). Furthermore, we also observed participants stopping discussions when other participants made changes on a different part of the process model. This leads us to the assumption that modifications are noticed even by participants that do not contribute to the modifications directly.

We also observed large differences between the different groups. While one group stayed together for almost the entire course of the workshop, another group only did so for about 50 % of the time. During the remaining time they mainly worked in pairs or in a group of three with a single participant working on a different part of the process model at the same time. These differences in the way participants collaborated also had a profound effect on the time it took them to assemble the process model. The group that only stayed together for about half of the time was twice as fast as the group that stayed together for almost the entire time. This difference cannot entirely be attributed to the way they collaborated but it provides an indication that working in flexible group constellations can have a positive effect on workshop efficiency.

Our analysis pointed out some limitations of this way of collaborative process modeling using CubeBPM. First participants have to be knowledgeable about the modeling notation used. This was not a big problem during the course of our study since the participants all had used the modeling notation before and since the process did not require them to create complex structures. There however were occasions during which participants asked the facilitator whether or not they had used the modeling notation in the correct way. We expected this to happen more often when the complexity of the models increased. Second we found a huge gap with respect to activity of participants during the course of the workshop. Some participants were active almost all the time while others rarely contributed (verbally or directly). This behavior can at least partly be attributed to the fact that it was entirely left to the participants whether or not they wanted to contribute.

All in all, it can be stated that allowing participants to alter a process model using a large interactive touch display wall affects the way they collaborate. The setting affords participants to actively contribute to process modeling and affects the way they collaborate since it allows for different groups to form on demand. The setting potentially requires more guidance by a facilitator since not all participants contributed or could contribute equally.

12.5 Discussion

The previously described case studies provide indications for positive as well as negative effects of using interactive technology in different collaboration styles in the context of collaborative business process modeling.

First, we found all styles to increase the perceived **efficiency** of a workshop. This can partly be attributed to the fact that participants were not limited to verbal contributions. They rather could directly interact with the process model in all of the styles which subsequently eliminated the facilitator bottleneck. We thus assume that using interactive technology positively influences the participants' perception of efficiency.

Second, we found all collaboration styles to **increase the sense of participation** for the participants which positively influenced their **motivation to participate** during the course of a workshop. This again can mainly be attributed to the fact that all participants could directly alter the process model at any point in time. It should however be noted that it was not possible for participants to alter contributions in collaboration style 1. This was perceived as being not adequate by the facilitator and the participants alike.

Third, we found for collaboration style 3 to increase the **sense of ownership** for the process model. We did not find indications for this during collaboration style 1 and 2. This might be attributed to the fact that a sense of ownership for the process model as such can only be developed when:

- A common understanding about a process is reached,
- Participants agree about how to alter a process and
- Participants feel that their input has been valued and considered.

During collaboration styles 1 and 2 it is only marginally possible to reach a common understanding or agree to changes to a process since every participant only works on certain aspects of a process. In collaboration style 3 however it is possible to reach a common understanding about the process as a whole and agree on changes.

It was not only the possibility to directly interact with all parts of a process model that positively affected the perception of collaboration of the participants. There was also the possibility to focus the attention to those **aspects of a process model that participants were interested in**. This became evident while testing collaboration styles 1 and 2. Both styles allowed switching between focusing on the large touch display wall in order to gain an overview and focusing on smaller devices in order to work on specific aspects of a process model. While testing collaboration style 3 we found multiple occasions during which participants collaborated in different constellations on different parts of a process model that they were interested in.

There were also some **drawbacks** with respect to the different collaboration styles which can subsequently serve as a basis to improve the concept. First we have to note that despite the possibility to alter a process model at any point in time **some participants remained passive**. This became especially evident in collaboration style 3. Some participants decided to not interact with the displayed process model despite the possibility to do so. A facilitator who particularly asks those participants to contribute can potentially help in these situations.

Furthermore, we found **coordination** to be an issue for different settings. The facilitator reported that it was hard to decide when to bring groups back together. The participants sometimes found it hard to identify which tasks had been assigned to them and which had been assigned to a different group. This leads us to the conclusion that the system should provide better support for coordination between participants as well as between participants and the facilitator.

The previously described studies also have some inherent **limitations**. First each collaboration style was tested individually and only in one setting. Furthermore, the number of participants as well as the tasks and time for the workshops varied between each setting. This limits the generalizability of the results. Finally, the interfaces used were not the same for each style.

Despite these limitations the studies can still serve as a prove of concept that the proposed collaboration styles can positively influence collaborative modeling workshops We found indications that the collaboration styles and their respective setting indeed have a positive effect on the efficiency of workshops. We also found the settings to increase the participants' sense of participation and ownership, which potentially affects their motivation to contribute during workshops. It can thus be stated that multi surface environments can positively affect collaboration and collaboration outcomes in a context of collaborative process modeling. There is however a necessity for future studies especially with respect to intertwining different collaboration styles within a single workshop in order to further explore the potential of multi surface environments in the context of collaborative modeling.

12.6 Conclusion and Outlook

Current approaches in collaborative process modeling are strongly dependent on a facilitator and limit participants to verbal contributions. This subsequently limits collaboration among participants and potentially affects the resulting process models in a negative way. We identified interactive surfaces and multi surface environments as a way to overcome the limitations of current workshop approaches and presented an environment that aims at allowing participants to become more active during workshops. We proposed three distinct collaboration styles and tested each of them individually. Results from the studies provided indication that interactive technology potentially improves involvement by participants, speeds up workshops and subsequently improves the quality of collaboration outcomes. We also identified means of how to improve the proposed approaches mainly with respect to approaches to intertwine them.

In the future we are planning to conduct further studies on the impact of interactive technology on collaborative modeling workshops. We are particularly interested in how they change collaboration among participants. We aim at identifying patterns of collaboration that can subsequently be supported in multi surface environments. The concept should be extended to support a larger variety of ways to collaborate. We will continue to explore different ways of facilitation thus aiming at creating a more effective collaboration between facilitators and workshop participants.

References

1. Harmon P (2016) The state of business process management. Business process trends
2. Dumas M, La Rosa M, Mendling J, Reijers HA (2013) Fundamentals of business process management. Springer
3. Allweyer T (2009) BPMN 2.0, Business process model and notation. Books on demand, Norderstedt
4. Mendling J, Strembeck M, Recker J (2012) Factors of process model comprehension—findings from a series of experiments. Decis Support Syst 53:195–206
5. Reijers HA, Mendling J (2011) A study into the factors that influence the understandability of business process models. IEEE Trans Syst Man Cybern, Part A. 41:449–462
6. Herrmann T (2009) Systems design with the socio-technical walkthrough. In: Whitworth B, de Moore A (eds) Handbook of research on socio-technical design and social networking systems. Idea Group Publishing, Hershey, pp 336–351

7. Hoppenbrouwers S, Proper H, Weide T (2005) Formal modelling as a grounded conversation. In: Proceedings of the 10th international working conference on the language action perspective on communication modelling (LAP05), pp 139–155
8. Prilla M, Nolte A, Herrmann T, Kolfschoten G, Lukosch S (2013) Collaborative usage and development of models: state of the art, challenges and opportunities. Int J e-Collaboration. Special issue on “Collaborative usage and development of models.” 9:1–16
9. Rittgen P (2010) Collaborative modeling: roles, activities and team organization. Int J Inf Syst Model Des (IJISMD). 1:1–19
10. Gjersvik R, Krogstie J, Følstad A (2001) Participatory development of enterprise process models. In: Proceedings of EMMSAD’01, pp 195–215
11. Richardson GP, Andersen DF (1995) Teamwork in group model building. Syst Dyn Rev 11:113–137
12. Bostrom RP, Anson R, Clawson VK (1993) Group facilitation and group support systems. In: Jessup LM, Valacich JS (eds) Group support systems: New perspectives. Macmillan, pp 146–168
13. Rouwette EAJA, Vennix JAM, Mullekom T (2002) Group model building effectiveness: a review of assessment studies. Syst Dyn Rev 18:5–45
14. Herrmann T, Nolte A (2014) Combining collaborative modeling with collaborative creativity for process design. In: Rossitto C, Ciolfi L, Martin D, Conein B (eds) COOP 2014–proceedings of the 11th international conference on the design of cooperative systems, 27–30 May 2014, Nice (France). Springer International Publishing, pp 377–392
15. Herrmann T, Nolte A (2010) The integration of collaborative process modeling and electronic brainstorming in co-located meetings. In: CRIWG 2010, LNCS 6257. Springer, pp 145–160
16. Wiechers M, Nolte A, Ksoll M, Herrmann T (2014) Using mobile devices to overcome idle times in modelling workshops. Mensch and Computer 2014–Workshopband: 14. Fachübergreifende Konferenz für Interaktive und Kooperative Medien–Interaktiv unterwegs-Freiräume gestalten. 327
17. Seyed T, Burns C, Costa Sousa M, Maurer F, Tang A (2012) Eliciting usable gestures for multi-display environments. In: Proceedings of the 2012 ACM international conference on interactive tabletops and surfaces. ACM, pp. 41–50
18. Streitz NA, Gei\ssler J, Holmer T, Müller-Tomfelde C, Reischl W, Rexroth P, Seitz P, Steinmetz R (1999) i-LAND: an interactive landscape for creativity and innovation. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, New York, pp 120–127
19. Döweling S, Tahiri T, Schmidt B, Nolte A, Khalilbeigi M (2013) Collaborative business process modeling on interactive tabletops. In: Proceedings of the 21st European conference on information systems, pp. 1–7
20. Kolb J, Rudner B, Reichert M (2013) Gesture-based process modeling using multi-touch devices. Int J Inf Syst Model Des 4:48–69
21. Oppl S, Stary C (2014) Facilitating shared understanding of work situations using a tangible tabletop interface. Behav Inf Technol 33:619–635
22. Herrmann T, Nolte A, Prilla M (2013) Awareness support for combining individual and collaborative process design in co-located meetings. Comput Support Coop Work (CSCW) 22:241–270
23. Wiechers M, Christmann N, Nolte A, Herrmann T (2015) Taking pictures from a large wall display—Why would you do that? In: Workshop on collaboration meets interactive surfaces (CMIS): walls, tabletops, mobiles, and wearables
24. OMG: Business process modeling notation (BPMN) specification. object management group (2006)
25. Scheer A-W (1998) ARIS—Modellierungsmethoden, Metamodelle. Springer, Anwendungen
26. OMG: UML 2.0 superstructure—final adopted specification. object management group (2003)
27. Ackermann F (1996) Participants’ perceptions on the role of facilitators using group decision support systems. Group Decis Negot 5:93–112

28. den Hengst M (2005) Collaborative modeling of processes: what facilitation support does a group need? In: AMCIS 2005 Proceedings, pp 73–80
29. Renger M, Kolfschoten GL, De Vreede GJ (2008) Challenges in collaborative modelling: a literature review and research agenda. Int J Simul Process Model 4:248–263
30. Andersen DF, Richardson GP (1997) Scripts for group model building. Syst Dyn Rev 13:107–129
31. Briggs R, de Vreede G-J (2009) ThinkLets: building blocks for concerted collaboration. center for collaboration science
32. Vennix JAM, Andersen DF, Richardson GP, Rohrbaugh J (1992) Model-building for group decision support: issues and alternatives in knowledge elicitation. Eur J Oper Res 59:28–41
33. Lee Y (2006) Design participation tactics: redefining user participation design. In: Design research society international conference
34. Luebbe A, Weske M (2011) Tangible media in process modeling–a controlled experiment. In: Advanced information systems engineering. Springer, pp 283–298
35. Klein B, Burkhard RA, Meixner C, Treyer L (2015) Dynamic multi-view, multi-format, multi-user visualizations: for future cities. In: 2015 19th international conference on information visualisation (iV). IEEE, pp 360–365
36. Vessey I, Galletta D (1991) Cognitive fit: An empirical study of information acquisition. Inf Syst Res 2:63–84
37. Moody DL (2009) The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering. Soft Eng IEEE Trans 35:756–779
38. Sheppard SR (2005) Landscape visualisation and climate change: the potential for influencing perceptions and behaviour. Environ Sci Policy 8:637–654
39. Brown RA, Eichhorn D, Herter J (2012) Virtual world process perspective visualization
40. Harman J, Brown R, Johnson D, Rinderle-Ma S, Kannengiesser U Augmenting process elicitation with visual priming: an empirical exploration of user behaviour and modelling outcomes. Inf Syst. (forthcoming)
41. Matin E, Shao KC, Boff KR (1993) Saccadic overhead: information-processing time with and without saccades. Percept Psychophys 53:372–380
42. Bowman DA, McMahan RP (2007) Virtual reality: how much immersion is enough? Computer 40:36–43
43. Anslow C, Marshall S, Noble J, Tempero E, Biddle R (2010) User evaluation of polymetric views using a large visualization wall. In: Proceedings of the 5th international symposium on software visualization. ACM, pp 25–34
44. Prilla M, Nolte A (2010) Fostering self-direction in participatory process design. In: Proceedings of the eleventh conference on participatory design 2010. ACM, pp 227–230
45. Dourish P, Bellotti V (1992) Awareness and coordination in shared workspaces. In: Proceedings of the 1992 ACM conference on computer-supported cooperative work. ACM, pp 107–114
46. Chokshi A, Seyed T, Marinho Rodrigues F, Maurer F (2014) ePlan multi-surface: a multi-surface environment for emergency response planning exercises. In: Proceedings of the ninth ACM international conference on interactive tabletops and surfaces. ACM, pp 219–228
47. Shakeri Hossein Abad Z, Anslow C, Maurer F (2014) Multi surface interactions with geospatial data: a systematic review. In: Proceedings of the ninth ACM international conference on interactive tabletops and surfaces. ACM, pp 69–78
48. Anslow C, Marshall S, Noble J, Biddle R (2013) Sourcevis: collaborative software visualization for co-located environments. In: 2013 first IEEE working conference on software visualization (VISSOFT). IEEE, pp 1–10
49. Bragdon A, DeLine R, Hinckley K, Morris MR (2011) Code space: touch+air gesture hybrid interactions for supporting developer meetings. In: Proceedings of the ACM international conference on interactive tabletops and surfaces. ACM, pp 212–221
50. Seyed T, Costa Sousa M, Maurer F, Tang A (2013) SkyHunter: a multi-surface environment for supporting oil and gas exploration. In: Proceedings of the 2013 ACM international conference on interactive tabletops and surfaces. ACM, pp 15–22

51. Jetter H-C, Geyer F, Schwarz T, Reiterer H (2012) Blended interaction–toward a framework for the design of interactive spaces. In: Workshop DCIS. Citeseer
52. Wigdor D, Shen C, Forlines C, Balakrishnan R (2006) Table-centric interactive spaces for real-time collaboration. In: Proceedings of the working conference on advanced visual interfaces. ACM, pp 103–107
53. Nolte A, Brown R, Poppe E, Anslow C (2015) Towards collaborative modeling of business processes on large interactive touch display walls. In: Proceedings of the ACM international conference on interactive tabletops and surfaces, pp 379–384
54. Ellis CA, Gibbs SJ (1989) Concurrency control in groupware systems. In: SIGMOD '89: proceedings of the 1989 ACM SIGMOD international conference on management of data. ACM, New York, NY, USA, pp 399–407
55. Diehl M, Stroebe W (1987) Productivity loss in brainstorming groups: toward the solution of a riddle. J Pers Soc Psychol 53:497–509
56. Shaw ME (1981) Group dynamics: the psychology of small group behavior. McGraw-Hill College
57. Rittgen P (2013) Group consensus in business process modeling: a measure and its application. Int J e-Collaboration (IJeC) 9:17–31
58. Nielsen J (1994) Enhancing the explanatory power of usability heuristics. In: CHI '94: proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 152–158
59. Nolte A, Brown RA, Poppe E, Craig A (2015) Exploring collaborative modeling of business processes on large interactive touch display walls. In: Workshop on collaboration meets interactive surfaces (CMIS): walls, tabletops, mobiles, and wearables
60. Rittenbruch M, Sorensen A, Donovan J, Polson D, Docherty M, Jones J (2013) The cube: a very large-scale interactive engagement space. In: Proceedings of the 2013 ACM international conference on interactive tabletops and surfaces. ACM, New York, NY, USA, pp 1–10

Chapter 13
Interactive Digital Cardwalls for Agile Software Development

Martin Kropp, Judith M. Brown, Craig Anslow, Stevenson Gossage, Magdalena Mateescu and Robert Biddle

Abstract Agile software development is characterized by very intensive communication and collaboration among members of the software development team and external stakeholders. In this context, we look specifically at cardwalls, noting that despite the wide availability of digital cardwalls, most Agile teams still use physical cardwalls to support their collaborative events. This is true even though a physical cardwall hinders efficient distributed software development and causes extra effort to capture story artefacts into digital tools to meet traceability and persistence requirements. We conducted two empirical studies in industry to understand the use of existing digital Agile cardwalls and to find out the needs for an ideal digital Agile cardwall. The *first study* was with eight Agile teams of committed digital cardwall users. The study showed the reasons why some teams use projected digital cardwalls and their detailed experiences with them. The study showed that most digital cardwalls seem not be sufficient for the highly interactive and collaborative Agile workstyle. The *second study* was with eleven Agile companies. The study comprised of the development of *aWall*, a software prototype of a large interactive high-resolution multi-touch display that supports varied Agile meetings where cardwalls are used. The results of the study emerged with design considerations for digital Agile cardwalls from the evaluation of aWall in a user workshop. Both studies, which were

M. Kropp (✉) · M. Mateescu
University of Applied Sciences Northwestern Switzerland, Windisch, Switzerland
e-mail: martin.kropp@fhnw.ch

M. Mateescu
e-mail: magdalena.mateescu@fhnw.ch

J.M. Brown · S. Gossage · R. Biddle
School of Computer Science, Carleton University, Ottawa, ON, Canada
e-mail: judithm.brown@carleton.ca

S. Gossage
e-mail: stevenson.gossage@carleton.ca

R. Biddle
e-mail: robert.biddle@carleton.ca

C. Anslow
Department of Computer Science, Middlesex University, London, UK
e-mail: c.anslow@mdx.ac.uk

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_13

conducted concurrently, began with an interest in new large interactive surface technologies which might have the potential to provide not only the required interaction possibilities to support intensive collaboration, but also the required large display format necessary for a collaborative space. The results of the studies collectively seem to confirm our assumption, that large interactive surface technologies could bring the support for the collaboration of Agile teams to a new level, potentially making the teams more productive.

13.1 Introduction

As expressed in the original Agile Manifesto [1], Agile software development is a highly collaborative, communicative and interactive software development method. Transparency, openness and continuous feedback play an important role for the success of this development approach. As well as being a successful approach, Agile developers often show a very high identification with their Agile team and project [2, 3]. One of the core tools to support this approach are *cardwalls*. Cardwalls play a central role with respect to

- supporting collaboration among team members
- serving as an information radiator about the project state
- providing immediate feedback about state change
- providing the transparency about the project
- fostering the Agile team spirit

Despite the availability of many digital Scrum board tools, by far most Agile software development teams still use physical cardwalls for their daily stand-up meetings, as our own studies and others [4, 5] show. However, using physical cardwalls hinders efficient distributed software development and causes extra effort, since artefacts must be captured into external digital tools, to provide the often required traceability and persistent storage requirements.

We believe that large interactive surfaces have the potential to provide not only the required interaction possibilities but also the required large size for the type of collaborative workspace needed by Agile teams. With this type of cardwall, the whole team can meet in front of the wall and interact with it directly, potentially sharing results with remote team members instantaneously.

In this chapter we provide an overview of the usage of digital cardwalls in software development and our own research work in this area. The rest of this chapter is structured as follows. Section 13.2 provides an overview of work in the area of physical and digital cardwalls. We then present two independent studies, one conducted in North America and the other in Europe. Section 13.3 presents the results of a North American study of 64 digital cardwall users. The study resulted in a series of guidelines for developing collaborative digital cardwalls using large multi-touch displays. Section 13.4 presents a European prototype of a practical, large, digital cardwall that supports collaborative Agile practices, called aWall and a user study. Section 13.5

discusses the two studies in conjunction with a focus on multi-touch digital cardwalls to support collaboration in Agile teams. Section 13.6 concludes with a summary of this chapter.

13.2 Background

We now present background on cardwalls for Agile software development teams focusing on physical cardwalls, web-based cardwalls, story repositories, and digital cardwalls. We present case studies about cardwalls and tools that support these tasks.

13.2.1 Physical Cardwalls

The cardwall is a physical artefact that is typically used as a tool for planning and tracking the progress of an iteration during an Agile software development project. But how can the necessary level of detail and complexity be captured on a few cue cards pinned to a wall, and how can the cardwall help software development teams to meet their goals?

These questions were addressed in Sharp et al.'s five-year observational study of an XP software development team's use of physical storycards and cardwalls where she addressed the topics of both physical and social interactions with these artefacts [5]. The authors applied Green's Cognitive Dimensions framework [6] to understand the value of notations used in storycards and cardwalls and how they support (or fail to support) the various cognitive dimensions of Green's framework (See Table 13.1). The study also addressed how the social context of the XP team's process frames the underlying agreements about how these artefacts are used, and how they support the goals of producing working software.

Sharp et al. found that the storycard notation supported the following cognitive dimensions: abstraction, closeness of mapping, low diffuseness, provisionality, and low viscosity. User Stories capture requirements and are therefore an abstraction of them, which also means they are necessarily close to the domain which supports closeness of mapping. Low diffuseness is supported by the stories being written in the language of the user; they are brief and terse by design because they are only intended to be a reminder for further discussion. The storycard medium on which the story is presented (i.e., on an index card or sticky note) gives the storycard a feeling of provisionality. This medium also supports low viscosity because it encourages the engagement of a storycard by the users of the cardwall. However, the storycard does not have much support for the following dimensions, error proneness, progressive evaluation, premature commitment, hidden dependencies, and hard mental operations.

Table 13.1 Cognitive dimensions used by Sharp et al. [5]

Cognitive dimension	Definition
Abstraction	Can elements be encapsulated? If so, to what extent?
Closeness of mapping	How directly can the entities in the domain be expressed in the notation? Does the notation include entities that match the key concepts or components of the domain?
Consistency	When some of the language has been learned, how much of the rest can be inferred? Are similar features of structure and syntax used in the same way throughout?
Diffuseness	How many symbols or graphic entities are required to express a meaning?
Error-proneness	Does the design of the notation induce 'careless mistakes'?
Hard mental operations	Does the notation use mechanisms such as nesting and indirection that require mental unpacking or 'decoding'? For example, are there places where the user needs to resort to fingers or additional annotation to keep track of what's happening?
Hidden dependencies	Is every dependency overtly indicated in both directions? Is the indication perceptual or only symbolic?
Premature commitment	Do developers have to make decisions before they have the information they need?
Progressive evaluation	Can a partially-complete representation be executed or evaluated to obtain feedback on 'how am I doing'?
Provisionality	Can indecision or options be expressed?
Role-expressiveness	Can the reader see how each component relates to the whole, and what the relationships between notational elements are?
Secondary notation	Can developers use layout, colour and other cues to convey extra meaning, above and beyond the 'official' semantics of the language?
Viscosity	How much effort is required to perform a single change? How much effort is required to perform multiple changes of the same type? Does making one change then have the 'knock on' effect of requiring other changes?
Visibility	Is every part of the notation simultaneously visible—or is it at least possible to juxtapose any two parts side-by-side at will? If the notation is dispersed, is it at least possible to know in what order to read it?

The cardwall generally supported the following dimensions: provisionality, low viscosity and process visibility. It is easy to move cards, change labels and, start new iterations which all contribute to the cardwall's high provisionality and low viscosity. The cardwall's columns help reveal the underlying process and can be easily understood, which makes the visibility of the process high. There were also dimensions that were not directly supported by cardwalls: consistency, hidden dependencies, role expressiveness, progressive evaluation, and error-proneness.

The storycard or cardwall notation alone did not directly support all of the dimensions. Social agreements, discipline and interactions that framed the use of the storycards and cardwall, contributed to the support of the missing cognitive dimensions. Without the social interactions the benefits of the storycard and cardwall would be significantly reduced. Sharp et al. concluded by advising "Any Agile team looking to move towards digitising the team's support will need to take account of the complex relationships that exist within this social system if they wish to retain key properties of successful teams [5]."

For example, while a general template for stories usually exists such that key information is presented as follows: "As A < *Role* >", "I want < *Description* >" "so that < *Benefit* >", one still finds differing notations being used across Agile teams. However, within any one team the notation and use of cards is strictly adhered to. Everything on the card has meaning to the team including the location of the storycard, its colour, the size of the lettering and other meaning-laden annotations. A mature established team might have a well-defined notation while a new team might still be looking for what works best for them. To support these social behaviour storycards must be modifiable.

The use of the cardwall is also an extremely flexible procedure but has its generalities in that teams use walls for the duration of a project and leave them on constant display where they are easily seen—usually in a common space where anyone walking by could get an idea of the progress of the project. The cardwall is generally regarded as an 'Information Radiator' [7] and helps ensure the transparency of the project. Like the storycard, the cardwall is full of meaning not obvious to an observer who lacks familiarity with Agile methodologies or team specific notation. However, key information that one should be able to gather almost instantaneously from any cardwall is the general progress of the project. The placement of stories signifies whether or not they are in active development, waiting to be started or finished. Again the cardwall is generally without structure when considering its use among distinct teams, but it is used in an extremely consistent manner within any one team.

Using cardwalls requires an active participation between customers and developers and was originally described as 'The Planning Game' [8], where the objective is to prioritize the cards, sort them into releases and sprints, assign them to developers and have the developers accept or decline them until the cards were sufficiently sorted into at least the next sprint; unassigned stories are left in the project backlog, to be dealt with at the next iteration or release planning meeting.

The social interactions involved in the whole process enables teams to determine their best use of the storycard and cardwall's notation, which includes the organization and layout of the storycards on the cardwall. These interactions reveal the importance and meaning of the stories and thus drives their physical placement on the cardwall, which, in turn drives the story's progress through the system. The physical size of the cards is also of utmost importance since the size limits the information it can hold, and therefore encourages a communicative dialog.

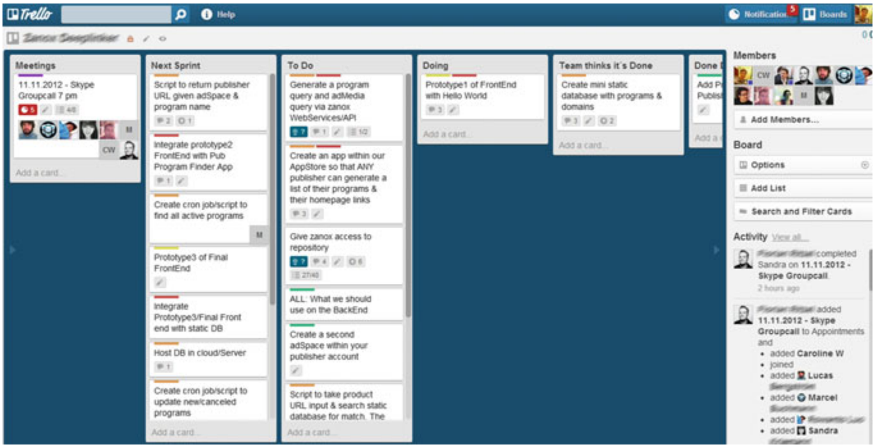


Fig. 13.1 Trello—web-based cardwall for a software development project

13.2.2 *Web-Based Cardwalls and Story Repositories*

Some tools have explored adapting physical cardwalls and story repositories to the web, notably Trello a web-based card wall and Jira a web-based story repository.

Trello [9] was created by Fog Creek Software. Perhaps more than any other digital tool for story management, Trello captures the simplicity of the traditional physical cardwall. The simple design allows flexibility in how it can be used. Trello can align with many different workflows, from simple to-do lists to Agile development as seen in Fig. 13.1, and also to other personal, business or management applications.

Each new board starts with three empty lists titled: Todo, Doing and Done, but you can add as many lists as you want, each with its own title. Similar to the storycard, in Trello you add new content to a list by clicking the “add a card” button at the bottom of any list. The cards have two views, a minimal view used while viewing the board at large and a detailed view, where you can see and edit all the extra content that is hidden on the back of the card. Each list can grow arbitrarily and the cards allow the user to easily add rich content, such as images and URLs or even embedded videos. With Trello, everything is saved automatically so there is no need to remember to save or update. On the Trello homepage, they explain that the simplicity of their applications allows users to use it in a flexible manner that can reflect the way they think or the processes they follow.

Trello has been designed very well from a user interaction perspective, but it does have limitations. For example, it is not possible to view the details of a storycard while still viewing the cardwall. Similarly, one can only see the details of one particular story which makes it difficult when planning and the discussion involves the details of more than one story. Finally, Trello is not designed to support simultaneous, co-located multi-user interaction which may have an impact on its support for collaboration.

JIRA [10] is an issue tracker tool developed by Atlassian. JIRA allows software development teams to track and assign issues as well as help to track the activity of teams and their members. Issue Trackers, also known as bug trackers, ticket support systems, or management workflow systems, allow teams to enter and track the progress of whatever the particular system is designed to track. They provide useful search features and reporting capabilities, including graphs to help visualize the progress, and reports for management. Such systems are not really designed to support Agile planning with User Stories, but are widely used for this purpose in practice.

JIRA is one of the most popular issue tracker tools and has been adopted and repurposed by Agile software development teams to manage their User Stories. The major impetus for this development was that Agile teams were looking for a solution to the distributed team dilemma where team members working from different locations had no access to the team cardwall. The use of JIRA by Agile teams for this purpose became so popular that an add-on was developed to add a cardwall view on top of JIRA as seen in Fig. 13.2; this add-on is available under the name JIRA AGILE. Several of the teams involved in our field study, which will be described in Sect. 13.3 used JIRA in combination with the Greenhopper plugin (the predecessor of JIRA AGILE).

13.2.3 *Digital Cardwalls*

Large high-resolution displays are now readily available, as is the support for multitouch capabilities. Leveraging these technologies seems like an obvious place to start

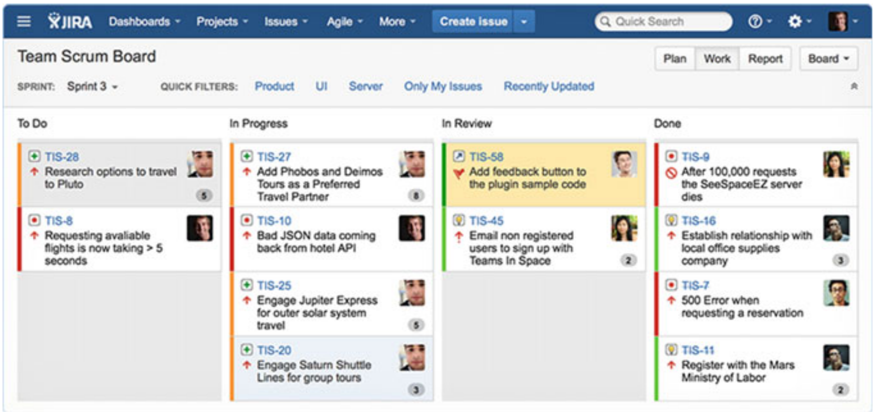


Fig. 13.2 JIRA—typical cardwall used by the teams in our study using the JIRA Agile type of cardwall. Each card is associated with an issue in JIRA. Such cardwalls were projected on walls in team meetings

when thinking about developing a digital cardwall. Every day more devices are being produced at a reasonable cost with support for two or more simultaneous touches; a critical feature for the development of truly collaborative tools.

The Agile Planner for Digital Tabletop (APDT) [11] was designed based on a prototype by Weber et al. [12] which was intended for co-located collaboration on a single touch surface. APDT chose to use this as a starting point, but wanted to enhance it with support for multi-touch, the ability to interface their cardwall with other Agile planning tools, and real-world evaluation based on user studies. It was designed after observing traditional Agile planning meetings, as well as meetings conducted using the Distributed Agile Planner (DAP) [13]. As the name suggests DAP was designed to support distributed Agile teams in the planning and maintenance of an Agile project through the use of a digital whiteboard and storycards. DAP had been developed with a traditional single user interface paradigm (one keyboard, one mouse), to enable users to collaborate remotely; it did not support multi-user interactions in a co-located environment. APDT also studied and drew from the literature available on the use of multi-touch tabletops for group collaboration. APDT was developed as a multi-touch enabled tool, specifically for two tables designed by Smart Technologies Ltd. using Smart's proprietary SDK. The first table used DViT (Digital Vision Touch) [14] technology and had support for two concurrent touches. The second table used FTIR (Frustrated Total Internal Reflection) [15] technology and had support for 40 concurrent touches. The two touch capabilities of the DViT table limited the users' ability to work concurrently, while the small form factor of the FTIR table meant that it was difficult to leverage its support for a much greater number (e.g., 40) simultaneous touches.

Our previous work includes Collaborative Multi-touch Agile Planner (CMAP) [16]. APDT (as described above) was a highly functional, full featured tool, however, it only supported two simultaneous touches; a limitation that influenced our design. With CMAP hardware and operating system independence was a goal. We also wanted to support multiple concurrent touches. CMAP extends previous work on Agile tool support by addressing key issues outlined by Scott et al. about group collaboration around digital tabletops [17]. We also attempted to address issues found by Sharp et al. about how the physical nature of storycards and cardwalls affect their use and have a "reflexive relationship" with the social interactions in which the use of the cardwall is grounded [5]. Sharp and her colleagues stress that software designed to digitise the storycard and cardwall must carefully consider both the notational and social aspects of these artefacts.

The basic goals for the development of CMAP was to design a digital cardwall (a horizontal tabletop) with support for multiple concurrent inputs (to support multiple users), and was not limited to any specific hardware. We wanted to explore the use of touch gestures and how they could be used efficiently to manipulate the stories. At the time, we were working with horizontal surfaces, so we needed to understand the implications of user orientation and their effects on the usability of the tool. We also wanted to create a distributed system with a back-end that could be accessed simultaneously from multiple sites.

To support these goals several design decisions were made. The use of PyMT [18] and Python meant built-in platform independence, multiple concurrent user interaction and support for multiple inputs. PyMT allowed the use of gestures and rotatable widgets to deal with orientation issues. The need to support persistent and shared data was achieved by using a combination of open source projects including Git, GitHub and the Petaapan Google Application Server. Git took care of local and remote storage while the Google App Server took care of change notification through a publish/subscribe mechanism and notification hook between it and the GitHub repository.

CMAP took a structured approach to the storycard and cardwall. This was an attempt to capture all the relevant data that seemed to be critical to the Agile planning process. XML was used to define the data of the storycards and other artefacts and the view was a form-based widget with labels and text fields. Users could use traditional data entry via real or virtual keyboards. In an attempt to limit the amount of information contained by an artefact, a default minimal view was created, which only allowed the entry of a name for the artefact and a description. A second view was created for stories to allow the user to enter more information. This view was built so that the developer could, in conversation with the customer, elaborate on the information provided by the minimal view.

Some other researchers have looked at multi-touch interactive surfaces for supporting Agile teams. dBoard [19] provided an Agile cardwall on a single vertical touch display to support distributed teams with video conferencing capabilities. One screen was located locally while another remotely. However, the tool only utilized one screen at each location and did not support all the types of Agile team meetings. SourceVis [20] focused on code reviews for pairs of developers around large tabletops. SourceVis supports a suite of visualizations to explore code artefacts. However, the tool did not support many other types of Agile team meetings and was not connected to any issue tracking or source code repositories. CodeSpace [21] provided an environment based on CodeBubbles for exploring code that used a vertical touch screen, 3D gestures with Kinect, tablets, and laptops for collaboration. The aim was to provide support for software teams to explore code in a team setting. However, the tool did not focus on any Agile processes per se.

We now present a case study on the use of digital Agile cardwalls in practice, followed by a novel digital cardwall prototype and a user study.

13.3 Case Study: Cardwall Usage

As a result of a review of existing online tools, we observed there are many software tools to help manage Agile projects using user stories, but none of these tools are specifically designed for multi-touch technology or for large high-resolution displays. We envisioned a large high-resolution vertical cardwall that would use these technologies to support the known advantages of the large vertical physical cardwall (e.g. being an "information radiator" resulted from the physical cardwall's vertical nature), while allowing the introduction of new functionality only possible with a computerized system. Our research question was: How should one design a *digital* story cardwall which captures the known benefits of physical story cardwalls *and* provides additional functionality only possible with a software solution?

We wanted our design to be informed by the actual needs of real-world Agile teams. We therefore conducted a field study using observations and interviews to explore the priorities from the perspective of developers who currently use digital cardwalls. We designed a study to investigate real work environments of this type. We wanted to understand more of the reasons leading to the adoption of digital story cardwalls and the frustrations experienced by teams using digital cardwalls. Our study is described in detail in our recent paper [22]. Here we summarize our results and present the implications for the design of large multi-touch digital cardwalls that follow from the outcomes of our study.

We began by conducting a pilot study of a team of physical cardwall users to ensure we were aware of the behaviours and advantages of physical cardwall usage. We then designed a week-long expedition to observe and interview 8 professional, digital cardwall Scrum teams in four different organizations—two in Canada and two in the United States of America. Our observations would help us to *observe* cardwall behaviours and the interviews would help us to *explain* the behaviours we observed. The teams we studied used either JIRA/Greenhopper (now JIRA Agile) (See Fig. 13.2) or an in-house digital cardwall called StoryBoard as their digital cardwall. We observed 64 participants, using standard ethnographic methods [23–25], in iteration planning meetings (IPMs), daily standups, and retrospectives. We interviewed 8 team members for one hour each, using a semi-structured interview technique [24, 26–28].

Our data consisted of notes from 13 meetings, and 8 interview transcripts. We entered our data into a qualitative analysis program called Atlas.ti, and used grounded theory [29] combined with a thematic analysis technique [30] to analyse the data. Both these techniques use an inductive approach. One strength of this approach is that the researcher remains grounded in their data. The end result is that themes that are generated are a very good fit to the data collected. Our paper [22] describes how we found 15 saturated codes and reduced them to 7 themes. We next briefly summarize these 7 themes and present design implications for each one.

13.3.1 Cardwall Formats

The teams we observed were mostly using projectors to display cardwalls of external customer projects, although occasionally internal projects would be managed by a dedicated inhouse physical cardwall. Projected cardwalls have the advantage that they are portable and can be used in any available meeting space where there is a projector. The project lead was typically responsible for updating the cardwall as the team worked. This solved a few problems: the limited wall space, the need for customers who attended team meetings to only see their cardwall and not the cardwalls of other customers, and the need for remote team members to view the cardwall on their laptops and see it update in real time during team meetings.

Digital cardwalls allow teams to conduct meetings in shared locations like meeting rooms, which ensures privacy and confidentiality for customers. This means that teams working on sensitive projects can reap the benefits of a cardwall, while their customers can rest assured that their sensitive data is not compromised. This also has the benefit of reducing the requirement for wall space and allows meetings to be conducted in any office or boardroom (including the customer's site) using existing standard equipment.

A down-side to this practice is that teams lose the benefits incurred from the always-on and constant display of the physical cardwall as "information radiators." Therefore, teams should also consider using a dedicated display, located in a shared space close to the team. In this way, teams can benefit both from the flexibility of this 'display anywhere' solution, and from having the cardwall available in a central dedicated location.

With respect to interaction possibilities with projected cardwalls, we consistently saw the Scrum master interacting with the cardwall on behalf of the rest of the team, which kept the work tightly coordinated, but also slowed it down. The ability to interact directly with the cardwall through physical touch would be ideal and would support the principle of maintaining a level playing field among team members.

Digital cardwalls open up the possibility for combining cardwalls in multiple formats. We observed cardwalls displayed on large shared surfaces for meetings, but we saw the same cardwalls displayed on laptops or smaller desktop displays for standup meetings or for remote team members. Furthermore, although we did not see participants using smaller personal devices like smartphones or tablets, these formats may also be important and should be considered when designing. With a web-based cardwall, there could be any number of distinct form factors used for displays, ranging from smaller personal devices to larger shared displays. For displays with truly high resolutions like the 4K and 8K displays or tiled displays, we advise using general guidelines for designing on these types of high-resolution surfaces [31–36].

G1: Support diverse screen sizes with appropriate design to support the varied team practices around digital cardwalls and storycards.

13.3.2 Scaling the Design of the Cardwall

Our participants pointed out that there were problems with maintaining either physical or digital cardwalls, especially when projects were large. For digital cardwalls, the problem of page refreshing is an annoyance, but with physical cardwalls clutter is a challenge because of space limitations. Interestingly, the added benefits afforded by the digital environment were viewed as either a blessing or a deterrent. One of our project managers loved how you could link JIRA with other tools. However, a developer working on the same project recounted his frustrations with the number of email notifications he was receiving as a result of changes to the code base.

G2: Design for large Agile cardwalls that support displaying many cards at once. Consider the placement of cards for usability, their visibility, and the responsiveness of the user interface. Limit the number of notifications sent to developers, which can disrupt individual and collaborative workflows.

13.3.3 The Big Picture and Story Relationship Visualisations

Participants mostly described how digital stories helped developers to track their work. In the physical cardwall study we observed how the visualization of relationships between stories was supported through the use of simple methods like the use of colours, annotations, and swimlanes. In the digital cardwalls we observed support for some of these simple methods, but teams did not attempt to visualize sophisticated story relationships. From our interviews we learned that the ability to see these relations was either difficult to do, or was not supported by the cardwall being used. Participants wanted to see relationships between stories made more evident, but currently kept track of such dependencies 'in their heads'.

Large displays facilitate team planning work and are ideal for capturing a big picture view of the project. In our field study, all of the iteration planning meetings and retrospectives used projected images of the team's cardwall which was large enough for it to be easily seen at a distance by all participants. We also observed that during iteration planning meetings (IPMs), each story being considered as a candidate for the next iteration was viewed and discussed in detail. Fortunately, large displays have enough screen real-estate such that a group of co-located collaborators can easily view the details of one or more stories. From our observations, the short stand-up meetings were generally not held in meeting rooms and they did not generally have access to a large view of their cardwall. For these meetings, the big picture cardwall was used more as a point of reference similar to a cue card, and was ideal for keeping the meeting focused and on point.

G3: Always support a large format overview of the entire project to ground team and stakeholder discussions around a large display.

While the large display format is important, it is also important to support the scenario where a cardwall application is viewable on a variety of formats, including large displays. Our guidelines are based on Shneiderman's visual information seeking mantra: overview first, zoom and filter, then details on demand [37].

Overview first: Since one of our goals is to try and leverage the benefits of the physical cardwall, we make the assumption that whatever is on display on a large display should be discernable from a distance of approximately five to ten feet. An overview first view is best.

G4: Always begin with, and default to, a general overview display. If the display is large this view should be discernable at five to ten feet to ensure the display works as an "information radiator."

Zoom and pan: Since a digital cardwall can be viewed on virtually any display format, it can be difficult for the software to adapt to the actual size of the display and the resolution. Being able to re-size the current view of the cardwall should be possible. When zoomed in sufficiently, such that parts of the current view are no longer visible, panning allows the user to easily access the parts of the view that are hidden. When zooming in users should be made aware of the ability to pan and the directions in which they can pan to reveal the hidden content.

G5: Support zoom and pan to allow users to adjust dynamically to the demands of the social practice that they are engaging in.

Details on demand: We observed that the details of any particular artefact are not always important and therefore should not always be present. However, a digital cardwall needs a mechanism which allows users to show and hide details whenever they are available. To reduce user frustration, the user should see a visual cue indicating there are additional viewable details.

G6: Allow optional display of details that would otherwise clutter the current view to support detail-oriented conversations.

Designing for touch interaction: Many large display formats are now beginning to support touch interaction.

G7: Interactive components, whatever the display format, must always be displayed in a way that they are large enough to enable touch interactions easily. This is especially a concern for large high-resolution displays. This necessary prerequisite for interactivity removes a potentially frustrating roadblock to collaborative interactions.

13.3.4 Exploring and Filtering Information

Experienced Agile practitioners described how cardwalls help them keep track of the status and progress of multiple projects. More experienced participants envisioned how added functionality could help increase awareness about important, but currently invisible, aspects of their projects like how long it would take to implement a group of stories, or how many defects are associated with it. They wanted to group stories by software component so that the team could select different software components and see the associated stories, the progress of those stories and other aspects. This would tell them who would be impacted by a change.

One project manager wanted to see a feature that would help him remain aware of business goals because this could impact technical solutions. He also wanted to easily drill down from higher-level, more abstract ideas to lower-level, more detailed-oriented issues. He imagined the ability to visualize features on a cardwall would be ideal for meetings with high-level executives, but he also imagined wanting to drill down to the epics and stories which would help increase the overall awareness of the projects' real progress and provide a more realistic understanding of the work involved in the implementation of a feature amongst executives.

With traditional physical cardwalls, users cannot explore details and complex relationships since the whole idea is to present a big picture of a work in progress. The digital cardwall can be much richer and it can allow users to explore the project in several ways including viewing previous iterations, the backlog, and even other projects. When we further enhance the digital cardwall with the ability to filter, we open the door for more tailored explorations via a query language capability. In our field study, the participants expressed their frustration with regards to difficulty of searching, filtering, and tagging capabilities in their existing digital cardwalls. They also wanted to see dependencies and relationships between stories, to break down epics into smaller stories, and have the ability to trace a story back to an epic. Furthermore, the participants wanted to create sets that could be used as the data for a query. Based on this, the following guidelines have been identified.

Working sets and queries: Working sets identify a collection of stories of interest. A good default for a working set of stories is the contents of the current view or iteration. The point of the working set is to help users focus and quickly find information that may not be obvious or visible. Working sets can be outcomes of queries. In addition, users should be able to save and label a working set so that it can be re-visited. Ease of use is an important consideration in implementing this feature.

G8: Allow users to create sets of stories based on story details and further explore them via the composition of queries. This supports 'chunking' at an individual and social level, speeds interactions and eases demands on memory.

Query: The operations we need to query working sets are a simple query language based on predicate calculus, with basic support for the unary "NOT" operator, the binary "AND" and "OR" operators, the for-all and for-each operators as well as parenthesis for changing the order of precedence. Queries can be used to ask complex questions like "Find all the stories that have been completed, but for which there are reported bugs."

G9: Make simple queries easy, but also support a rich set of query operators. This functions enables important conversations about sets of stories.

Custom attributes and annotations: The ability to apply attributes to artefacts allows users to filter the working set by including or excluding artefacts that share a particular attribute. Filtering can be performed before or after a query and should be dynamic such that the addition or removal of categories should automatically refresh the results of the working set to reflect the change. It should be possible to use named attributes directly in user queries. The literature review of physical cardwalls found that teams often marked up their storycards with meaningful symbols, however, we did not see this behaviour in our study because this feature was not supported by the digital cardwalls being used. However, in our data there was evidence to support the use of custom attributes to mark different components, business goals, roles, people, and epics.

G10: Allow users to mark-up their cardwall and the stories within it. Optionally, let them assign meaning to their annotations. This supports teams to develop specific work practices relevant to their workplace.

13.3.5 Managing Backlogs

Backlogs were displayed congruently with a digital cardwall, and developers dragged stories from their backlog and dropped then onto a cardwall. The backlog is a significant element for large projects; the ability to see the backlog's stories, and to sort and prioritize the stories becomes increasingly important as the size of the project increases because larger projects have larger backlogs. Team members were very interested in the prioritization of the backlog items and how the stories were selected for each new iteration. The backlog was sorted by priority, but items came off the backlog into an "In Progress" list, which was "just a bucket." Developers could choose any item in the bucket. If a developer was done and no other developer wanted to pair program, that person would choose the next item in the backlog. Prioritization as a practice, also varied with how close the team was to releasing. The closer to releasing, the more the team prioritized.

We also saw how teams sometimes used different backlogs for infrastructure-type work that needed to be done, and another for customer stories.

Sometimes teams wanted to use the backlog to store information that would be useful for planning, estimating and reporting purposes such as bugs or vacations, because this improves visibility of these items.

G11: Backlog management is an important part of cardwall design. Sorting techniques within a backlog should support the concept of 'buckets' to support overall structuring of the backlog and also traditional sorting from most to least important within a bucket. Items in backlogs should be taggable. Allowing parts of backlogs to be sorted while others are not, mirrors the way backlogs are used in an Agile practice where just enough work is done to advance the team at the time.

13.3.6 Multi-disciplinary Use of Stories

We knew that stories were important to developers, but we learned that digital stories are vital to testers, designers and project managers, too. Cardwalls are becoming a multi-disciplinary team tool. However, we also noticed that the effectiveness of the information radiator aspect of a wall depended on the role of the person using it. For example, for developers and designers, the constant display of the current sprint is very useful for increasing the awareness of the status of the project, and it can even serve a motivational function. However, for the team's testers, their interest in the project is limited to the column of the wall that contains a list of stories that require testing, which tells them the size of their task.

Although the teams we observed were using digital story cardwalls, these tools were limited in their support for concurrent multi-users. Their cardwalls allowed distributed users to access and view the cardwall from their remote location however, there was no support for real-time concurrent use in terms of editing and viewing the details of more than one story simultaneously. This sometimes caused inconveniences, where the details of distinct stories of interest had to be remembered instead of simultaneously viewed.

G12: Allow simultaneous multi-user access to allow all team members equal access to the cardwall so work will flow.

A cardwall where the stories could be pulled from a variety of sources would help teams in a support role adopt the cardwall as a collaboration tool within their own teams. Teams from QA, testing and interaction design would potentially benefit from this ability. It is common for team members occupying roles of this nature to work simultaneously on different projects, and as of yet they do not have standardized methods for organizing work across projects. It could be useful to allow such groups to get a big picture of their work which could be organized into multiple projects using swimlanes.

G13: Support roles where team members' responsibilities extend beyond a single project. This will allow other groups like QA and UX who work on multiple projects at once, to also use cardwalls.

G14: Stories associated with various roles should be visually identifiable and the use of swimlanes is ideal for this purpose. Whether or not those who fulfill the role of QA or UX are seen as part of the team or outside the team, their work contributes value and the status of their work should also be visible.

Cardwalls can also be valuable tools at a personal level. In this situation, the story someone is working on could be broken up into tasks. Each task could be treated as a story, including estimates, and as tasks are completed the overall progress of that story on the team cardwall can be updated to show the progress.

G15: Design digital cardwalls for use also by a single user.

One important reason that cardwalls can be easily aligned with so many distinct processes is that teams can customize the columns to match their process. To seamlessly support virtually any team, the digital cardwall must allow teams to add, remove and rename columns. This way each column on their cardwall aligns with a distinct step in their process.

Another important customization option is the ability to customize and create new annotations, which would be re-usable and would allow teams to quickly identify artefacts that share the same annotations. This can be further enhanced by designing the system to differentiate between distinct annotations so that they can be used for querying, searching and filtering.

G16: Allow teams to customize the cardwall so that it aligns with their particular process.

13.3.7 Updating the Cardwall

Users complained that there were too many steps to accomplish simple tasks with their existing digital cardwalls. At the top of their list was saving and refreshing their cardwalls. Our participants felt that it was an annoyance to manually remember to save updates and to refresh their client to see changes made by others, which others may or may not have even saved.

Automatic save and refresh: Every action performed by users on the cardwall should automatically be saved. A cardwall for collaborative work designed as a distributed system with single database should automatically push local changes in real-time. A conflicting change should be sent back to the originating source (or sources) for manual resolution. Similarly, remote changes should automatically be sent to all clients.

G17: Support automatic synchronization of cardwalls; extra steps to save and refresh should be avoided. This supports workflow when some team members are remote.

To enable awareness, when a client receives an update, there needs to be a visual cue that draws the attention of users. In this way, the instances of change blindness will be reduced when teams are collaborating at-a-distance.

G18: Create awareness of remote changes to reduce errors and increase usability.

While this detailed observational study produced useful guidelines for digital cardwall construction, another research team in Switzerland used information from surveys and their own experiences with digital cardwalls to build a large interactive digital cardwall called aWall. This is the topic of the next section of this chapter.

13.4 Case Study: aWall—a Large Digital Cardwall

The previous sections have shown the limitations of current physical Agile cardwalls, especially with respect to their ability to support the highly interactive Agile collaborative workstyle. These limitations include the inability of physical cardwalls to be flexible, their lack of transparency where missing information is concerned, their ease of use, the inadaequate haptic experience, and the mismatch between required and available space. Physical cardwalls, however, remain an essential tool for team

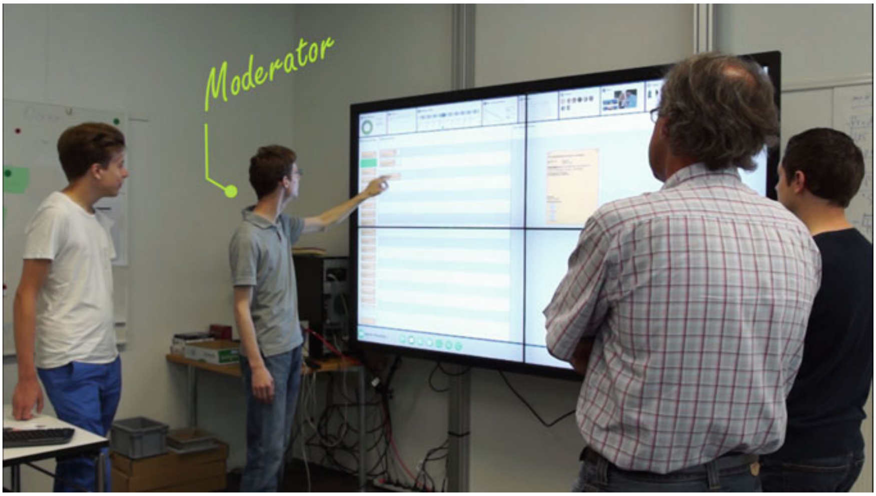


Fig. 13.3 aWall—digital Agile cardwall displayed on a large high resolution multi-touch wall (2×2 46 Inch 4 K displays) for Agile planning and team meetings

collaboration in software development, and are still used in most Agile teams [4, 22]. These findings are confirmed by our own study, which shows that 10 out of 11 teams still use physical cardwalls, typically in combination with digital tools [38]. Despite their prevalence, physical cardwalls still have issues as content is not digitalized and not integrated with other software development tools such as issue tracking and project management systems, and are not suited for distributed team collaboration.

In an endeavour to overcome these limitations and issues with physical cardwalls, we developed a research prototype *aWall*, which is a large high-resolution multi-touch digital Agile cardwall (see Fig. 13.3). The goal of aWall is to provide a digital collaborative workspace which supports Agile teams in their Agile workstyle. aWall supports elements of the natural aspects of physical cardwalls, and integrates with existing issue tracking and project management systems. aWall aims to support co-located, as well as distributed teams in their collaboration, allowing them to conduct typical Agile team meetings with the cardwall. aWall is described in more detail in [39].

13.4.1 Field Study

To deepen our understanding of how Agile teams collaborate and use cardwalls in practice, and what the requirements for an "ideal" digital cardwall would be, we conducted a field study with 44 participants from 11 companies in Switzerland [38]. The study consisted of semi-structured interviews: 10 group interviews and 3 individual interviews. The group interviews lasted 2 hours and the individual interviews lasted 1 hour. All interviews were audio recorded and transcribed, for later analysis.

When asked about the requirements for a digital Agile cardwall, the interviewees stressed the importance of non-functional requirements, like the need for a large display size, configurable views, instant availability of information, overview of information, always-on information, easy reachability of context dependent information, easy readability of information, simultaneous multi-user touch interaction, direct interaction with data, and "navigation-less" operation.

Our hypothesis, that existing digital tools do not adequately support the communication and collaborative aspects for Agile team meetings effectively, was confirmed through our study, which also revealed that mostly only one team member (often the Scrum Master) fed information into project management tools.

Based on our study results, we developed *aWall* to support Agile teams (co-located or distributed) more effectively than existing physical and digital tools. aWall is designed to support Agile team meetings like daily stand ups, sprint plannings, and retrospectives by providing information dashboards, maintaining user stories and tasks, supporting the customization of Agile processes, and integrating with external issue tracking systems, like JIRA [10]. aWall was developed by an interdisciplinary project team of computer scientists and psychologists (from the School of Engineering, and the School of Applied Psychology, from the UAS Northwestern Switzerland).

13.4.2 Design Considerations

Based on the requirements elicited during the interviews of our field study, we identified the following core design considerations for digital Agile cardwalls which helped guide the design of aWall.

Physical Size. Digital cardwalls need to satisfy not only the requirements for interacting with digital content, but also provide enough physical space to display information to effectively support team collaboration. Therefore, the size of a digital cardwall needs to be at least comparable to that of physical cardwalls. Thus aWall consists of four 46 inch displays (2×2), for a wall size of 2.05 m width and 1.25 m height (see Fig. 13.3).

High Resolution Display. Digital cardwalls should have a high resolution display to provide enough real estate to display large amounts of information at once while still ensuring the readability of text elements, widgets, and views. Each display in aWall is 3840×2160 pixels, for a total resolution of 15360×8640 pixels.

Multi-user and Multi-Touch. Digital cardwalls should support multi-touch capabilities to allow multiple users to work simultaneously with artefacts and provide an accurate and effective touch experience. aWall consists of a 12 point multi-touch infrared optical overlay (PQ Labs frame[1]) which is attached to the display wall.

[1] http://multitouch.com/.

Integration with Issue Tracking Systems. Digital cardwalls should be integrated with issue tracking systems as they are fundamental for the work flow process within software development teams. aWall is designed to run on top of existing third party issue tracking systems such as JIRA. Therefore, infrastructure functionality can be reused and already customized Agile processes can be utilized.

Availability of information and transparency. Digital cardwalls should be installed in a team's open office area, always being switched on, and have a permanent view of the task board. Therefore digital cardwalls can replace physical cardwalls and act as the team's external memory of the project, thus provide the desired transparency in Agile teams. aWall is designed to be large and portable, which allows large amounts of information to be displayed and can be moved to different locations within an open office area.

Ubiquitous and Deployable. Digital cardwalls should support co-located and distributed teams by being ubiquitous and easily deployable applications. aWall was developed as a web application based on HTML5, JavaScript, and interact.js for multi-touch support.[2] Though we especially focused on the support of co-located Agile team work, aWall could be extended with specific concepts for distributed team work like those developed by Esbensen et al. [19].

Foster communication and collaboration. Digital cardwalls should foster communication and collaboration, serve as a teams external memory, and provide all context specific needed information. aWall implements the concepts of *information widgets* which provide additional information on the display to the teams core activities. For example this information includes: Definition-of-Done Lists, Burndown charts, Pair-Programming planner, team viewer, and meeting timer.

13.4.3 User Interface

The aWall user interface contains a number of different views, widgets, and interaction techniques designed to support the different types of Agile team meetings.

Action and information view. One of the findings of the study is that most cardwall interactions take place during Agile meetings. Each meeting, however, has its own specific goals, operates on different data, and requires various supporting tools and information. To support these different types of information needs, the aWall display is divided into an *action view* and an *information view*. Figure 13.4 shows the view for a daily standup meeting highlighting the separation into information view on top (red bordered) and action view in the center (blue bordered). The action view is the main working area, which is dedicated to the core artefacts of a specific meeting. The main interactions during a meeting are performed by users on the action view. It is designed to make changes on the core artefacts as easy as possible, and with

[2] http://interactjs.io.

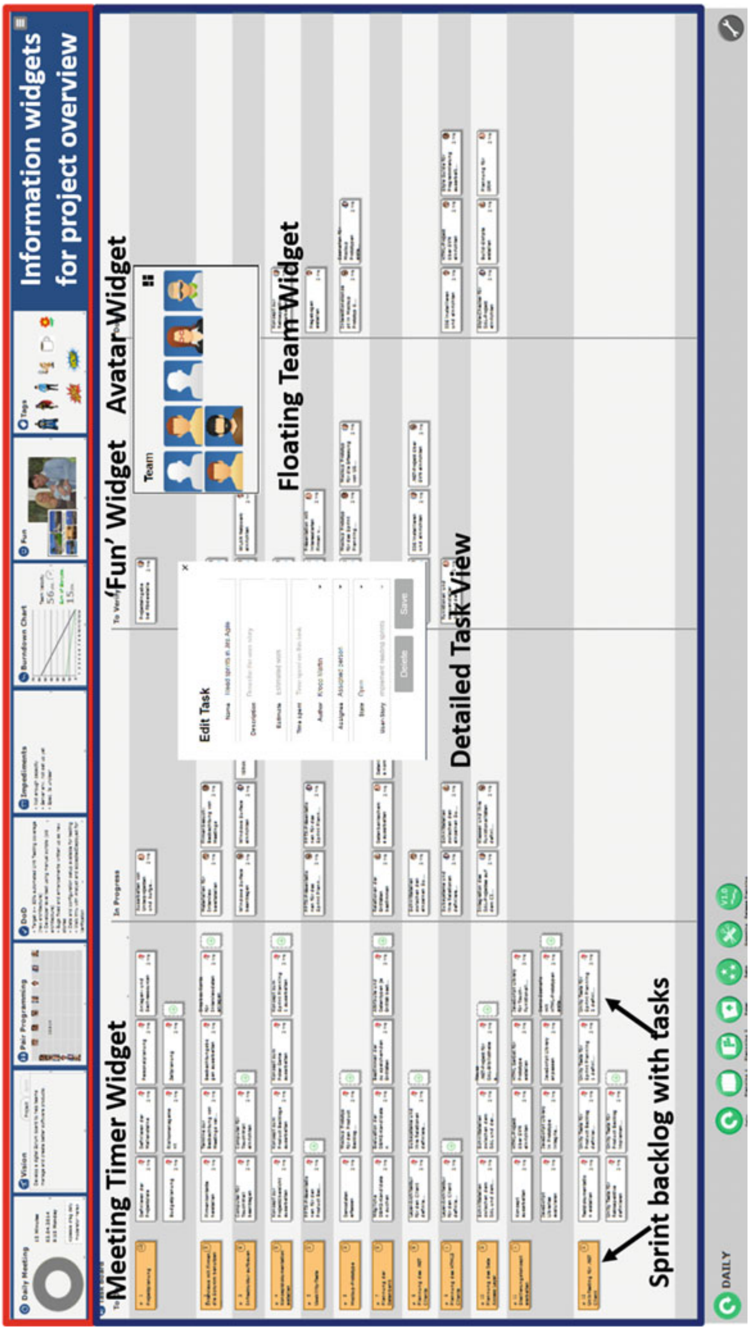


Fig. 13.4 Daily standup meeting screen with information view (*top* section with *red* border) and action view (*middle* section with *blue* border) and meeting navigation at *bottom*

this foster interaction and collaboration in the team (for example, moving cards by simple drag-and-drop gesture). The information view provides supporting information and tools needed for the meeting. It represents the *dynamic memory* of the team. As any dynamic system, the information view allows for change, and is specifically customized for the different meeting types. For example, the information view for the daily standup meeting contains specific information, like a timer widget showing the name of the meeting moderator and a countdown, a team widget showing the names of team members, a definition-of-done (DoD) widget, an impediment list widget, and a burndown chart for an iteration. All widgets can be switched on and off on the information view as needed.

Dedicated views. To provide the information needed for the different activities of Agile teams, aWall provides dedicated views for each Agile meeting. These views are tailored to the specific needs of this meeting. Figure 13.4 shows the classical task board view for the daily standup meeting. Figure 13.5 shows the screen for the sprint planning 1 meeting. The action view is divided into three columns. The left column shows the top priority user stories of the product backlog. The centre column shows the user stories that have been selected thus far for the next iteration. The right column shows a detailed view of the currently selected user story. The information view again shows supporting information that might be needed during the sprint planning meeting. So this view is optimized for the discussion and clarification of open issues during the sprint planning meeting with the development team. Relevant documents can be easily attached and opened on the wall. As another example, Fig. 13.6 shows the retrospective meeting view after team members have sent feedback on the iteration; their notes have been ordered on the right side. Users can navigate between the different meeting views using the navigation bar at the bottom of the screen.

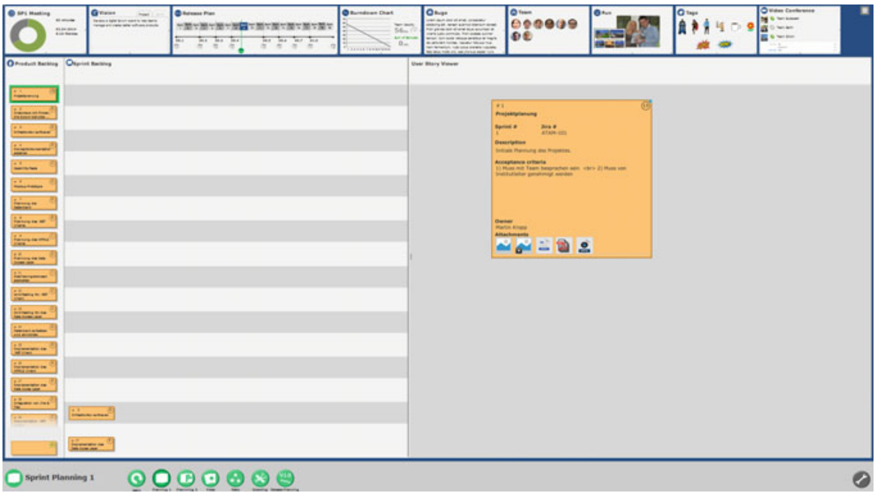


Fig. 13.5 Sprint planning meeting with a user story detail view

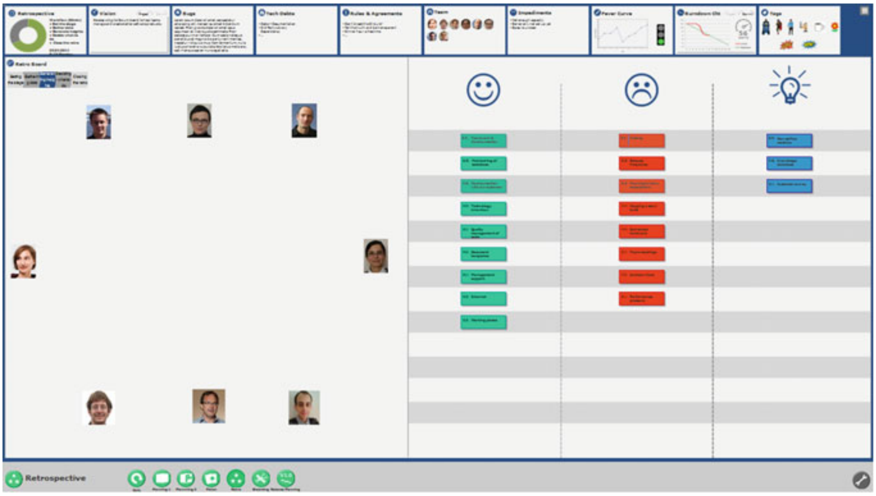


Fig. 13.6 Retrospective meeting view

Information Widgets The information view consists of a set of widgets (e.g. team widget, timer widget, burndown chart, definition-of-done widget, fun widget, avatar widget—see Figs. 13.4, 13.5 and 13.6) and can be independently configured for each Agile meeting. Each widget is designed to support distinct aspects of the collaborative Agile process. The team widget shows avatars and the names of the team members, and can be used to assign people to tasks during a daily standup meeting, for example. The timer widget supports time boxing during the meeting and can be used by the team when choosing a meeting moderator. The moderators' names are stored in the application and future moderators can be suggested based on previous selections. The fun widget allows users to post personal or fun images to the information view to help to evoke emotion from cardwall users and foster team thinking. The avatar widget can be used to drag avatars to any position on the wall or it can be attached to tasks or user stories. Both the fun and avatar widgets are designed to help with the interpersonal process in Agile teams (emotion management, team spirit). All widgets can be detached from the information view and moved around the cardwall to facilitate user interaction and ease of access (see Fig. 13.4).

Availability of Information. Any information needed for a meeting is visible and easily accessible; either on the action view or on the information view. If the team needs different supporting information, additional widgets can be switched on or off in the configuration button on the right side of the information view.

Interaction. aWall supports multi-touch and multi-user interaction. Fluid interaction with widgets and cards is enabled by gestures like tap, double tap, drag-and-drop, and pinch-to-zoom supporting changing task and user storycard position, moving widgets around on the cardwall, and changing the size of a widget. Data can be entered either on the cardwall with a virtual or physical keyboard or via the underlying issue tracker system, or mobile devices such as tablets.

Scalability of Information. The results of the field study identified that teams usually do not need to see all details at any time, since they can often remember the card content just the title, or even by its position on the board. So by default, user story cards and task cards show only minimal information in large font size (e.g. title, priority and story points). By increasing the card size with a pinch-and-zoom gesture more information is displayed. The text size increases concomitantly with the widening of the card so that information can be more easily read depending on the distance from the cardwall. When all information is shown, the widget automatically switches into edit mode, so that data can be added or modified.

13.4.4 User Study

To evaluate the design of aWall we conducted a user study with professional Agile practitioners. The main purpose of the study was to evaluate the usability of functionality in aWall, the support of the Agile workstyle, and the applicability to real life situations in Agile teams performing the daily standup and sprint planning meetings. The user study was conducted with the aWall prototype where participants had to complete various tasks with the aWall working in groups. In particular the study aimed to address the following research questions:

- How easy is it to find and manipulate information?
- Is all the necessary information available and transparent?
- Does the platform stimulate discussion and communication?
- Can the appropriate tasks be fulfilled efficiently?

13.4.4.1 Participants

We recruited 11 employees (9 men and 2 women) from the same companies that participated in our interview study [38]. Most participants had extended experience in IT (mean 11.5 years), and several of them in Agile development (mean experience 2.8 years). They worked previously in different fields and covered a wide spectrum of Agile team roles (four Scrum Masters, two Agile coaches, two senior developers, one Agile grandmaster, one UX consultant and one head of a software development department). The companies also operated in different domains (two insurance domain, one manufacturing, two service providers, one engineering, and one from an enterprise software development company). Four companies sent two employees, and three companies sent one employee each. All companies had been applying Agile processes for at least one year.

13.4.4.2 Procedure

Prior to the user study, the participants received a presentation of the interview study results, but did not receive any information about the aWall application. We divided the 11 participants randomly into two groups. Both groups completed the same tasks with the aWall. Upon signing an informed consent statement, the participants were asked to act as a team during the workshop. Each participant received three tasks to be solved together in groups using aWall. The tasks involved a daily standup meeting and a sprint planning meeting. After receiving the task, each participant read the task out aloud to the other participants and completed it with their help.

The *daily standup tasks* were to start the daily standup meeting, choose a moderator for the meeting, and update the task board during the meeting by moving tasks to the appropriate columns. The tasks were formulated as follows:

- In this team you play the role of team member Dario. aWall shows already the daily standup view. Please find the functionality to start a daily standup meeting. The application suggests a moderator. Please ask the team member suggested by the application to play the moderator. The team member is willing to take this role and starts the meeting with the meeting timer. Please act as a team according to the received instructions.
- In this team you play the role of team member Roger. You report that the task "Implement Login Dialog" has been completed. Undertake the appropriate action on the board to visualize the new state. Now you want to start the task "Write acceptance tests for login". Again, perform the appropriate action on the board to visualize the new state.
- Switch off the Fun-Widget and switch on the Impediment list widget.

The *sprint planning tasks* were to show and discuss a user story during the meeting and move the selected user story to the sprint backlog. They were formulated in the same way as the daily standup tasks.

After completing the tasks for each type of meeting the participants discussed the benefits and deficiencies of aWall for that type of meeting with their two moderators. Both workshops were conducted by two moderators and lasted one hour each, and the discussions and results were recorded.

13.4.4.3 Findings

The overall feedback for aWall was very positive, with the participants considered the prototype to be effectively usable, capable of supporting Agile processes in general, and especially capable of supporting the collaborative workstyle of Agile teams. We now present the main findings with respect to our research questions.

Size aspects. The participants especially valued the large size and high resolution of aWall. The large size supports real team collaboration capabilities, similar to physical cardwalls. Displaying large amounts of information at once was deemed positive. As one participant stated[3]:

> With the large size you can display many user stories and tasks.

Readability of information. Most participants considered the displayed information to be legible, especially since the card titles are relatively large. Some participants considered the actual cards to be too small. Therefore, it is very important to be able to display the whole content of a card and enlarge the font size so that the whole team can read it from a distance (3–5 feet). One participant stated:

> That's really a nice feature, that cards can be enlarged and font size increases to improve readability.

Availability of information. The participants especially valued the availability of additional information and functionality for the different meetings. The separation of the display into action view and information view was easily understood and valued. Some participants mentioned that elements placed on the upper side of the display wall might be out of reach for smaller people. Another participant liked the extra features:

> I like the extra features around the main view and the additional information.

Discoverability of functionality. The participants discovered most of the functionality of aWall by themselves and could easily interact with the application. There were some issues with discoverability of those functions that were not a straightforward transfer of physical cardwalls into the digital world. For example, the timer widget has no corresponding artefact in the practice of Agile teams. Whereas, direct implementations of the pin-board's functionality (e.g. the task-board shown in the daily standup meeting) were instantly understood and deemed as valuable by the participants. That was also the case for the widgets inspired from Agile practices such as the team widget which is based on the observation that Agile teams sometimes write the team members' names on the cards or even hang their pictures on the pin-boards.

Third-Party system integration. The integration with third-party tools was positively rated. Tasks modified during the daily standup meeting are immediately synchronized in the Agile project management tool (JIRA). There is no extra effort to update the tasks manually, as is typically required after a meeting using a physical cardwall. One participant stated:

> The link to JIRA with automatic update of data is important.

Flexibility and customization. Increased flexibility with respect to both the manner of conducting the meetings and the display of information was considered important by the participants. For example, the timer widget solicited choosing a moderator

[3] All quotes have been translated from German.

at the beginning of a meeting. The flexibility provided by aWall was also positively rated, especially with respect to conducting retrospective meetings that sometimes might prove strenuous. The participants considered that it is important to create a proper environment especially for this type of meeting as sometimes they tend to transmute into a drill. Most participants were in favour of a greater flexibility of the time boxing capabilities, preferring that it only optionally choose a moderator and not show the elapsed time, but the time of day during the daily meeting. The participants valued the team widget, but wanted to have more information displayed (e.g. absences, vacation days and the like) and wanted more ability to customize. Furthermore, the participants remarked that they should be able to add functionality to aWall on their own and not be dependent on standard functionality as is often the case with other Agile tools.

Agile collaborative workspace. Offering tags and avatars as well as the fun view was positively seen as bringing positive emotion into collaborations. One participant mentioned the positive effect of avoiding media disruption, because his team was able to do all of their interactions with only one medium:

> With such a board we could probably avoid media discontinuity.

Filtering and representation of information. The participants especially requested to have filter functions, to highlight and show the desired information. As an example, participants wanted to highlight all tasks of a team member when touching that person in the team view. Participants suggested using different colours for different types of user stories to increase readability (e.g. to distinguish between technical tasks, bug reports, or user requirements).

Task time recording. Some participants suggested automatically capturing the time spent on a task combined with computing the work hours on the task, which they felt would help provide further metric details of performance.

Provenance of information. Some participants suggested having automatic recordings of meetings with voice recognition and transcriptions of the discussions and the interactions in front of the display wall for later recollection and analysis of the meetings.

The user study with aWall has shown that large multi-touch wall technology combined with appropriate interaction and visualization features have the potential to provide a collaborative workspace, information transparency, direct interaction with information, as well as serve as an information radiator. We found that users especially valued the large size of the wall due to the physical space affordances, the dedicated views with context specific information, and the always visible and direct information access. A combination of these technologies and features has the potential to replace physical cardwalls in the future. We now discuss the previous cardwall usage study and guidelines with the aWall prototype.

13.5 Discussion

Cardwalls have been part of Agile software development since the beginning, and have roots in displays of index cards common in the pre-history of Agile methods, such as CRC cards [40].

Since the earliest days, there has been praise for characteristics of physical paper cards on an actual wall, emphasizing flexibility of cards and layout, scale of large walls with wide visibility and team access, as well as immediacy and speed of interaction. Studies of physical cardwalls confirmed these advantages [4, 5] and exposed the rich benefits of team use of cardwalls. In contrast, there were early cautions about digital cardwalls, with warnings about the detriments of small screens, rigid formats, and interaction limited to a single person.

As time has moved on, however, the limits of digital cardwalls have been reduced, with very large screens now commonplace, and touch surfaces allowing more than one person to interact. At the same time, use of workflow tracking systems has come to be regarded as essential, and distributed teams ubiquitous; both of these trends would be better supported by digital, rather than physical cardwalls.

Our two studies were done independently, and our two groups only began working together afterwards. The first study was done in North America, and the second in Europe. The studies explored the present and, especially together, suggest a possible future.

The present, as examined in our first study, is that many teams use a form of digital cardward, typically a display based on a workflow tracking system, such as JIRA, and typically present the Cardwall using a projector during team meetings for sprint planning and reviews. On the basis of our observations and interviews, our study identified a number of guidelines for improving the experience. The guidelines fell into several categories, repeated in the list below.

1. Cardwall Formats
2. Scaling the Design of the Cardwall
3. The Big Picture and Story Relationship Visualizations
4. Exploring and Filtering Information
5. Managing Backlogs
6. Multi-Disciplinary use of Stories
7. Updating the Cardwall

The first and second category concerned size, emphasizing flexibility, but especially the ability to scale up to sizes common in physical cardwalls and the importance of designing for team interaction. The third and fourth categories involved support for seeing the big picture and relationships: the team needs to see the situation overall, without clutter, but also needs to explore relationships and filter information. Both of these abilities support team discussions. The fifth and sixth categories identified the need to include information sometimes ignored, such as the backlog, or work that is done that is not software development work, but adds value. Finally, the seventh guideline is a reminder of the importance of synchronizing records, such as integration with workflow management systems or a distant digital cardwall.

The second study looked to a future that could begin now. The study examined the aWall system [39], which attempts to use currently available technology to create a digital cardwall to compete with physical cardwalls. The aWall prototype was designed around an array of four 46 inch 4 K (3,840 × 2,160) displays. The total display is therefore 7680 × 4320 pixels, and 92 inches diagonally. This is comparable in both resolution and size with many physical cardwalls. Moreover, the aWall uses touchscreen technology, allowing immediate and multiple simultaneous interaction.

A study of aWall involved professional developers assessing realistic Agile team tasks that require a cardwall, and the findings were positive overall. In particular, there is much alignment between comments from the participants and the guidelines identified in our first study. For example, the size and resolution receive positive comments and indicate they are sufficient for a real-world application, offering both project coverage and readability. Moreover, there were also comments on the presentation of stories on the aWall, offering an uncluttered view of the main content, but with easily discoverable features that allow exploration. The abilities for filtering were seen as very useful by participants. All these also align with guidelines from study one. Lastly, the support for other meetings was seen as important, and the integration with JIRA was praised for supporting updates from either system to maintain a shared state.

Despite our two studies being done independently, the result are strongly in alignment, and suggest the same implications: that there are certain issues that must be addressed for digital cardwalls to be successful; and that we now have both the technology and the knowledge to make that a reality. The time for digital cardwalls may now have arrived.

13.6 Conclusions

In this chapter, we presented two studies that inform the design of large multi-touch digital cardwalls for software development. Both studies were empirical. One was in the workplace and the other was conducted in a lab setting. The results of the studies were design guidelines and considerations for digital agile cardwalls.

The first study in this chapter was an ethnographic study of digital cardwall users. Observational data of teams at work and interview data was collected from 8 Scrum teams (64 participants) in Canada and the United States. The study identified the current needs of digital cardwall users and the difficulties they experienced with their digital cardwalls. The seven themes were: (1) the need for digital cardwalls to be designed for multiple formats to support different types of meetings and remote team members, (2) the need to design cardwalls that will support large projects with many stories, (3) the need for a big picture view that can also display relationships between stories, (4) the desire to explore and filter sets of stories to support decision-making, (5) the need for specialized functions for managing backlogs, (6) the potential for cardwalls to support team members in other roles, and (7) that a usability cardwall is one that updates automatically, keeping everyone up to date on the status of the work.

Guidelines for the design of large multi-touch digital cardwalls emerged from each theme. These guidelines are valuable for prioritizing the work of digital cardwall developers.

The second case study—*aWall*—presents the design and implementation of a digital cardwall for large multi-touch displays. It discusses the design considerations for the implementation, the user interface and the evaluation of the current implementation. The design considerations are based on a previous study with in-depth interviews with 8 teams (44 participants) in Switzerland [38]. *aWall* was designed with a special focus on supporting co-located and distributed Agile teams. *aWall* provides a collaborative workspace using large multi-touch displays, information transparency, direct information interaction without the need for navigation, support for the whole Agile process, and dedicated views for different types of meetings. The study we presented here evaluated aWall with 11 professional software developers from varied industries, and shows that the Agile practitioners especially valued the large-size of the wall due to the physical space affordances, the dedicated views with context-specific information, and the always visible and direct information access.

The studies show that neither physical cardwalls nor current digital cardwall tools seem to provide all affordances needed by the highly interactive and collaborative Agile software development approach, neither for co-located nor for distributed teams. On the other side, the second study seems to show that todays technology for large displays and multi-touch interactive surfaces seems to have the potential to provide the needed tools for the highly collaborative workstyle. Both studies provide basic design guidelines, but nonetheless more research work is needed.

References

1. Agile Manifesto. Web, March 2016. http://agilemanifesto.org/
2. Kropp M, Meier A, Biddle R (2016) Agile adolescence to maturity: experience leads to collaboration. Presented at OOP Conference, Munich, 2016
3. Whitworth E, Biddle R (2007) The social nature of agile teams. In: Proceedings of the Agile software development conference, Washington D.C. IEEE
4. Azizyan G, Magarian MK, Kajko-Matsson M (2011) Survey of Agile tool usage and needs. In: Agile conference (AGILE), 2011, pp 29–38
5. Sharp H, Robinson H, Petre M (2009) The role of physical artefacts in agile software development. Interdisc J Human-Comput Interac 21(1–2)
6. Thomas RG (1996) Green and Marian Petre. Usability analysis of visual programming environments: A 'cognitive dimensions' framework. J Visual Lang Comput 7(2):131–174
7. Cockburn A (2006) Agile software development: the cooperative game, second edition. Addison-Wesley Professional
8. Beck K (1999) Extreme programming explained. Addison-Wesley Professional
9. Fog Creek Software (2016) Trello. Web, Feb 2016. http://trello.com/
10. Atlassian (2016) JIRA. Web, Feb 2016. http://www.atlassian.com/software/jira
11. Wang X (2009) Using digital tabletops to support agile project planning. Master's thesis, University of Calgary, Department of Computer Science, Calgary, Alberta, Canada, September 2009

12. Weber S, Ghanam Y, Wang X, Maurer F (2008) Apdt- an agile planning tool for digital tabletops. In: Proceedings 9th international conference on agile processes and eXtreme programming in software engineering (XP2008)
13. Morgan RE (2008) Distributed agile planner: a card-based planning environment for agile teams. Master's thesis, University of Calgary, Department of Computer Science, 2008
14. Smart Technologies (2010) Digital vision touch. http://www.smarttech.com/
15. Schöning J, Brandl P, Daiber F, Echtler F, Hilliges O, Hook J, Lchtefeld M, Motamedi N, Muller L, Olivier P, Roth T, von Zadow U (2008) Multi-touch surfaces: a technical guide. Technical Report TUM-I0833, University of Munster, 2008
16. Gossage Stevenson (2010) CMAP: a collaborative multitouch Agile planner. University of Carleton, School of Computer Science, Honours project
17. Scott SD, Grant KD, Mandryk RL (2003) System guidelines for co-located, collaborative work on a tabletop display. In: Proceedings of the eighth conference on european conference on computer supported cooperative work, ECSCW'03, Norwell, MA, USA, 2003. Kluwer Academic Publishers, pp 159–178
18. PyMt.eu. PyMt, Multitouch Framework. http://pymt.eu, December 2010
19. Esbensen M, Tell P, Cholewa JB, Pedersen MK, Bardram J (2015) The dboard: a digital scrum board for distributed software development. In: Proceedings of the ACM international conference on interactive tabletops and surfaces (ITS). ACM, pp 161–170
20. Anslow C, Marshall S, Noble J, Biddle R (2013) SourceVis: collaborative software visualization for co-located environments. In: Proceedings of the international working conference on software visualization (VISSOFT). IEEE
21. Bragdon A, DeLine R, Hinckley K, Morris MR (2011) Code space: touch+air gesture hybrid interactions for supporting developer meetings. In: Proceedings of the ACM international conference on interactive tabletops and surfaces (ITS). ACM, pp 212–221
22. Gossage S, Brown JM, Biddle R (2015) Understanding digital cardwall usage. In: Agile conference (AGILE), 2015. IEEE, pp 21–30
23. Bernard HR (2011) Research methods in anthropology. AltaMira Press
24. Lofland J (2006) Analyzing social settings: a guide to qualitative observation and analysis, chapter 5. Wadsworth/Thomson Learning, 4th edition, pp 81–116
25. Sanjek Roger (1990) Fieldnotes: the makings of anthropology. J Hist Behav Sci 28(3):274–276
26. Fontana A, Frey JH (1994) Interviewing: the art of science. SAGE Publications, pp 361–376
27. Rubin HJ, Rubin I (1995) Qualitative interviewing: the art of hearing data, chapter 5. Sage Publications, pp 93–121
28. Smith JA (1995) Rethinking methods in psychology, chapter 2. Thousand Oaks, pp 9–26
29. Corbin J, Strauss A (2008) Basics of qualitative research: techniques and procedures for developing grounded theory. SAGE Publications
30. Gubrium JF, Sankar A (1994) Qualitative methods in aging research, vol 168, chapter 12. SAGE Publications, pp 189–210
31. De Almeida RA, Pillias C, Pietriga E, Cubaud P (2012) Looking behind bezels: French windows for wall displays. In: Proceedings of the international working conference on advanced visual interfaces, AVI '12, New York, NY, USA, 2012. ACM, pp 124–131
32. Endert A, Bradel L, Zeitz J, Andrews C, North C (2012) Designing large high-resolution display workspaces. In: Proceedings of the international working conference on advanced visual interfaces, AVI '12, New York, NY, USA, 2012. ACM, pp 58–65
33. Jagodic R (2011) Collaborative interaction and display space organization in large high-resolution environments. PhD thesis, University of Illinois at Chicago, Chicago, IL, USA, 2011. AAI3551257
34. Jagodic Ratko, Renambot Luc, Johnson Andrew, Leigh Jason, Deshpande Sachin (2011) Enabling multi-user interaction in large high-resolution distributed environments. Future Gener Comput Syst 27(7):914–923 July
35. Jakobsen MR, Haile YS, Knudsen S, Hornbæk K (2013) Information visualization and proxemics: design opportunities and empirical findings. IEEE Trans Vis Comput Graph 19(12):2386–2395

36. Yost Beth, North Chris (2006) The perceptual scalability of visualization. IEEE Trans Vis Comput Graph 12(5):837–844 Sept
37. Shneiderman B (1996) The eyes have it: a task by data type taxonomy for information visualizations. In: Proceedings of the 1996 IEEE symposium on visual languages, VL '96, Washington, DC, USA, 1996. IEEE Computer Society, p 336
38. Mateescu M, Kropp M, Greiwe S, Burkhard R, Vischi D, Zahn C (2015) Erfolgreiche zusammenarbeit in agilen teams: Eine schweizer interview-studie ber kommunikation, 22 Dec 2015. http://www.swissagilestudy.ch/studies
39. Mateescu M, Kropp M, Burkhard R, Zahn C, Vischier D (2015) awall: a socio-cognitive tool for agile team collaboration using large multi-touch wall systems. In: Proceedings of the ACM international conference on interactive tabletops and surfaces (ITS). ACM, pp 361–366
40. Beck Kent, Cunningham Ward (1989) A laboratory for teaching object oriented thinking. ACM Sigplan Notices 24(10):1–6

Chapter 14
Collaborative Interaction with Geospatial Data—A Comparison of Paper Maps, Desktop GIS and Interactive Tabletops

Sebastian Döweling, Tarik Tahiri, Jan Riemann and Max Mühlhäuser

Abstract Over the last two decades, researchers have thoroughly investigated the benefits and challenges of large interactive surfaces, highlighting in particular their potential for efficient co-located collaboration and coping with rich content (complex diagrams, multi-layer digital maps, etc.). However, comparative studies that actually evaluate the same tasks on tabletops and other types of systems are still scarce. We have identified crisis management (CM) as promising application context, in which to study such tasks. In CM, people from different organizations use, among others, large paper maps to establish a common understanding of a critical situation, and plan and coordinate appropriate countermeasures. What sets CM apart from other application areas are the very formalized (and different) user roles, and the variations in completeness of the operational picture between involved organizations, both necessitating regular information exchange and collaboration in planning. Based on these characteristics, we have designed a system for interactive tabletops that facilitates collaborative situation analysis and planning by users having different information and planning functionality available. We have then conducted a comparative study, in which 30 participants performed tasks reflecting actual CM work on the tabletop system, classical paper maps and an off-the-shelf desktop GIS. Our goal was to quantify the benefits of tabletops w.r.t. performance, usability, and teamwork quality. We found that users were most efficient using the tabletop and perceived its UX as superior; also, the tabletop

S. Döweling (✉) · T. Tahiri
SAP SE, Strategic Projects, Darmstadt, Germany
e-mail: sebastian.doeweling@sap.com

T. Tahiri
e-mail: tarik.tahiri@outlook.com

J. Riemann · M. Mühlhäuser
Telecooperation Group, TU Darmstadt, Darmstadt, Germany
e-mail: riemann@tk.tu-darmstadt.de

M. Mühlhäuser
e-mail: max@tk.tu-darmstadt.de

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_14

offered a teamwork quality comparable to classical paper maps. This indicates that tabletops may indeed be a valuable tool for collaboration in crisis management, and, more generally, for all application areas in which users with different roles collaborate around geospatial data.

14.1 Introduction

Research into interactive tabletops, or, more generally, large interactive surfaces, has established that these devices provide great potential for efficient interaction with rich content and effective collaboration: their size (and resolution) were found to increase the efficiency of navigation [2], in particular in spatial data [30], and the spatial proximity of users during interaction with them has positive effects on communication and collaboration [10]. Also, there is evidence that they increase participation, make interaction more fluid, and allow more equitable decision-making and access of information [25]. Researchers have highlighted their potential for collaborative interaction during brainstorming [6], but also for more complex tasks like business process modeling [8]. However while individual designs and technique have been studied quite well, there is currently a lack of comparative studies evaluating the same task(s) on both tabletops and other types of systems [4].

To address this, we have studied collaborative work in crisis management, which has been identified as a promising application domain for interactive tabletops [9, 16, 22, 23]. In crisis management, people from different public safety organizations (e.g. fire brigade, police, Red Cross) assemble in response to crisis and disaster situations, typically in a dedicated command and control room. Here, they collaborate as required to establish a common understanding of the situation in the field, and plan and coordinate appropriate countermeasures. As geospatial information plays a crucial role in crises response scenarios, one of the central artifacts for this collaboration is a large paper map with pins and paper-symbols representing the current understanding of the situation in the field. What sets crisis management apart from other application areas are the very formalized, and very different, user roles and responsibilities; also, as there is no single chain of command, the current understanding of the situation may vary in completeness and recency between involved organizations. This necessitates regular information exchange and collaboration in planning to effectively address the crisis situation. Generalizing on this from an interaction perspective, we deal with a situation in which multiple users with access to different (role-specific) information sources and system functions have to collaborate to make sense of geospatial data and conduct planning tasks that correspond to (hypothetical) actions in the real world.

In accordance with the previously discussed lack of studies comparing tabletops and other systems, our primary contribution is a comprehensive evaluation we conducted with 30 participants, who performed the aforementioned analysis and planning tasks using a paper map, a desktop GIS and a self-developed

Fig. 14.1 Study participants interacting with the our tabletop system for crisis management teams

tabletop-based system. As we were primarily looking at the named systems from an interaction perspective, we focused on an analysis of performance, usability, and teamwork quality in this experiment.

As a secondary contribution, we also report on the interaction techniques and designs that we created as part of the development of our tabletop solution (an example of the system in use is shown in Fig. 14.1).

The rest of this chapter is structured as follows: first, we shortly summarize requirements for collaborative work in crisis management teams (CMTs) and discuss related work. We then elaborate on the concepts and techniques we created or tuned during the design process of our own tabletop system for CMTs. Subsequently, we report on the named controlled experiment. We conclude with a summary and suggestions for future research.

14.2 Requirements for CMT Support Solutions

Crisis management requires the collaboration of a variety of people with different roles, often across organizational boundaries. As geospatial information plays a crucial role in crisis response scenarios, one of the central artifacts for this collaboration is a large (paper) map with pins and printed or hand-drawn symbols representing the current understanding of the situation in the field.

As part of previous work, some of the authors have conducted a comprehensive analysis of how these characteristics translate to requirements for a tabletop software that supports collaborative work in CMTs [9] ([26] have arrived at a similar list for collaborative decision making in maritime operations); we will use these requirements as a basis for the design of our CMT software described later on:

R1. Provide easy access to geographic information from standardized services
R2. Support collaborative planning, i.e. discussion and collaborative annotation and creation of geospatial information
R3. Provide means of synchronization of information across different devices as well as between different organizations
R4. Respect the roles and responsibilities present in the current workflows within CMTs (cf. [26])
R5. Allow attribution of actions to individual users and roles
R6. Allow easy, but selective information sharing (i.e. information should remain within the control of its owner unless explicitly shared with others) (cf. [5])
R7. Adopt familiar concepts and metaphors to reduce learning effort (cf. [18])
R8. Allow precise interaction (to exactly specify the coordinates of mission-critical annotations)
R9. Use commercially available hardware where possible to make acquisition and easier long-term support

14.3 Related Work

In our review of related work, we have focused on two areas: existing tabletop-based solutions that support crisis management and studies that compare the same task on tabletops and other systems.

14.3.1 Tabletop Systems for Collaborative Work in Crisis Management

A number of existing systems facilitate collaborative situation analysis and planning in crisis management scenarios on tabletops. Based on the requirements listed above, we list only those with the highest amount of requirements met:

The useTable [22] is a custom-built (~~R9~~[1]) tabletop with an integrated Anoto micro-dot patterns designed to support CMTs. It allows users to interact via touch input, tangibles and digital pens. Interactions beyond navigation are controlled with a physical puck that needs to be passed for coordination purposes. Annotations can

[1]Crossed out requirement numbers indicate a requirement is not met.

be created with the pens. Thus, the useTable meets the requirements regarding familiar metaphors (R7) and precise interaction (R8), as well as those for access to geographic information (R1) and support for collaborative planning (R2). Although not reported, pen interaction could be attributed to individual users (R5) via Anoto pen IDs, assuming pens are personal. Limitations of the system exist regarding role-specific interaction (~~R4~~), information sharing (~~R6~~) and synchronization (~~R3~~).

A similar system is CoTracker [16], which supports collaboration in CMTs (R2) with a custom-built (~~R9~~) tabletop that is complemented with physical widgets resembling typical artefacts of paper work (e.g. rulers); digital pens are also supported. Thus users find themselves supported with familiar metaphors (R7) and precise interaction (R8). Attribution of interactions (R5) is possible as pens and tangible controls are personal, but there is no support dynamic remapping at runtime. No explicit support for different work roles is given (~~R4~~), neither are there means for controlled information sharing (~~R6~~) nor for integration (~~R1~~) or for synchronization (~~R3~~) of standardized geospatial services.

uEmergency [23] is a system based on a multi-user adapted version of Google Maps, which runs on a custom-made very large tabletop (381*203 cm). It is comparable to CoTracker both regarding features and requirements met.

The system presented in [1] also uses a custom-made tabletop (~~R9~~), which can be combined with so-called Fovea tablets displaying higher-resolution versions of an area on the screen. The tablets are also used to access role-specific information and create annotations on the map (R4). This way, actions could also be attributed to users—however, this feature has not yet been implemented (~~R5~~). A basic form of information sharing (from tablet to tabletop) is supported, but selective sharing is not (~~R6~~). The main limitation is that users cannot create annotations directly on the tabletop, but have to use the tablets which have thus to be moved prior to each interaction.

As the requirements listed above were taken from [9], it seems natural that the coMAP-system presented as part of this work fulfills all the identified requirements. coMAP runs on an Samsung SUR40 off-the-shelf tabletop (R9), supporting collaboration in CMTs (R2) with an application built on top of Microsoft Bing Maps; using a computer-vision based technique, Anoto pens can be used on the tabletop for precise interaction (R8) based on a familiar metaphor (R7); additional concepts from CMT work are taken up with a role-specific menu (R4) using established vocabulary and ID badges allowing users to quickly change their role with the pen. The same badges are also Bluetooth-enabled, and form part of a tracking system that is used to attribute interactions to individual users (R5). Selective sharing is possible via drag-gestures between personal menus; tracking information is used to also handle implicit sharing (i.e. personal information being visible to everybody when shown on the tabletop). Information integration (R1) and synchronization (R3) are supported by a custom implementation of OGC Web Feature Services (WFS); support for OGC Web Map Services (WMS) is reported to be in the making.

While the coMAP system provides the most complete coverage of the identified requirements, there is still room for improvement: first, the custom implementation of the OGC standards creates maintenance overhead as there are many variants (in particular of the WFS standard), and thus partly works against the goal of minimizing maintenance efforts (R9). Then, attribution of interactions only works for pen interaction, reducing the benefit of multi-touch interaction and, to a certain extent, working against the goal of fluid collaboration (R2). We will address this as well as a number of smaller usability issues in the revised design presented later on.

14.3.2 Studies Comparing Tasks on Tabletop and Other Systems

As initially mentioned, studies that actually compare the same task or set of tasks on tabletops and other devices are currently scarce. [4] mention [25], where a travel planning task on an interactive tabletop and an interactive wall display is compared, finding that the tabletop affords more frequent role-changes and equitable decision making; in a later study (using a garden planning task), it is also found that laptop computers lead to higher inequity in contributions compared to an interactive tabletop [24]. In addition, [4] also presents findings from experiments on an idea generation task, which indicate that sitting/standing around a table is beneficial in general (also without a digital interface), but that paper still had better scores for equitable contributions.

When looking at shared display/device (typically for large interactive surfaces) versus separate devices (typical for desktop/laptop PCs), [10] showed that communication and collaboration suffer when users work at separate devices (in this case: an interactive wall display and a desktop computer) as compared to working at the same device (again, a travel planning task was used).

In general, none of the above studies compared the same task on more than two device types, e.g. tabletop, desktop, and paper. Specifically for the case of crisis management, no evaluations comparing typical crisis management tasks on the named devices have been conducted, yet. We will address this with the evaluation discussed later on.

14.4 A Revised Version of the coMAP Tabletop-System for CMTs

Based on the requirements established and our related work analysis as well as practical considerations, we chose to use the coMAP-system [9] as the starting point for a new tabletop-based support tool for role-based collaboration on geospatial data, as found e.g. in CMTs. As we had access to the source code, we re-used, with minor modifications, the implementation of the pen input technique for camera-based tabletops and the sensor part of the Bluetooth-tracking. All other features were re-implemented based on the specifications in [9] and evolved from there based on internal user testing. Because of its clear roots, we simply labeled our system coMAP2.

In the following, we describe the central features of coMAP2, focusing on aspects where it differs from the original coMAP system (which we will label coMAP1 for disambiguation), but including additional description where it aids the understanding. We will start with the two areas identified as candidates for improvement in the related work section, i.e. data integration and synchronization, and personalized touch input, then discuss techniques for data sharing, and finally discuss additional usability improvements.

14.4.1 Data Integration and Synchronization

Similar to coMAP1, the base of coMAP2 is a digital map operated with pen and touch interaction. However, while coMAP1 used the Microsoft Bing Maps Control and added support for standardized geospatial services with custom implementations, we based our implementation on the *ArcGIS Runtime SDK for.Net* for the tabletop client and *ArcGIS for Server* for the geospatial data server. Among others, this provided us with means for easy integration and synchronization of both Web Features Services (WFS) and Web Map Services (WMS) with full compliance to each of the standards (coMAP1 only supported newer versions of the WFS standard). Moreover, since ArcGIS is a commercial product with vendor support, updates to the standards can be easily integrated with an SDK update.

14.4.2 Personalized Input

As discussed in the requirements analysis, collaborative work in CMTs results in the need for personalized input/interaction, i.e. the system should be able to attribute interactions to an existing user account. The coMAP1 system had support for personalized pen input, but not for personalized touch input. In internal usability tests of an early development version of coMAP2 that basically mirrored the features of coMAP1, we found that the lack of personalized touch input was one of the major irritations that users encountered when using the system.

14.4.2.1 Personalized Touch Input

With the requirement for simplified maintenance and support by the use of commercial components (R9) in mind, we thus looked for a way to provide personalized touch input without adding custom hardware.

Existing research has already established a number of techniques for personalized touch input; however, most of them are either not compatible with the camera-based touch recognition used in many tabletops (e.g. [7]) or require additional hardware (e.g. [19]). While, in the long run, fingerprint-based detection (e.g. [14]) would be ideal, it has yet to be implemented in commercial hardware.

As a consequence, we have designed a new concept for personalized touch input on typical camera-based tabletop systems, which does not require additional hardware (beyond what is needed for the Bluetooth-based user tracking introduced with coMAP1). It builds on the idea presented in [31], where a machine-learning approach is used to identify user positions (but not user identities) based on finger orientation. However, we opted to use a computationally more light-weight approach that builds on touch blob orientation, and leverage the user location obtained from the named tracking service to also establish user identity.

Touch blob orientation has already been used for automatic orientation of interactive elements [29]; however, we have found it can also serve as a coarse-grained, but reliable prediction of the touching user's location relative to the tabletop. Based on the observation, that, in a rough approximation, the arm and hand of users touching the screen form a straight line, we can infer the side of the device they are located at by mapping the touch blob angle to four 90° sectors (starting at 45°) around the display. Once this estimated location is known, it can be matched against the measured location from the location service to identify users and subsequently associate the input event with their account.

Using the personalized touch interaction, users can now operate their personal role-aware menu [9] using touch and also share services with a touch drag gesture (previously this could only be done with a pen). Drawing on the map is, in the current implementation still limited to the pen; however, users can also create new objects of standardized size with a drag gesture from the menu.

Handling Touch Imprecisions and Accidental Touch Events

Two issues that can occur when users interact with a system via touch are imprecisions, i.e. touch targets either being too small for users to hit them precisely with their fingers or users trying to drag an object at its edges but slightly missing them, and accidental input, i.e. touch events caused by a user's palm or clothing.

While a number of techniques have been proposed for dealing with touch imprecisions (e.g. [3]), they typically impose overhead, i.e. additional interactions, and thus slow down work with the system. Thus, in practice, touch targets are typically designed large enough to avoid this issue. From a visual design standpoint this may not always be desirable and can lead to interfaces that are perceived as clunky. We have therefore decided to implement invisible buffers around critical elements (e.g. menus and virtual keyboard) that forward touch events to the original element, thereby increasing touch target-, but not visual size (cf. [8]). As the buffers are specified at design time, they can be used with minimal overhead. This approach might become obsolete in the long run, though, when more advanced touch recognition [13] becomes commercially available.

In a similar way, we also realize palm rejection: when a user interacts with a critical element, we create a temporary touch-blocking overlay below the user's touch contact point, thus preventing accidental interaction.

14.4.2.2 Personalized Pen Input

As discussed in the requirements, there is a need for interaction concepts that allow users to work with digital maps with the same ease that is common for the use of pens on paper maps. Similar to touch input, the respective input concept should be personalized. Both requirements can be fulfilled by a solution based on Anoto digital pens. coMAP2 inherits the technique developed for this purpose in coMAP1. The advantage of this approach is that it works in software only, i.e. it does not require custom-built tabletop hardware, thus allowing the use with commercially available tabletops that use camera-based touch sensing (such as the Samsung SUR40 we used for our development).

The technique is based on a three-step approach: first, heuristics on pen input event characteristics are used to determine whether a pen made contact with the screen. Then, an image-processing algorithm is used on the image from the touch-sensing camera to determine, where the pen actually made contact with the screen. Once the association between a pen and its screen coordinates has been made, subsequent pen movements are determined by a continuous analysis of the image of the touch camera (until a pen up event occurs) for light spots caused by Anoto pens; the corresponding screen coordinates are then matched against the last known pen coordinates (using a spatial hashing mechanism for improved performance). More details on this technique can be found in [9].

Figure 14.2 illustrates the image analysis and how this algorithm allows users to draw smooth lines with Anoto pens on the tabletop screen.

Fig. 14.2 (*Left*) Touch camera image of pen and hand/arm (after grayscale conversion and filtering); (*right*) map annotations drawn with an Anoto pen on a Samsung SUR40 tabletop

14.4.3 Data Sharing Techniques

Similar to the technique for pen input, coMAP2 also inherits the techniques for data sharing from coMAP1. As discussed in the requirements, collaborative decision making in CMTs, or more generally collaborative analysis and planning on geospatial data by users who can access different information and system functions, often requires information to be shared. To support this while still allowing users to keep some control over shared information, coMAP2 includes techniques for implicit and explicit sharing of both information services and system functions.

Implicit sharing happens when users activate a service from their personal menu for which other users at the screen do not have sufficient access rights, and thus effectively share their read access to that service. Information from such a service is only visible as long as the activating user is in zone 1, i.e. relatively close to the table, to ensure this user has at least peripheral awareness of its use. When the user leaves that zone, information from the service is faded out.

To avoid this effect, users can *explicitly share* the respective service with trusted collaborators. We have designed a simple drag-to-share gesture as the universal way of explicit sharing—it works with both pen and touch and can be applied to both information services and system functions (technically, the latter equals write access to the respective services). The default for *explicit sharing* "is read-only with authorized writing", i.e. information can be displayed, but not manipulated without the consent of a user with the necessary access rights (typically the user who shared the service); however, users can also decide to share information explicitly as read-only (without any option for writing)". This way, users maintain control over their information and accountability is guaranteed, even when information is shared. The respective authorization dialogs can only be confirmed by users with the necessary access rights; they orient automatically towards these users. In the personal menu, shared services are indicated by an icon for the organization of the user who shared the service. Example for sharing are shown in Fig. 14.3.

14.4.4 Additional Usability Improvements

In addition to the two major improvements discussed above, we added or refined a number of features based on feedback we collected in internal usability testing (some of which were already discussed in [9]).

1. *Location search & virtual keyboard*: We have added a global search box on the top left of the screen (screen orientation) that allows users to search for and directly jump to a location (users can pick locations from a list of search results that is rotated according to their orientation). To start a search, users tap the box and bring up a custom virtual keyboard (based on the design and the processing pipeline presented in [8]), which is automatically placed and oriented towards the user, but can be moved and rotated to the user's convenience; we also

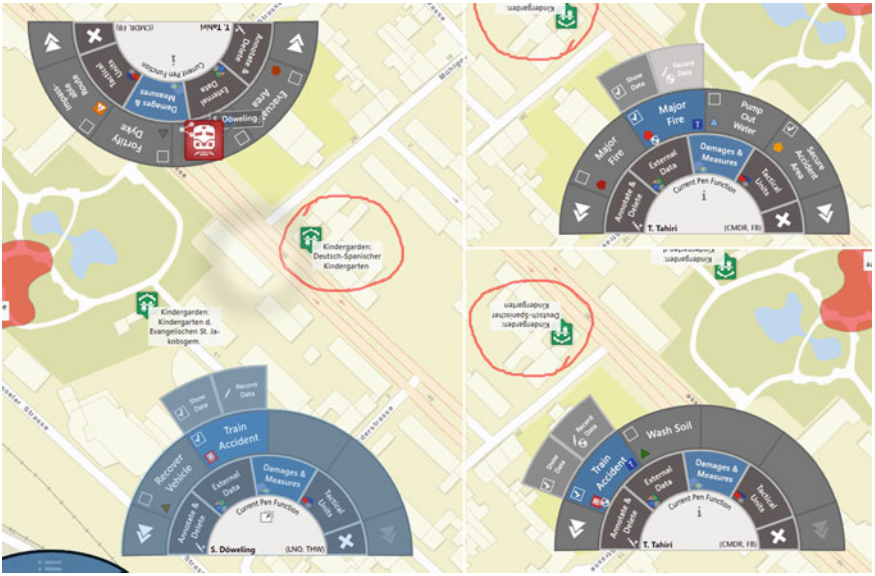


Fig. 14.3 (*Left*) Service icon during sharing gesture; (*right*, *top*) service shared as read-only; (*right*, *bottom*) service shared as read with authorized write

implemented word suggestions using common English words as well as the names of major German cities. Search works in the typical search-as-you-type way.

2. *Interactive legend*: we have also added an interactive legend showing currently active layers on the map, that also allows users to change the base map and to disable labels for specific layers.
3. *User widget*: Our testing indicated a lack of visual feedback that a user's position at the tabletop was recognized correctly; we thus designed a widget in the form of a half-ellipsis always placed where the user is currently standing (as detected by the Bluetooth tracking). We included information about user role and active pens, and also allow access to the personal menu from there. Moreover, to avoid unintended navigation in the map, once the desired operation area is on the screen, the user widget allows map locking (the lock is global and can be released by any user). An example of a user widget is shown in Fig. 14.4.
4. *Improved Menu*: We also found a number issues with the personal menu design of coMAP1 (a two-layer pie menu covering 3/5 of the full circle):

   - Users sometimes accidentally switched between menu items on the first level, causing confusion as to why items on the second level changed.
   - When users tried to move the menu, they sometimes targeted the edges of the menu, but missed them because of touch imprecisions.

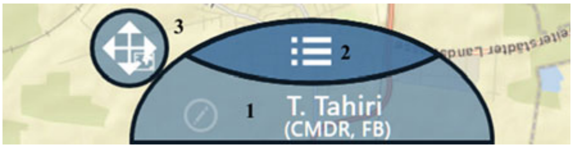


Fig. 14.4 The user widget provides users with basic awareness about their situation (*1*); it also provides access to the personal menu (*2*) and allows locking the current map position (*3*)

- Users sometimes covered the menu's outer parts with their hand, thus missing the option to navigate between pages of items on the second level.
- There was no distinction between activating a layer for reading/viewing and writing; thus, if data was shared, the receiving user always had write access. In stress-intense moments, this might also lead to accidental creation of data on the map.

The first two issues are addressed with the techniques for handling touch imprecisions and palm rejection (see Sect. 14.4.2); to address the other two, we have redesigned the menu into a two-and-a-half layer menu (the third layer has only two elements for read and write access) resembling the stacked half-pie menus in [11]. However, contrary to that work, our menu orients automatically towards the user to avoid occlusion (as in coMAP1).

5. *Data compass*: We observed users having problems to assess distances and directions to resources needed to fulfill a planned task. We thus designed a compass-shaped widget (the "data compass") that provides users with this information; it is activated when a user enables a service in the personal menu or selects one in the interactive legend. It shows distance and direction to the closest object on the layer; if objects from the service are visible on the current screen, they are also highlighted by black arrows to make it easier to spot them on crowded maps. Zoom buttons have been integrated for quick navigation. An example of a data compass is shown in Fig. 14.5.

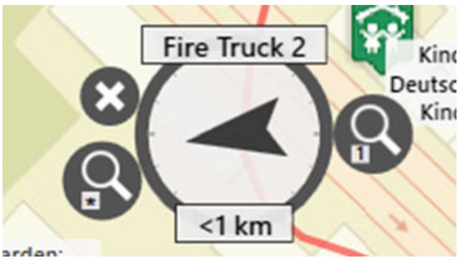


Fig. 14.5 The data compass supports orientation and distance assessment for mission critical objects; buttons allow users to directly navigate to the respective object(s)

14.5 Evaluating Paper Maps, a Desktop GIS and Tabletops for Collaborative Work with Geospatial Data

As outlined at the beginning of this paper, our main objective in this work is the evaluation of interactive tabletops against other systems for the same task(s). Driven by a potential application of tabletops in crisis management, we are specifically looking at series of situation analysis and planning tasks on geospatial data, conducted by users with access to different (role-specific) information sources and system functions. Besides interactive tabletops, we included classical paper maps (which are the currently predominant tool for collaboration in crisis management) and desktop geospatial information systems (GIS) (which are sometimes proposed to improve efficiency in crisis management operations).

In particular, we are interested in evidence for our hypothesis that a tabletop-solution can be as efficient as a desktop GIS, but offer a similarly simple user experience and comparable degree of teamwork-support as a paper map in the outlined type of tasks. To do so, we conducted a lab study, in which 30 users worked in groups of two with systems of all three types; tasks were derived from typical work steps in actual crisis management work.

14.5.1 Apparatus

Tabletop solutions were represented by coMAP2 in this study; we ran the system on a Samsung SUR40 with Microsoft Pixelsense (40″, FullHD screen). Paper maps were represented by a set of map printouts of approximately the same size (one for each of the scenarios), placed on a normal office table[2]; they were accompanied by a number of tabular printouts with information that would normally come from the field or internal resource management systems. We also provided a few labeled wooden blocks for tactical units (maps were placed horizontally on the table, thus the use of blocks instead of pins), and a set of (erasable) colored pens for drawing. For the desktop GIS, we used a recent version of ESRI's *ArcGIS for Desktop*, the arguably most widely used commercial desktop GIS. We installed it on two typical computers (Core i5, 8 GB RAM, 120 GB SSD) with two screens attached (22″ + 19″); data sharing was possible via an instance of *ArcGIS for Server* installed on one of the machines and pre-configured in both desktop installations. The computers were placed on two office tables that were facing each other; screens were placed in a way that allowed users to look over or lean beside the screen for

[2]The main reason for using a (horizontal) table instead of a (vertical) pinboard was to eliminate any differences, in particular regarding collaboration style, that might result from orientation between tabletop and paper map.

Fig. 14.6 Set up of the different systems/stations (*top left*: paper map, *top right*: tabletop, *bottom*: desktop computers)

easier direct communication. All stations/systems were set up so that participants could move around freely (cf. Fig. 14.6).

14.5.2 Participants

30 participants took part in the study (3 female); the average age was 30.73 years (SD = 5.62). Some participants had practical experience in crisis management from volunteering, yet the general experience (self-reported on a 5-point Likert scale) was low (mean = 1.53, SD = 1.14), as was the experience with ArcGIS (mean = 1.53, SD = 0.90). Experience with tabletops was slightly higher (mean = 2.4; SD = 1.40), but only for paper maps it reached an *average* level (mean = 3.07, SD = 1.20).

14.5.3 Procedure

Participants were grouped into teams of two, forming a total of 15 groups; we recruited users and assigned groups so that users would always know each other (to our best knowledge that also represents the typical case in (German) emergency management organizations, where crisis management teams are often built from experienced officers that typically know each other quite well from previous missions). We chose a within-subject design, i.e. all groups took the roles of two crisis management officers (commander-in-chief of the fire brigade and liaison officer of the German Federal Agency for Technical Relief) in three scenarios (one scenario per system). The order of presentation for the systems was balanced using a Latin squares design to avoid sequence effects; the order of scenarios was balanced where possible, but priority was given to ensuring that all system-scenario pairings occur equally often to avoid system-scenario confusions; roles were swapped after each scenario.

Scenarios were designed so that the number of tasks each participant had to perform was constant (each scenario had 9 tasks), and tasks were of comparable complexity. They were designed by the authors to reflect typical disaster situations and the respective actions in a CMT (e.g. planning the evacuation of a hospital), but also, more generally, typical sense making/situation analysis and planning tasks with geospatial data (an example of such a scenario can be found in Sect. 14.7).

Tasks were grouped in three sections (as indicated by the titles, the first section emphasized analysis, while the other focused on planning; however, the latter also included intermediate steps of collaborative assessment of a situation):

1. situation analysis and data synchronization
2. measures package A (infrastructure)
3. measures package B (population).

To compensate the relatively low expertise in crisis management, the task descriptions contained relevant domain knowledge that helped to decide which action to take (e.g. which type of tactical unit was suitable for a task). Also, all scenarios were placed around the next major city, so that basic geographic familiarity on the side of the participants could be assumed.

Each scenario was preceded by a short introduction, in which two of the authors demonstrated the features of the respective system in a demo scenario. Participants also had the possibility to try out systems for themselves for up to 5 min. When the group was ready, they received the written task descriptions and actual work on the scenarios would start. At this point, one of the authors would also start taking the time with a stopwatch; participants were informed that time was recorded and were asked to complete tasks as fast as possible without sacrificing completeness or accuracy. Participants were also reminded that they may move freely[3] and collaborate as they see fit to complete the scenarios.

During the sessions, two of the authors were available as coaches, in case participants had questions regarding the scenarios (which was rare), or regarding the system they were using (which was more common, especially in the case of the ArcGIS system). Participants were asked to complete all tasks; this was verified by the named authors.

[3]The tabletop was an exception in this regard as the current implementation does not allow two users to stand at the same side; participants were thus asked to avoid such placing.

After each scenario, participants were ask to fill out a questionnaire addressing perceived user experience (UX) and teamwork aspects of the system.

14.5.4 Questionnaire Design

UX-related aspects were assessed using the User Experience Questionnaire (UEQ) [17], which measures UX on the five scales: *attractiveness*, *perspicuity*, *efficiency*, *stimulation,* and *novelty* (each with a −3 to +3 range). We included UX aspects into our evaluation mainly for two reasons: first, a positive UX may promote software adoption (cf. [21, 28]), which is especially helpful in conservative fields such as crisis management that are slow to adopt new technology. Second, we also see the obtained quantitative scores as helpful to support (or challenge) qualitative observations made during an evaluation.

Teamwork was evaluated using the team work questionnaire (TWQ) [12]. The TWQ measures teamwork with the five scales: *communication*, *coordination*, *balance of contributions*, *mutual support,* and *effort* (each on a +1 to +7 scale). As the TWQ was originally designed to measure teamwork quality in long-running projects, we had to adapt it to make sure that the questions matched the comparably short-lived scenario we were presenting (wording was changed to focus on the immediate situation instead of a long-term project; also, questions that could only be answered for long-term collaborations were removed. The modified version can be found in Sect. 14.8).

UX and teamwork questions were preceded by a small number of background questions (age, gender, crisis management experience and experience with the respective system).

14.5.4.1 Hypotheses

Based on existing research as well as our own experience with tabletop systems and desktop GIS, we substantiated our basic assumptions into the following hypotheses (grouped according to common challenges in collaborative work):

Efficiency

Following the observations of [15], we expect that the tabletop system will allow participants to complete the tasks faster than the paper map (*H1a*), and as fast as or faster than the desktop GIS (*H1b*); we expect the desktop GIS to be faster than the paper map (*H1c*).

User Experience

Existing research (e.g. [20, 21]) reports positive perceptions of tabletops. Thus, we reason that participants will find the tabletop system more *attractive* than the paper map (*H2a*) and the desktop GIS (*H2b*); because of its assumed efficiency (see above) we expect that the latter will be perceived at least as *attractive* as the paper map (*H2c*). As we designed our tabletop software in a way that closely resembles interaction with classical maps, we expect *perspicuity* to be at least as good for the tabletop as it is for the paper map (*H3a*) and higher than for the desktop GIS (*H3b*) (which is an expert system); *perspicuity* will be higher for the paper map than for the desktop GIS (*H3c*) (for the named reason). (Perceived) *efficiency* will be higher for the tabletop than for the paper map (*H4a*), and not lower than for the desktop GIS (*H4b*); it will be higher for the desktop GIS than for the paper map (*H4c*) (all because we expected GIS software to increase actual performance). *Stimulation* will be higher for the tabletop than for paper map (*H5a*) and desktop GIS (*H5b*) (for the same reasons as for attractiveness); also, we expect it to be higher for the desktop GIS than for the paper map (*H5c*) (because desktop computers are arguably more state of the art than paper maps). The same will be true for *novelty* (*H6a-c*).

Teamwork

Following existing research on the positive effects of spatial proximity of users during interaction on communication and collaboration [10], we hypothesize that *communication* will be at least as good for the tabletop as it is for the paper map (*H7a*) and that it will be better compared to the desktop GIS (*H7b*) (where users use individual workstations); it will also be better for the paper map than for the desktop GIS (*H7c*) (for the same reason). Based on observations that tabletops increase participation, make interaction more fluid and allow more equitable decision-making and access of information [25], we expect similar effects for *mutual support* (*H8a-c*), *coordination* (*H9a-c*) and *balance of contributions* (*H10a-c*). We do not expect significant differences in *effort* (because of the somewhat artificial scenario of a user study), neither between tabletop and paper map (*H11a*), nor between tabletop and desktop GIS (*H11b*), or desktop GIS and paper map (*H11c*).

14.5.5 Results and Observations

For questionnaire data, a series of Shapiro-Wilk tests revealed that for 4 of the 11 measured scales normal distribution could not be assumed; Levene tests indicated that for 6 of the 11 scales variances were not homogeneous, too. Thus, subsequent analyses were conducted using Friedman's test with post hoc Dunn-Bonferroni tests for pairwise comparison. We found significant differences for all scales except for effort. A detailed discussion of results that also includes qualitative observations can be found below, grouped according to the three areas of analysis, i.e. efficiency, user experience and teamwork. Following this, we also shortly discuss observations regarding the techniques presented in the coMAP2 section.

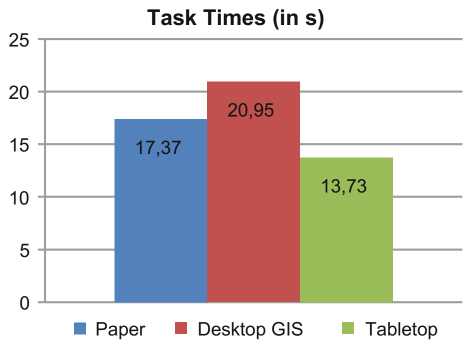


Fig. 14.7 Average task times with the different systems

14.5.5.1 Efficiency/Task Times

Task times were significantly different between systems ($\chi^2(2) = 45.067$, $p < 0.001$), with post hoc tests indicating times using the tabletop were significantly lower compared to the paper map ($p < 0.001$) and the desktop GIS ($p < 0.001$); differences between paper map and desktop GIS were not significant ($p > 0.999$). Thus *H1a* and *H1b* were confirmed, but *H1c* was rejected (cf. Fig. 14.7). Based on our observations, the main reason why the tabletop worked faster on average was the inclusion of GIS functionality, i.e. participants were alleviated from the need to manually integrate information from the field into the map, and also had support in identifying relevant information/resources for their planning tasks. The relatively bad performance of the desktop GIS (which runs counter to the performance improvement found in [15]) is likely to be linked to multiple sources: first, the ArcGIS software we used for purpose is an expert system with a long development history, thus, user experience in many ways does not match that of current consumer software (see also next section); then, as the system has not been designed for quick data sharing, the respective process takes relatively long (data has to be published to the ArcGIS server and then imported on the other workstation); finally, as participants did not have a single display, more communication overhead was required to synchronize actions during the scenario.

14.5.5.2 User Experience

In general, the tabletop system received the most favorable ratings on the various user experience related scales, markedly better than scores for both paper map and desktop GIS (cf. Fig. 14.8).

With regard to individual scales, attractiveness scores were significantly different ($\chi^2(2) = 45.638$, $p < 0.001$); pairwise differences found the tabletop to be significantly more attractive than the paper map ($p < 0.001$) and the desktop GIS ($p < 0.001$), but found no significant difference between desktop GIS and paper map ($p = 0.184$), thus confirming *H2a-c*.

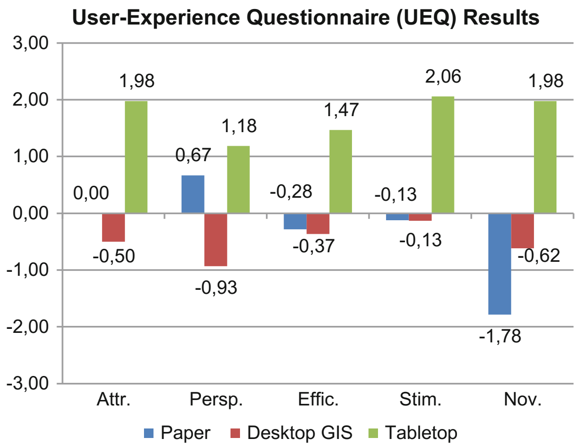


Fig. 14.8 Average user experience ratings obtained via the UEQ (scales are—from *left* to *right*—attractiveness, perspicuity, (perceived) efficiency, stimulation, and novelty)

Perspicuity was significantly different between systems ($\chi^2(2) = 25.373$, $p < 0.001$), with the tabletop ($p < 0.001$) and the paper map having significantly better scores than desktop GIS, but with no significant differences between tabletop and paper map ($p = 0.526$); this confirms *H3a-c*. As mentioned above, the relatively low score of the desktop GIS/ArcGIS software can be attributed to multiple factors: first, ArcGIS is an expert system with a long development history, thus, the user interface is convoluted with options that only a fraction of the users actually need for their workflows. Second some user interactions work different than in currently established consumer software (e.g. zooming with the mouse wheel works the opposite way it does current Web mapping software like Google Maps).[4]

(Perceived) efficiency scores were, again, significantly different ($\chi^2(2) = 35.580$, $p < 0.001$), with the tabletop being perceived as significantly more efficient than both paper map ($p < 0.001$) and desktop GIS ($p < 0.001$), but no significant difference between desktop GIS and paper map ($p > 0.999$); this confirms *H4a+b*, but rejects *H4c,* i.e. the assumption that desktop GIS software would be perceived as faster. This is, naturally, closely linked to the low actual efficiency of the desktop GIS software, which can be partly attributed to the discussed challenges in perspicuity and the lack of efficient data sharing mechanisms; however, we also observed that despite having significantly more processing and graphics power available (the SUR40 we used for our tabletop system uses a relatively low power dual-core AMD CPU, has 4 GB of RAM and is equipped only an HDD; the desktop computers we used had Intel quad-core CPUs, 8 GB of RAM and an SSD drive), the rendering of both base maps and map content was notably slower on the

[4]It should be noted that ESRI recognized this problem and recently started to offer a new software called ArcGIS Pro (http://www.esri.com/en/software/arcgis-pro) which claims to address this problem. We did not have access to ArcGIS Pro at the time of the study, though, and thus cannot report any first-hand experience in this regard.

desktop GIS than on our tabletop implementation (despite both relying, at their heart, on ESRI technology).

Stimulation score differences were significant ($\chi^2(2) = 37.165$, $p < 0.001$), as were pairwise differences between tabletop and both paper map ($p < 0.001$) and desktop GIS ($p < 0.001$), but not between desktop GIS and paper map ($p > 0.999$); thus confirming *H5a+b*, but rejecting *H5c*. This, again conforms to the observations above that the tabletop is seen as an attractive and efficient device that offers meaningful improvement on paper maps, while the desktop GIS does not match the user experience of current consumer software well enough to be perceived as interesting.

Novelty scores were also significantly different ($\chi^2(2) = 51.812$, $p < 0.001$); pairwise differences were significant between tabletop and paper map ($p < 0.001$), tabletop and desktop GIS ($p < 0.001$), and between desktop GIS and paper map ($p = 0.024$). The novelty scale is, however, arguably less relevant as the novelty of tabletops, when compared to paper maps and a desktop software, was evident right from the start.

14.5.5.3 Teamwork

The results obtained from the (modified) teamwork quality questionnaire (TWQ) indicate that the tabletop system was indeed able to facilitate a similar level of teamwork as classical paper maps, and that both allowed better teamwork than the desktop GIS setup (cf. Fig. 14.9).

With regard to individual scales, communication scores were significantly different ($\chi^2(2) = 27.638$, $p < 0.001$), with pairwise differences showing significant advantages of tabletop ($p < 0.001$) and paper map ($p = 0.017$) over desktop GIS, but with no significant differences between tabletop and paper map ($p = 0.051$), i.e. results confirm *H7a-c*. This also reflects our observations: with both tabletop and paper map participants were able to communicate naturally, with pens or fingers

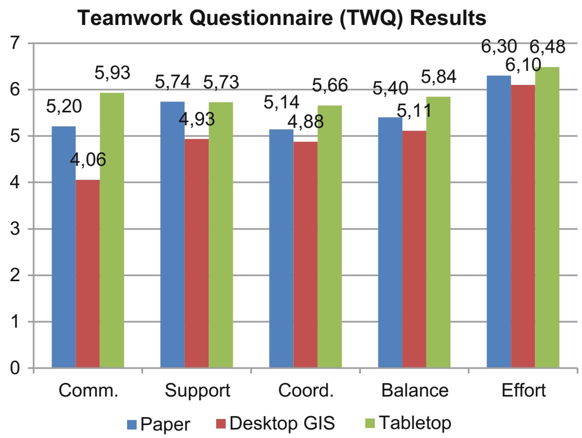


Fig. 14.9 Average teamwork quality ratings obtained via the modified TWQ (scales are —from *left* to *right*— communication, mutual support, coordination, balance of contribution, and effort)

often used to indicate/reference specific objects or areas (we observed the use of the pen for this purpose more often in coMAP; arguably as it could also be used as an input device for menus and dialogs, and participants thus often held it in their hands over longer parts of the sessions). For the desktop GIS, the lack of a shared screen, on which objects or places on the map could be referenced by pointing, caused more communication overhead to achieve the same goal; also, as the computer displays partly blocked participants' sight of their partner, the awareness about the other user's actions was decreased and more verbal synchronization was necessary to align on the current completion of the tasks.

The same factors have likely also contributed to the differences in mutual support scores ($\chi^2(2) = 15.228$, $p < 0.001$), for which tabletop ($p = 0.002$) and paper map again were both rated better than desktop GIS ($p = 0.004$), but no significant difference between tabletop and paper map ($p > 0.999$) were found (this confirms *H8a-c*).

Differences in coordination scores were significant ($\chi^2(2) = 12.302$, $p = 0.002$), with the tabletop ranking significantly better than the desktop GIS ($p = 0.004$), but no significant differences between tabletop and paper map ($p = 0.085$) or between paper map and desktop GIS ($p = 0.905$); this confirms *H9a+b*, but rejects *H9c*. While the advantage of the tabletop over the desktop GIS can be attributed to easier communication, better workspace awareness and the simplified sharing mechanism, the comparably lower ratings for paper maps was unexpected. Based on our observations, we believe that it can at least partly be attributed to the relatively tedious process of integrating information from other sources (which were provided to participants in the form of printed tables and maps), which detracted focus from the shared workspace, i.e. the map. However, further investigation would be necessary to add backup to this hypothesis. One notable phenomenon regarding the coordination of work on both tabletop and paper map was that interaction was less territorial than it could have been assumed based on previous work [27] (in particular for the paper map, but also for the tabletop); based on observations we also made during other studies with interactive tabletops, we believe this may be partly linked to participants posture, i.e. standing/moving freely around the table instead of sitting next to it; however, the way participants organized their work with the tabletop systems leads us to the assumption, that the differences in available information and planning functionality (as represented by different available entries in the personal menu) had an effect on this—in some cases, when discussing next steps in the scenario, participants would share the necessary information in advance, then interact only in their respective territories on the screen; however, in many cases they chose to collaboratively work on the same aspects of a scenario and contribute their data and functionality in an ad hoc way.

Similar to coordination scores, those for balance of contributions differed significantly between systems ($\chi^2(2) = 17.960$, $p < 0.001$), with only the tabletop being rated significantly better than desktop GIS ($p < 0.001$), but no significant differences between paper map and desktop GIS ($p = 0.467$) or those between tabletop and paper map ($p = 0.051$); this confirms *H10a+b*, but not *H10c*. As with communication and balance of contributions, our observations and participant comments indicate that the unexpectedly low scores for the paper map were linked to the effort that had to be invested into searching and integrating external information into the main map; depending on the individual participants speeds in this tasks, this sometimes hampered their ability to fully contribute to all steps in the scenario.

As expected there was no significant difference in the effort participants invested for the respective systems ($\chi^2(2) = 5.246$, $p = 0.073$), i.e. *H11a-c* were confirmed.

14.5.5.4 Interaction Techniques and System Features

As the coMAP2 system, which we developed for the purpose of our study, also featured a number of interaction techniques and system features designed to ease collaborative interaction and work with geospatial data, we were also interested in their performance. For this purpose we relied mainly on observation and video analysis (and occasionally on participant comments).

With regard to the personalized input techniques, i.e. personalized pen input and personalized touch input, we found both to work quite well most of the time.

In particular personalized touch input worked relatively flawless—in a typical session of about 12–15 min, despite frequent use of touch interaction, only 1 or 2 touches would not be registered correctly (in all cases we observed, this event would then not be attributed to any user). However, based on earlier studies we conducted, this has to be attributed partly to the dynamic broadening of the angle segments used for the mapping of touch events to users based on touch blob orientation: in those earlier studies, where we tested personalized touch input without that dynamic broadening, we observed a slightly higher number of malfunctions of the input technique (in one case, the technique would not work as expected in five cases, in a session of comparable length). While this was not a problem in our lab study (participants typically simply repeated their interaction and succeeded), it indicates that further improvements would be desirable for stressful scenarios like actual crisis management work.

Personalized pen input was affected by a slightly higher number of mishaps (2 or 3 events in an average session); however, contrary to personalized touch input, which showed a relatively consistent performance across all groups, this was highly dependent on the individual users. The main problem here was that some participants used only light pressure and made only very brief contact with the screen when trying to interact with the system using the pen; the Anoto pens we use, however, require stronger pressure (similar to actually writing on paper) to trigger events correctly. Thus, the named participants first had to get used to exerting more force when using the pen.

A problem that affects both pen and touch input which is related not to the techniques themselves, but rather to the way the SUR40 tabletop does touch detection, are spurious touch events: when users hold their wrists low while touching the screen, or when they were shirts or pullovers with long sleeves, this sometimes causes unintended touch events in the system (we apply filters based on touch blob size here, but the problem still sometimes occurs). While this problem also occurs not too frequently (3 or 4 times in a session for a typical group—again, very related to the individual interaction styles of users), it was more confusing for users (as it e.g. opened popups, closed menu areas, etc.) and is thus likely to be a potential problem for the use of tabletops in stressful scenarios like actual crisis management work. In a similar way, the susceptibility of the SUR40's touch detection to direct impact of bright light (which may also cause spurious touch events) is not a problem for lab settings, but needs to be considered before deploying a system like ours into productive use.

Other techniques like the explicit sharing via drag&drop worked without issue for most users; occasionally a dragged item would be "lost" on the way due to the aforementioned spurious touches; however, this never occurred more than twice and also affected only a small number of groups. Search box, interactive legend and personal menu were also used without problems—with regard to the latter we found our redesign of the menu into a three level structure (with read and write operations on the third level) to be a notable improvement that eliminated operation confusion problems we saw with the original coMAP design in earlier studies. The data compass was well-understood and -used by most participants; occasionally, participants had trouble to understand the meaning of the "show all" and "show nearest" zoom buttons attached to the compass widget, indicating that it might be necessary to apply some fine-tuning to the design.

The tracking system underlying many of our techniques also worked well most of the time—for a small number of groups we saw one or two misdetections that incorrectly placed a user on the opposite side of the tabletop (from a system perspective); in most of these cases, this seemed to happen when a user leaned over the table or (significantly) to the side during an interaction. In addition, based on observations with the paper map (participants often moved to the same side of the table), we found that the current constraint that only one user may stand at a given side of the tabletop is probably too restrictive. Thus, an extended version of our technique that would allow multiple users to work from the same side of the tabletop seems desirable; this would, however, also require revised versions of the personalized input techniques that take this new possibility into account.

14.5.5.5 Discussion

Our results support the hypothesis that a tabletop-based solution can provide significant advantages when multiple users with different roles with access to different information and system functions collaborate in sense making and planning tasks on geospatial data.

Efficiency + Teamwork

In particular, we found that not only efficiency increases (the tabletop allowed participants to complete the scenarios fastest), but the devices also allow users to maintain the fluidity and ease of teamwork that is typically found in collaborative work with classical paper maps. While this is not a complete surprise given existing research on collaboration around interactive tabletops, our results support this research with quantitative data, and also provide a start to address the current lack of studies comparing the same task on tabletops and other devices (cf. [4]). As the characteristics from which we built up our software and the tasks in our study were derived from actual crisis management work, our study also provides a first indication that interactive tabletops could be a valuable tool for this setting.

For desktop GIS however—contradicting earlier studies on the use of GIS for crisis management [15]—we found the performance to be relatively bad (in fact the performance was slightly, though not significantly, slower than the paper map in our study). Based our observations this is likely to be linked to the user experience issues discussed below. Also, as indicated by earlier research [10] (which compared a collaborative planning task on a shared display and PC workstations), communication was perceived as more difficult with the two workstations, and participants found it harder to mutually support each other in shared tasks. Participants repeatedly commented on the lack of a convenient sharing mechanism as a source of this problem (while a custom GIS solution for desktop computers offering such a sharing mechanism would be generally feasible, such software is currently not available on the market).

User Experience

Similar to the areas of efficiency and teamwork, on the user experience side, our data supports the assumption that interactive tabletops in general, and our coMAP2 system in particular, offer an attractive and efficient platform for collaborative work on geospatial data, and, because of its interface concepts that were tuned the domain, also for crisis management scenarios. Additional support for the latter assumption also comes from task time measurements, which revealed that the given tasks (selected to represent typical actions in collaborative analysis and planning in CMTs), could be completed fastest with the tabletop; also, the study participants' ratings for coMAP2's perspicuity (which matched those of classical paper maps) indicate that relatively little training would be necessary when such a system is deployed to replace the current paper maps in CMTs. However, further research and a study with actual CMT personnel will be needed to actually confirm these hypotheses.

With regard to the desktop GIS case, the ArcGIS software we used as a reference had major issues on the user experience side, especially regarding perspicuity. The relatively low scores in this regard can be attributed to multiple factors: first, ArcGIS is an expert system with a long development history, thus, the user interface is convoluted with options that only a fraction of the users actually need for their workflows. Second, some user interactions work different than in currently established consumer software (e.g. zooming with the mouse wheel works the opposite way it does current Web mapping software like Google Maps). This can easily limit performance for non-expert users, as it was the case in our study, and directly translates into a problem for CMTs, where GIS expertise is rare, and regular contact to any software for the crisis case cannot be assumed (crisis events are supposed to be the exception and thus relatively rare).

14.6 Conclusion

It has become consensus among researchers that interactive tabletops—or, more generally: large interactive surfaces—offer great potential for efficient co-located collaboration and coping with rich content (complex diagrams, multi-layer digital maps, etc.). However, while individual designs and interaction technique have been studied quite well, there is currently a lack of comparative studies evaluating the same tasks on tabletops and other systems [4].

To address this and contribute to a future body of research in this area, we have specifically investigated collaborative work in crisis management, which has been identified as a promising application domain for interactive tabletops [9, 16, 22, 23].

In crisis management, people from different public safety organizations (e.g. fire brigade, police, Red Cross) collaborate to establish a common understanding of the situation in the field, and plan and coordinate appropriate countermeasures. As geospatial information plays a crucial role in crises response scenarios, one of the central artifacts for this collaboration is a large paper map, with pins and paper symbols representing the current understanding of the situation in the field. What sets crisis management apart from other application areas are the very formalized, and very different, user roles and responsibilities; also, as there is no single chain of command, the current understanding of the situation may vary in completeness and recency between involved organizations. This necessitates regular information exchange and collaboration in planning to effectively address the crisis situation. Generalizing on this from an interaction perspective, we deal with a situation in which multiple users with access to different (role-specific) information sources and system functions have to collaborate to make sense of available geospatial data and conduct a series of planning tasks that corresponds to (hypothetical) actions in the real world.

Based on these characteristics, we have designed a system for interactive tabletops, that facilitates collaborative situation analysis and planning by users who have different information and planning functionality available. As part of our design, we have created an easy-to-implement technique for personalized touch interaction (including strategies for dealing with touch imprecisions and palm rejection), that works without additional hardware (assuming a user tracking system that can determine at which side of the table a user is located, as it is the case with the original coMAP [9] we based our work on system as well as our extension). We have also discussed how using ESRI ArcGIS technology as the basis of our system improves data integration and maintainability, and how we addressed a number of usability issues of the original coMAP system with new designs.

Using this software, we have then conducted a comparative study, in which 30 participants performed sense making/analysis and planning tasks reflecting actual crisis management work on the tabletop system, classical paper maps and an off-the-shelf desktop GIS. Our goal was to quantify the benefits of tabletops w.r.t. performance, usability, and teamwork quality. To our best knowledge, we were the first to actually compare all three types of systems in the same tasks, both in a crisis management inspired scenario, but also more generally in terms of a comparison of collaborative work with geospatial data on tabletops and other systems.

Our data revealed that participants were able to complete tasks significantly faster using the tabletop than with paper maps and the desktop GIS; also they perceived its user experience to be superior in most regards. Our results support the assumption that teamwork is comparable with tabletops and paper maps, with both offering better support than desktop GIS in this regard. Contrary to prior studies (e.g. [15]), we did not find desktop GIS to have a positive impact on actual or perceived efficiency.

Our study also revealed that the interaction techniques we designed for collaborative interaction in the named setting, i.e. personalized input, easy (selective) sharing, support for orientation, worked with little problems in a lab setting. However, further research will be required to allow participants a more flexible movement around the tabletop (currently the tracking system we use allows only one user per tabletop side).

In addition, we see two further directions for future work: first, more comparative studies of tabletops and other devices would allow verifying if the results we obtained in our study are related to the specific setting or type of information visualization (i.e., maps), or if they apply more generally. One candidate for this would be collaborative process modeling with interactive tabletops (or wall displays), desktop solutions, and classical brown paper with post-its. Second, a longer-term evaluation within actual CMT training sessions would be interesting, to see how well a tabletop solution performs in practice; this may however, among others, require more robust tabletop hardware (the Samsung SUR 40 suffers from spurious touch events when it is exposed directly to bright light sources or hold their wrists low while making touch contact with the screen).

14.7 Scenario/Task Example

Train Accident caused major fires at train station Frankfurt West.

Scenario Situation

A train that, among others, carried fuel has derailed (through a misconfiguration of the railway switch); the back part of the train was detached in this event, and collided with a building on the side of the tracks. Fuel has leaked and caught fire.

Situation Analysis and Data Synchronization

- Navigate to station *Frankfurt West* (in *Frankfurt Bockenheim*) and adapt the map so that you have an overview from *Hausen* to *Bockenheim*.
- Examine *Major Fire* damages from both organizations and compare your data. Add or share relevant data that your organization is missing (only in the immediate mission section).
- (*THW*) Examine *Train Accident* damages to locate the two parts of the train.
- Check out and mark areas where there is a risk for explosion hazards (in particular: *Petrol Stations*).
  - Explosion can occur when the *Major Fire* damage gets too close to a *Petrol Station*.
  - To mark an affected area, please circle it with a *Freehand Drawing*.

Measures Package 1 (Infrastructure)

- (*FB*) Place a *Contain Fire* measure between each *Petrol Station* and the *Major Fire* damage that could affect it, and send a *Fire Truck* to the location to prevent the risk of an explosion.
- (*THW*) Place a *Recover Vehicle* measure next to the train part that has not caught fire and send an *Infrastructure Unit* to recover it.

Measures Package 2 (Population)

- To avoid further personal damage, plan an *Evacuation* measure for the *Kindergarden* ("Kindergarten d. Evangelischen St. Jakobsgem.") and the *Kindergarden* ("Deutsch-Spanischer Kindergarten") to one or more of the *Emergency Shelters* close by.
  - Check the number of people you need to evacuate against the capacities of the emergency shelters before you decide where to evacuate.
  - Use *Freehand Drawing* to mark the route your unit is supposed to take. Be aware of barriers that cannot be passed like *Major Fire*-, *Impassable Route*- or *Train Crash*-damages.
  - Send one unit ((*FB*) *Fire Truck* or (*THW*) *Technical Platoon*) for each planned route to the coordinate the evacuation.

14.8 Modified Teamwork-Questionnaire (TWQ)

The following questionnaire is a condensed version of the Teamwork-Questionnaire (TWQ) presented in [12]. Modifications we applied were two-fold: first, questions that could be rephrased to reflect short-lived teamwork sessions were adapted accordingly; second: questions that could not be rephrased (while preserving their essence), or which can only be answered from teamwork over extended periods of time, were dropped. Questions were ranked on a 7-point Likert scale with the items on the edges labeled as “Strongly disagree” and “Strongly agree”. If uncertain, participants were asked to choose the center item.

1. Useful information/data was not provided (or could not be provided) to other team members in certain situations.
2. I was satisfied with the precision and granularity of the information/data we exchanged in our team.
3. If conflicts came up, they were easily and quickly resolved.
4. I was satisfied with the swiftness with which our team exchanged information/data.
5. The team closely made sure to align on/harmonize work done in subtasks.
6. Suggestions and contributions of the individual team members were valued in our team.
7. Our team was unable to reach consensus regarding important decisions/actions.
8. Relevant information/data was shared (openly) by all team members.
9. Conflicts/Problems occurred in our team, due to an imbalance in individual contributions (to the team’s goals).
10. There were factors which limited direct/personal communication within the team.
11. Discussions and controversies were conducted constructively.
12. All team members actively worked towards achieving the team’s goals.
13. In our team, the individual members helped and supported each other as best as they could.
14. In our team there were conflicts/obstacles limiting the information/data flow.
15. The team was clear about the specific possibilities of the individual team members.
16. I was satisfied with the usefulness of the information/data we exchanged in our team.
17. Every team member tried their best to contribute to the fulfillment of the team’s goals.
18. Subtasks defined within the team were clearly defined and fully comprehended.
19. All team members were contributing to the achievement of the team’s goals in accordance with their specific possibilities.
20. Our team had problems to agree on subtasks and their goals.
21. Suggestions and contributions of team members advanced the team’s progress.

References

1. Bader T, Meissner A, Tscherney R (2008) Digital map table with Fovea-Tablett®: Smart furniture for emergency operation centers. In: ISCRAM ’08: international conference on information systems for crisis response and management, pp 679–688
2. Ball R, North C (2005) Effects of tiled high-resolution display on basic visualization and navigation tasks. In: CHI ’05: extended abstracts of the SIGCHI conference on Human factors in computing systems. ACM, pp 1196–1199

3. Benko H, Wilson AD, Baudisch P (2006) Precise selection techniques for multi-touch screens. In: CHI '06: SIGCHI conference on human factors in computing systems. ACM, pp 1263–1272
4. Buisine S, Besacier G, Aoussat A, Vernier F (2012) How do interactive tabletop systems influence collaboration? Comput Hum Behav 28:49–59
5. Chokshi A, Seyed T, Marinho Rodrigues F, Maurer F (2014) ePlan Multi-Surface: a multi-surface environment for emergency response planning exercises. In: ITS '14: international conference on interactive tabletops and surfaces. ACM
6. Clayphan A, Collins A, Ackad C, Kummerfeld B, Kay J (2011) Firestorm: a brainstorming application for collaborative group work at tabletops. In: ITS '11: international conference on interactive tabletops and surfaces, pp 162–171
7. Dietz P, Leigh D (2001) DiamondTouch: a multi-user touch technology. In: UIST '01: ACM symposium on user interface software and technology. ACM, pp 219–226
8. Döweling S, Tahiri T, Schmidt B, Nolte A, Khalilbeigi M (2013) Collaborative business process modeling on interactive tabletops. In: ECIS '13: European conference on information systems. AIS, p Paper 29
9. Döweling S, Tahiri T, Sowinski P, Schmidt B, Khalilbeigi M (2013) Support for collaborative situation analysis and planning in crisis management teams using interactive tabletops. In: ITS '13: international conference on interactive tabletops and surfaces, pp 273–282
10. Hawkey K, Kellar M, Reilly D, Whalen T, Inkpen KM (2005) The proximity factor: impact of distance on co-located collaboration. In: GROUP '05: ACM international conference on supporting group work. ACM, pp 31–40
11. Hesselmann T, Flöring S, Schmitt M (2009) Stacked Half-Pie menus: navigating nested menus on interactive tabletops. Paper presented at the ITS '09: international conference on interactive tabletops and surfaces
12. Hoegl M, Gemuenden H (2001) Teamwork quality and the success of innovative projects: a theoretical concept and empirical evidence. Organ Sci 12:435–449
13. Holz C, Baudisch P (2011) Understanding touch. In: CHI '11: SIGCHI conference on human factors in computing systems, pp 2501–2510
14. Holz C, Baudisch P (2013) Fiberio: a touchscreen that senses fingerprints. UIST '13: ACM symposium on user interface software and technology. ACM
15. Johansson B, Trnka J, Granlund R (2007) The effect of geographical information systems on a collaborative command and control task. In: ISCRAM '07: international conference on information systems for crisis response and management, pp 191–200
16. Kunz A, Alavi A, Landgren J, Yantaç AE, Woźniak P, Sárosi Z, Fjeld M (2013) Tangible tabletops for emergency response: an exploratory study. In: MIDI '13: international conference on multimedia, interaction, design and innovation. ACM, p Article 10
17. Laugwitz B, Held T, Schrepp M (2008) Construction and evaluation of a user experience questionnaire. In: USAB '08: symposium of the workgroup human-computer interaction and usability engineering of the Austrian computer society. Springer, pp 63–76
18. Mackay WE (1999) Is paper safer? The role of paper flight strips in air traffic control. ACM Trans Comput-Hum Interact 6:311–340
19. Marquardt N, Kiemer J, Ledo D, Boring S, Greenberg S (2011) Designing user-, hand-, and handpart-aware tabletop interactions with the TouchID toolkit. In: ITS '11: international conference on interactive tabletops and surfaces. ACM, pp 21–30
20. Novak J, Aggeler M, Schwabe G (2008) Designing large-display workspaces for cooperative travel consultancy. In: CHI '08: extended abstracts of the SIGCHI conference on human factors in computing systems. ACM, pp 2877–2882
21. Novak J, Schmidt S (2009) When joy matters: the importance of hedonic stimulation in collocated collaboration with large-displays. In: INTERACT '09: Proceedings of the IFIP TC13 conference on human-computer interaction. Springer, pp 618–629
22. Paelke V, Nebe K, Geiger C (2012) Designing multi-modal map-based interfaces for disaster management. In: ACHI '12: international conference on advances in computer-human interaction, pp 95–100

23. Qin Y, Liu J, Wu C, Shi Y (2012) uEmergency: a collaborative system for emergency management on very large tabletop. ITS '12: international conference on interactive tabletops and surfaces. ACM
24. Rogers Y, Lim Y-K, Hazlewood WR, Marshall P (2009) Equal opportunities: do shareable interfaces promote more group participation than single users displays? Hum-Comput Interact 24:79–116
25. Rogers Y, Lindley S (2004) Collaborating around large interactive displays: which way is best to meet? Interact Comput 16:1133–1152
26. Scott SD, Allavena A, Cerar K, Franck G, Hazen M, Shuter T, Colliver C (2010) Investigating tabletop interfaces to support collaborative decision-making in maritime operations. In: ICCRTS '10: international command and control research and technology symposium, pp 1–15
27. Scott SD, Carpendale S, Inkpen KM (2004) Territoriality in collaborative tabletop workspaces. In: CSCW '04: ACM conference on computer supported cooperative work. ACM, pp 294–303
28. Venkatesh V, Davis FD (2000) A theoretical extension of the technology acceptance model: four longitudinal field studies. Manage Sci 46:186–204
29. Wang F, Cao X, Ren X, Irani P (2009) Detecting and leveraging finger orientation for interaction with direct-touch surfaces. In: UIST '09: ACM symposium on user interface software and technology. ACM, pp 23–32
30. Yost B, Haciahmetoglu Y, North C (2007) Beyond visual acuity: the perceptual scalability of information visualizations for large displays. In: CHI '07: SIGCHI conference on human factors in computing systems. ACM, pp 101–110
31. Zhang H, Yang X-D, Ens B, Liang H-N, Boulanger P, Irani P (2012) See me, see you: a lightweight method for discriminating user touches on tabletop displays. In: CHI '12: SIGCHI conference on human factors in computing systems, pp 2327–2336

Chapter 15
Envisioning the Emergency Operations Centre of the Future

Edwin Chan, Craig Anslow, Teddy Seyed and Frank Maurer

Abstract Emergencies, crises, and disasters happen frequently, with significant impact on the lives of countless people. To respond to these events, many organizations including the Police, EMS, and Fire departments work together in a collaborative effort to mitigate the effects of these events. In addition, these agencies are often joined by third-party organizations such as the Red Cross or utility companies. For all of these groups to work together, an Emergency Operations Centre (EOC) acts as a hub for centralized communication and planning. Despite the significant role of the EOC, many existing EOCs still rely on aging technologies, leaving many potential improvements available by adopting new technologies. Considering the impact of emergencies on human lives and lost resources, and the scale of these emergencies, even a minor improvement can lead to significant benefits and cost-savings. Emergency Operations Centre of the Future (EOC-F) is an exploration into the integration of various novel technologies in EOC design, in an effort to make emergency response more efficient and collaborative. We have built a multi-surface environment (MSE) which utilizes various digital surfaces including display walls, tabletops, tablet devices, and mobile/wearable computing devices. Within this multi-surface EOC, we look at proxemic interactions and augmented reality as useful ways to transfer and access information. We also discuss how analysis of information gathered within the EOC, as well as social media, can lead to more informed decision making during emergency response.

E. Chan (✉) · C. Anslow · T. Seyed · F. Maurer (✉)
University of Calgary, ICT 602, 2500 University Drive NW,
T2N 1N4 Calgary, AB, Canada
e-mail: chane@ucalgary.ca

F. Maurer
e-mail: fmaurer@ucalgary.ca

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_15

15.1 Introduction

Emergencies, crises, and disasters happen when people least expect them to. Some notable examples include: earthquakes in Christchurch, New Zealand (2011), tsunamis in Japan (2011), flooding in Southern Alberta, Canada (2013), and a missing plane in Malaysia (2014). To respond to these disasters, organizations such as the Fire department, Police Department, EMS and others work together to discuss and plan within a co-located emergency operations center (EOC) (Fig. 15.1). Crisis management teams that meet face to face in emergency situations also exist in major corporations and public organizations. These teams have specific needs but their information system support can be very limited. Despite accommodating various teams within the same space, existing EOCs provide few supporting tools to encourage collaboration between the teams. There is a significant opportunity to utilize new technologies to address these concerns, while providing a more effective response to emergencies.

Beside reductions to lost lives and injuries, an improved response has also a substantial cost savings potential, both for the public sector as well as for industry. The costs of the Southern Alberta floods in 2013, earthquakes in Christchurch, and tsunamis in Japan are estimated to be \$5 billion, \$18 billion, and \$35 billion USD respectively. While the cost reductions coming from a more effective handling of the crisis are hard to estimate, even small percentage gains can potentially lead to large savings.

Fig. 15.1 Calgary Emergency Management Agency (CEMA). The City of Calgary, 2016

The Emergency Operations Centre of the Future (EOC-F) is a collaborative project between the University of Calgary and C4i Consultants to explore and prototype emergency operation planning and operation tools. The goal of this project is to investigate how analytics-based, spatially-aware multi-surface environments (MSE) can support teams managing emergencies in an EOC. Perhaps the greatest challenge for any decision-making entity is the ability to efficiently gather, process, and visualize information in a timely manner, allowing the best decisions to be made as early as possible. By investigating local emergency agencies and their EOCs, we have identified many opportunities for improving emergency response operations including: inter-organization interoperability, communication within the EOC, communication between the EOC and field responders, and situational awareness of field responders. In addition, we look at how social media analytics can be harnessed as a valuable source of citizen-based, on-the-ground information, without creating significant overhead for EOC operators.

We prototype and qualitatively evaluate an EOC design which improves on existing solutions by making use of new technologies to address the problems identified above. The system was built on design principles derived from both existing research and the constant feedback of emergency personnel, discussed subsequently. We then present a usage scenario for the new system to demonstrate the potential, and compare to existing systems. We conclude with some of our results, and discuss some of our continuing work for EOC-F.

15.2 Background

15.2.1 Disaster Management

Disasters occur on a daily basis, on various scales, and emergency services can receive thousands of calls per day [1, 11]. While some situations can be resolved with relatively few resources, emergencies often require the cooperation of multiple agencies, often involving personnel whose primary job responsibility is *not* emergency management. For example, a fire in a populated downtown area may require the police to manage civilian access and evacuations, while firefighters focus on controlling the fire. EMS may be on-site to treat injuries, while providing support to the firefighters operating in a hazardous environment. Utility companies collaborate with these agencies to assess and reduce potential dangers, such as damaged gas pipes or electric wires. When an emergency situation becomes prolonged, it is common for the involved parties to establish a shared headquarters, the emergency operations center (EOC), to facilitate information sharing, resource management, and operations planning. The EOC becomes the central command and control facility, interacting with other entities such as the media as well as NGOs like the Red Cross. Given its significant role in emergency management, it becomes apparent that an improved EOC will benefit every aspect of emergency response.

15.2.2 Emergency Operations Centers

An underlying problem with existing EOC solutions is the lack of built-in support for collaboration within and between teams, often from multiple organizations [27]. While EOCs often have designated areas and computing terminals for various organizations such as the police or EMS, support for inter-organization collaboration usually means having enough space to physical accommodate the various teams. In many cases, members of an EOC work independently at their own terminals, with few tools to encourage collaboration. An example of this is the Calgary Emergency Management Agency (CEMA) which can be described as somewhat disorderly, despite being touted for its rapid response during the Southern Alberta Floods in 2013 [53]. Operators were seated at individual cubicles, and simply shouted out any emerging needs to other operators, with corresponding parties shouting back. It was likened by the mayor of Calgary to a game of Whack-A-Mole.

With a lack of connectivity between members of the EOC as well as responders in the field, it becomes apparent how information transfer can be slow, inaccurate, and often very repetitive. Imagine an operator receiving a call from a field responder, with information about an ongoing event. The operator would manually record information such as the location of the event, while making notes of the event. The operator would then have to manually locate relevant parties, such as the incident commander of the EOC and members of relevant agencies. The information would have to be repeated to each person perhaps separately, a very time-consuming process. Assuming a decision was made for backup to be dispatched, the response team in the field would also have to receive the same information manually, before finally heading to the location. An improved response through greater connectivity could allow the operator to digitally retrieve the location of the caller, record any event details, and digitally distribute the information to relevant parties. The resulting decision would be automatically forwarded to the response team, along with any relevant information about the event. Because location data is sent digitally, the response team can easily enable navigation without having to manually enter the address. The movements of the response team can then be tracked live in the EOC, again through an automated process.

With so many apparent benefits to greater connectivity and automation within EOCs, it seems logical that these tools should be integrated into all EOCs. However, this is not currently the case, because many commercial solutions target specific roles within an EOC, while others provide only a part of the EOC [17]. It is not uncommon for multiple vendors to supply various parts of an EOC, each with its own software suite. For example, WebEOC supports incident management and information access on individual terminals, but lacks integration with large display surfaces used for face-to-face collaboration between multiple users [49]. Other parts of the EOC remain disconnected, and any potential integration with existing or future components can be costly.

Despite difficulties in developing and maintaining a fully interconnected EOC, the benefits are nontrivial and it is worthwhile to examine the possibilities of such a system. Beyond the immediate advantages leveraged during an incident, a connected system can also benefit the response preparation and training phases. Events can be logged from all the connected devices, detailing when information was received, the people involved, and the decisions made during emergency management. These records can prove very useful when reviewing an incident, revisiting all the captured events leading up to any actions taken. The data can then be further analyzed for improving future incident responses, and reused for training purposes.

In recent years, solutions which promoted better connectivity and integration between various teams have been deployed to great results. IBM's Intelligent Operations Center was deployed in response to Typhoon Haiyan to coordinate numerous distributed teams, while the NYPD's Domain Awareness System has become the leading example of how large-scale deployment of connected devices can empower EOCs. Similarly, we believe the connectivity and collaborative spaces of multi-surface environments can be applied to EOC design to great effect.

15.3 Related Work

15.3.1 Designing an Emergency Operations Center

Although most relevant works focus on specific technologies within the emergency response domain, some recent studies have started looking at the EOC as a whole, discussing the role of technology in relation to many aspects of the EOC. An influential report published by the European Commission discusses several important topics, including: the need for an EOC to support multiple devices, the functions of a large wall-sized display, the physical design of an EOC to support social media analysis, and individual "lenses" to facilitate independent interactions within collaborative interactions [10, 20, 23]. Their findings helped guide us toward our current design of EOC-F, with respect to the inclusion and placement of devices, the role of large displays, and personalized devices for users.

Another recent study into collaborative work in disaster response stressed the volatile nature of disasters, and the need for an EOC which can handle four types of uncertainties. Uncertainty in the environment and in equipment available pushed us towards a modular design, based around "redundancy and graceful degradation" [19]. For EOC-F, a modular design not only means the potential to scale up the system, but also the ability to function with minimal pieces of the EOC-F. One device can have multiple configurations, allowing it to perform several roles within the EOC. Devices such as a digital whiteboard can be written on like a regular whiteboard to help conduct planning, and if a connection is available, it can also share the hand-written information with other devices. The third uncertainty is that of available data, such as satellite imagery, local maps, and population data. Fischer et al. describe a design which provides "flexible support for situation analysis", suggesting the need for incorporating various streams of information in an adhoc manner. When collecting information from varying sources, we run into the fourth uncertainty regarding the origin and integrity of information. Articulating and accounting for this uncertainty is important for making informed decisions, affecting our design for social media integration.

Previous research by others, supported by repeated user consultations with many emergency management organizations in various roles, have led us to develop EOC-F, a multi-surface environment for emergency response.

15.3.2 Multi-surface Environments

A multi-surface environment (MSE) is a room where multiple computational devices are located and potentially communicate together. MSEs may contain any combination of phones, tablets, laptops, digital tabletops, projectors and large high-resolution display walls for various domain specific applications. An example of one of the earlier MSEs is the Wild Room [5].

MSEs offer rich opportunities for new applications, interactions, and collaboration. Creating these environments is difficult and integrating traditional software is a challenge for the design of MSE applications [18]. While some researchers have explored emergency management applications on individual devices, we are unaware of any research that has used and evaluated MSEs for emergency management purposes. For this reason, we consider our research project innovative, with potential for significant contribution to the scientific field as well as high commercial potential.

Our eGrid system was a prototype application utilizing a digital tabletop for utility companies to enable collaboration of control center team members in their daily tasks of analyzing and managing the electrical grids of a city [43] as well as dealing with outages. The application uses a multi-touch table that allows multiple users to interact concurrently with domain specific Geographic Information Systems (GIS) data. However, the application does not consider the specific context of emergency management nor the necessary integration of other devices within a team.

Another MSE application is coMap, an interactive tabletop application that facilitates collaborative situation analysis and planning in crisis management teams [16]. Users can interact with coMAP using multi-touch as well as pen input with Anoto digital pens directly on the table. Others that have also explored Anoto digital pens on tabletops (for air traffic control rooms) found that the input was problematic and using the digital pens required specially designed proprietary paper [42]—something quite limiting during an emergency situation.

CERMIT uses light emitting tangible devices and mobile phones to interact with a tabletop for emergency management [37]. CoTracker is an application that uses tangible graspable objects on a tabletop for emergency management [2, 26]. μEmergency is a multi-user collaborative application for emergency management on very large-scale, interactive tabletops which allows people to carry out face-to-face communication based on a horizontal global map and uses tangible objects placed on the table for input [38]. However, none of these applications have been integrated into a larger MSE nor did they integrate with commercial emergency planning and operations software.

CodeSpace is an application that used phones, laptops, and a vertical touch display to support collaboration in meetings targeted at software development [7]. The application allows information to be shared across devices but does not support different roles, which are necessary in an EOC.

The Sky Hunter system is an application that uses a tabletop and iPads to display geospatial data, which allows a heterogeneous group of analysts to explore possible locations for oil and gas exploration [46]. The application allows geospatial information to be shared between devices, but the application is limited to a small digital table and a single iPad.

The MRI Table Kinect is an application for visualizing volumetric data such as CT and MRI imagery that uses an iPad and a tabletop [44]. The application supports slicing the volumetric data by moving an iPad or hand in the physical space above the table to explore the data in more detail which is displayed either on the iPad or another large screen. The approach can be utilized in an EOC to interact with volumetric geospatial data (e.g. 3D models of buildings or streetscapes).

15.3.3 Multi-surface Environment Toolkits

Creating applications that support multiple devices in MSEs is challenging, as it requires development for different form factors and platforms. Ideally, one application could be deployed to many different devices; however, this usually limits the user experience on each of the devices and also has yet to be applied to an EOC situation. Paterno et al. present a framework for describing various design dimensions that can help in better understanding the features provided by tools and applications for multi-device environments [36]. jQMultiTouch is a lightweight web toolkit and development framework for multi-touch interfaces that is designed to perform on many different devices and platforms [33]. XDStudio is an attempt to support interactive development of cross-device web interfaces, which has two modes [34]. In the simulated mode, one device is used as the central authoring device, while target devices are simulated. In the on device mode, the design process is also controlled by a main device, but directly involves target devices. XDKinect is a lightweight framework that facilitates development of cross-device applications using Kinect to facilitate user interactions [35]. None of these applications have been integrated into a MSE for emergency management.

Our Multi-Surface Environment API (MSEAPI) was developed for sharing information amongst devices, using proxemics and gestural interactions [3, 9]. Using Microsoft Kinect cameras, the system can detect and track where people and devices are located in the environment. This spatial awareness allows simple proxemic interactions to be used in information transfer between users and devices. For example, a user can point a tablet at another user in the room, and simply flick on the screen towards the other user. The latter will then receive the information on their device, making digital content sharing as natural as passing a physical document around. One of the projects which made use of MSEAPI was ePlan, a software tool for simulating large scale emergencies to train civic operators on responding to different emergencies [13].

Our Society of Devices Toolkit (SoD Toolkit) is the successor to MSEAPI, and supports more proxemic interactions compared to MSEAPI. This new toolkit creates spatially-aware environments that are modular and easily extendable with new devices and can be spread over multiple rooms. As a result, projects such as EOC-F which rely on the toolkit are also modular and can be scaled for different scenarios. The SoD Toolkit integrates additional sensors and devices to provide greater environmental awareness. Several new additions include the LEAP sensor, iBeacon, and Google Tango. Beyond integrating data streams from each of these devices, the SoD Toolkit makes sense of this information and affords higher-level interactions between connected devices. The toolkit also provides APIs for multiple platforms, making it possible to integrate new sensors and devices as they become available. The extensibility of the SoD Toolkit makes it suitable for supporting EOC-F by providing ease of integration of new technologies. Novel proxemics and gestural recognition make interactions in EOC-F natural and intuitive.

15.3.4 Gesturing in a Multi-surface Environment

Determining what gestural interactions are suitable for multi-surface environments (MSEs) is an open research question. Various researchers have explored interactions for visualization walls, tabletops, and the combination of many devices in a MSE. However, gesture preferences are specific to different scenarios and use cases, and gestural interactions within an EOC remain untested. Designing interactions appropriate for applications in MSE EOCs is one of the important research challenges that our team is addressing.

Nancel et al. conducted a user study of mid-air interactions on a large visualization wall [32]. They studied different families of location independent, mid-air input techniques for pan-zoom navigation on wall-sized displays. They also identified key factors for the design of such techniques: handedness (uni vs. bimanual input), gesture type (linear or circular movements), and level of guidance to accomplish the gestures in mid-air (movements restricted to a 1D path, a 2D surface or free movement in 3D space).

Wobbrock et al. conducted a user study with 20 participants to explore what gestures would be appropriate for a tabletop [54]. Participants performed a total of 1080 gestures for 27 commands with one and two hands, which resulted in a user-defined set of gestures. The findings showed that participants rarely cared about the number of fingers used in a gesture, one hand was preferred over two, and desktop idioms strongly influenced how users came up with gestures.

Seyed et al. conducted a Wizard of Oz user study to elicit gestures in a multi-surface environment using an iPad, tabletop, and wall display [45]. Participants performed a total of 816 gestures for 16 commands. Initial designs of gestures and peripheral interactions in MSEs have been proposed for pulling content from another device, pouring content from a tablet onto a tabletop, and sending content through flick gestures [13, 39]. However, these gestures and possible alternatives have not been empirically evaluated with EOC personnel. The resulting set of gestures likely does not cover all the tasks performed in an EOC. Considering the relative infancy of MSE research and the growing popularity of MSEs, further evaluations of gestures and other interactions within specific domains will be necessary.

15.3.5 Proxemic Interactions

Proxemic interactions are another type of interaction enabled by spatially-aware MSEs. Like gesture interactions, proxemics allow users to perform intuitive actions that are natural to them. Existing research on proxemics interactions examine how users perceive their relative positions to other people and devices, and how this perception can facilitate different actions. Hall [21] defined proxemic zones surrounding a person, including intimate distance, personal distance, social distance, and public distance. Vogel and Balakrishnan [52] explored proxemics in relation to public ambient displays, while Ballendat et al. [4] used sensors to track people and devices within a ubiquitous environment. Marquardt et al. [31] looked at spatial relationships within ubiquitous environments, specifically focusing on five proxemics dimensions: orientation, distance, motion, identity, and location. The combination of gestures and spatial awareness have resulted in natural actions for content transfer, including: simulating a throwing action, flicking towards another device, or pouring content from one device to another [9, 15]. The intuitive nature of these actions allow users to easily learn and adopt an unfamiliar system, a procedure which is often encountered in EOCs when new personal needs to be quickly integrated during emergencies to create a coordinated response.

15.3.6 Sense-Making, Visual Analytics, and Social Media

Although EOCs already aggregate various information streams from multiple agencies, a significant amount of information can be harnessed from the public. While it takes time for first responders to arrive on the scene of an incident, citizens on the ground are often able to provide critical information via social media as an incident unfolds, making this information extremely valuable. Making effective use of this information can reduce resource costs for deployment, but the flood of information can be overwhelming to process as well. Sense-making and visual analytics can help extract critical pieces of the information, in a timely manner essential to emergency response.

Sense-making is the process of searching for a representation and encoding data in that representation to answer task-specific questions [40]. Different operations during sense-making require different cognitive and external resources. Representations are chosen and changed to reduce the cost of operations in an information-processing task.

Visual analytics builds upon sense-making and is the science of analytical reasoning facilitated by a visual interactive interface and the use of information visualization techniques [14]. Visual analytics can attack certain problems whose size, complexity, and need for closely coupled human and machine analysis may make them otherwise intractable.

A number of researchers have explored using visual analytics and information visualization techniques for emergency management [24, 28, 55, 56] and understanding social media in the context of emergency response and crisis scenarios for earthquakes, fires and floods [41, 47, 51]. However, we are unaware of any published results exploring visual analytics and social media for emergency response management integrated with MSEs.

15.3.7 Wearable Computing

Although significant research has focused on extracting information from public media sources, communications with first responders remains mostly unchanged with many emergency agencies still using VHF or UHF radio [8]. This is very interesting, when we consider the increasing capabilities of mobile devices to capture and communicate much more information than radios. Despite mobile devices reducing in both size and cost, they have been unable to replace the radio as the primary tool for information transfer during an emergency. Several factors contribute to this dilemma, including a greater learning curve for responders, lack of resources to process the additional information, and an additional burden on responders to make use of the device.

A recent trend in mobile computing with body-worn devices may finally be able to overcome these problems. Wearable computing is the study of designing, building, or using miniature body-borne computational and sensory devices [6]. Wearable computers may be worn under, over, or in clothing, or may also be themselves clothes. Although wearable computers have only become popular among consumers recently, the idea itself has existed for much longer.

As early as 1994, 1996, a “wearable computer system equipped with head-mounted display, camera(s), and wireless communications” called WearCam already existed as an early precursor to existing wearable computers [29]. Early exploration of wearable computing for emergency response involved firefighters playing a simulated game with a gas mask [25], but the prototype was not evaluated within a real scenario. Cernea et al. developed their own wearable device for firefighters to use on their forearms [12]. However, the wearable unit was considered too big and heavy to be successfully employed in real rescue operations. This rather common limitation is beginning to disappear, with the advent of smaller and more powerful devices.

An important aspect of modern wearable computers is the number of sensors embedded into them, constantly collecting information about the wearer and their surroundings. Through these sensors, EOC operators can easily discern the status of field responders including their safety, location, and movements such as chasing after a suspect. In addition, responders can send back visual information through body-mounted cameras, while information from the EOC can be easily visualized by the responders. While visual and location information cannot be communicated effectively over radio, an address can be directly visualized on a head-mounted display (HMD) in a map, with navigation support for the responder. More recently, Google's Project Tango enabled augmented reality in the form of mobile phones and tablets [48], while Microsoft's HoloLens combined augmented reality with head-mounted displays [22]. Using augmented reality, information can be overlaid on real-world objects to further improve how we display and interact with information. With so many new channels of information transfer, communications and situational awareness can be improved over existing methods.

In addition to using wearable computing in the field, we also see opportunities for these technologies within the EOC. As far as we are aware, there has been no integration of modern wearable computing devices (such as Google Glass) into MSEs for domain-specific applications such as emergency response.

15.4 Requirements Gathering

To ensure our system was designed with users in mind, we consulted local emergency response agencies through multiple stages of our design. This was done in collaboration with our industry partner C4i Consultants, who specialize in training software for emergency response and military operations. More recent consultations include extensive requirements gathering with local firefighters, police officers, emergency management officers, industry groups, and research groups, conducted over 3 months. An emergency management workshop was then held in Banff at Cyber Summit 2015, featuring a demo of EOC-F. An open-house was then hosted at the University of Calgary, with over 60 professionals participating over two days. Subsequent interviews were held with the Calgary Police Service, focusing on communication and information needs for first responders.

A recurring theme was the desire for organizations to protect their responders, by improving communication channels and increasing situational awareness of both responders and EOC operators. Within this theme, we grouped the requirements into three categories: (1) field responder status, (2) location and navigation, and (3) communications and media support. First and foremost was the health and wellbeing of first responders. Beyond ensuring responders were alive, EOC operators wanted to know if responders were experiencing physical or mental fatigue. Next, EOC operators wanted to know the locations of responders at all times, with the ability to navigate them to points of interest including the locations of fellow responders. Finally, communications between the EOC and responders needed to be bi-directional and capable of transmitting different media formats such as photos or videos. The last category corresponds with literature from Toups et al. [50], which describes the challenges of one-way communication and the dangers associated with poor situational awareness.

15.5 Next Generation EOC

As previously stated, the Emergency Operations Centre of the Future (EOC-F) is an investigation into how analytics-based, spatially-aware multi-surface environments (MSE) can support teams managing emergencies in an Emergency Operations Centre (EOC) (Fig. 15.2). Emergencies are often unique, and an EOC has to handle a vast variety of scenarios. Similarly, EOCs can range from small localized teams to much larger collaborative efforts, situated in dedicated buildings or deployed as a mobile response. EOC-F is designed to be both mobile and scalable, so that it can be adapted and deployed in various situations even when faced with many uncertainties.

Fig. 15.2 EOC-F display wall, tabletop, and tablets

15.5.1 Technology

To support the numerous roles and activities within an EOC and the field responders interacting with the EOC, a comprehensive range of devices are included in EOC-F. Within EOC-F, collaboration planning is done around one or more interactive tabletops, while large wall-sized displays provide a common operating picture for the entire EOC. A digital whiteboard provides more traditional means of planning, but allows handwritten notes to be digitally distributed to other devices such as the tabletops. Operators carry tablet devices which facilitate planning with smaller groups or independently. Proxemic interactions between the various devices are enabled either by placing cameras within the EOC, or by using spatially-aware tablets. In the field, first responders are equipped with mobile phones or wearable devices to connect them to the EOC. Here we present details of each device in EOC-F (Table 15.1), while the following sections describe various use cases for EOC-F.

15.5.2 Spatial Awareness, Proxemics Interactions

While further analysis of interactions and gestures for these environments is required, EOC-F currently supports several novel interactions in addition to being a spatially-aware system. The interactions are part of the prototyping process and will be the basis for subsequent usability studies.

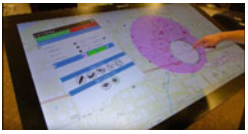
The two basic gestures are *flick* and *pour* (Fig. 15.3). The *flick* gesture can be performed on a tablet device by holding and swiping either towards or away from the user. Since the system is spatially-aware, the user can point their tablet at another device (wall display, tabletop, or tablet) and perform a *flick* gesture to send information to that device. For example, *flicking* across the room towards the tabletop will allow the tabletop to display the same location on the map as the tablet.

The *pour* gesture can be done by positioning a tablet above a tabletop, and flipping the tablet over as if to pour the contents of the screen onto the tabletop. This gesture can be used to share information from the tablet to the tabletop, essentially making the information public to the EOC. For example, a response plan drawn up by the police is initially only visible to the police, but can be shared with other organizations by *pouring* the plan onto the tabletop.

These gestures rely on the locations and orientations of people and devices within the MSE, and are provided by the SoD Toolkit and its sensors. In addition to gesture recognition, proxemics allows natural interactions to take place. One such use is the control of the wall display through a tablet. A user can walk up to the wall display, and are then able to modify what information is displayed on it. Another example is the sharing of information to everyone around you based on proximity, rather than having to individually share information one at a time during a group discussion.

Table 15.1 Technology components of EOC-F

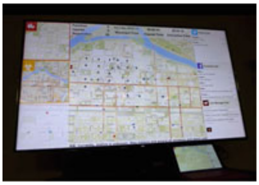
Tabletop



The Microsoft Perceptive Pixel is a 55” touch-enabled surface which supports collaborative planning around the table. It replaces traditional paper maps, providing a number of tools to improve the planning process. Some features include:

- live location and status updates of field responders
- annotation tools to draw up plans
- multilayer support so multiple plans can be considered
- route planning for responders, with automatic notifications sent to the field

Display wall



The Visualization Studio at the University of Calgary measures 4.88 metres by 1.85 metres (195 inches by 73 inches), and has a resolution of 9600 x 3600 pixels. It is used to provide a common operating picture to the entire EOC, and contains the following information:

- general information regarding ongoing incidents (eg. elapsed time, incident status, alerts)
- a large map synchronized to the view on the tabletop; increases situational awareness of all EOC operators, and can be used to present incident updates or response plans

Digital whiteboard

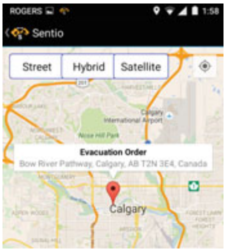
The SMART kapp® board is a digital whiteboard which bridges the familiarity of traditional planning tools with the connectivity of multi-surface environments. Notes and plans can be handwritten with regular dry-erase markers. Once completed, the contents can be digitally distributed to other devices such as the tabletop, wall display, or even view in the field on mobile devices.

Tablet

Microsoft Surface Pro 3 tablets act as portable planning devices, providing similar tools to the tabletop. Tablets are role-specific, and provide tailored tools for different roles. For example, evacuation and roadblock tools may be exclusive to the police. The tablets can be used for drawing up plans either independently or with a small group, before being shared to the EOC via the tabletop or wall display. Information can be shared simply by pointing the device to another surface, and swiping the content in that direction. Such proxemic interactions make content sharing intuitive, and reduce the learning curve of users not familiar with the EOC (eg. NGOs or volunteers).

In a mobile or impromptu EOC where large displays and tabletops are not available, tablet devices can be substituted to simulate a tabletop or shared display.

Mobile phone

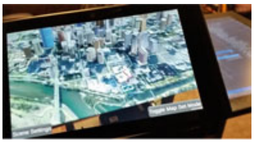


Although tablets can be deployed in the field, most responders do not carry a tablet device. However, mobile phones have become ubiquitous, with many responders carrying both a personal device and a work-issued device. These devices can be used as an extension of the multi-surface environment within the EOC, providing greater situational awareness to both the responder and the EOC. The EOC can track the location of responders through GPS embedded within the devices, while responders can receive notifications from the EOC. For example, an operator in the EOC can create an evacuation zone on the tabletop, with automatic notifications sent to all affected responders.

Wearable devices

Although not as common as mobile phones, wearable devices have become more popular in recent years, with many fitness bands, smart watches, and smart eyewear available commercially. One such device is the Recon Jet, a pair of sunglasses integrated with a video camera, GPS sensor, and a small LCD display. In EOC-F, geotagged photos and videos can be sent back to the EOC, and be directly displayed on the tabletop or wall display maps. Notifications from the EOC are displayed in the heads-up-display (HUD), and dispatch orders can be visualized on a map with navigation support. All this is done hands-free, allowing responders to focus on ongoing tasks.

Augmented reality



Using devices with depth sensing capability such as Google's Project Tango tablet, a device can become spatially-aware of its surroundings. Using this awareness, the devices can display 3D visualizations in augmented reality, creating an immersive planning environment. For example, virtual 3D models of buildings can be placed on the tabletop map, providing greater context to EOC operators. By moving the tablet through the visualizations of buildings, floor plans can be viewed as well.

Spatial awareness

Spatial awareness enables proxemic interactions within a multi-surface environment, by tracking the locations and orientations of people and devices. EOC-F uses the SoD Toolkit to make sense of this information, so that actions such as flick or pour can be used to transfer information intuitively. To do the tracking, Microsoft Kinect depth-sensing cameras are placed within the environment. Alternatively, spatially-aware devices such as those used for augmented reality in EOC-F can also be used to provide spatial tracking.

Fig. 15.3 Transferring content through proxemic interactions: Flick (*left*) and Pour (*right*)

15.5.3 Social Media Analytics

With the proliferation of networked mobile devices, the public have become a vital source of real-time information during an emergency crisis. The public is often able to alert officials to immediate problems, well before any responders arrive on scene. They are a low-cost source of information, and provide a lot of information for relatively little resources spent. However, the sheer amount of information received becomes problematic when emergencies allow little time for careful and thorough analysis. This is where social media analytics can help extract the most relevant information, reducing the burden on EOC operators and improving the overall response effort.

In EOC-F, social media streams such as Facebook and Twitter are displayed on the wall display. To ensure only useful information is displayed to the EOC, a Social Media Analyst filters through the vast amount of updates before they reach the rest of the EOC [30]. Social media analytics help automate and accelerate this process, through criteria filters such as keywords, timestamps, or geolocation data.

Interactions with 3D Geospatial Data in MSE
The integration of Google's Project Tango allows cutting-edge interactions with geospatial data, most notably 3D models in augmented reality (AR). Building models are overlaid on the tabletop map or can be positioned as virtual matter inside the EOC space, allowing much more information to be visualized in the same space (Fig. 15.4). With 3D models as virtual matter, personnel are able to interact with the same objects from different perspectives while the system maps their devices' coordinate systems to the physical space of the room. Operators can also access building plans by viewing cross-sections of these models. Rather than losing track of personnel once they enter a building, operators can closely follow their movements and support them with detailed directions.

The virtual models are placed on the table just like a real object, meaning multiple operators using their own devices will see the same models. For example, if someone points to a particular building, another user with their own device will see the same building being pointed to (Fig. 15.5). This shared space allows collaborative planning to continue beyond the 2D planning table. Both the technology and its integration are still in their early stages, providing a great opportunity for

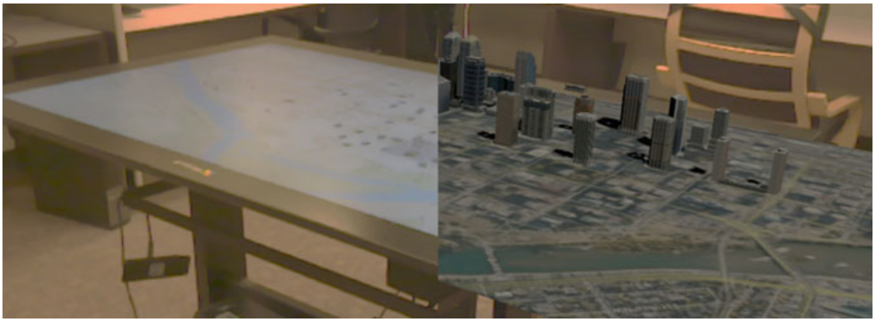


Fig. 15.4 Planning table with map displayed (*left*), and same table with 3D model overlaid using augmented reality (*right*)

Fig. 15.5 A user is pointing to a building, seen through his own device (*top*). The same user is visible through another user's device, pointing to the same location on the map (*bottom*)

future work to investigate interaction techniques with virtual matter in an EOC. These interactions can also be used in other applications, including remote collaborative planning or field-use by on-site responders.

15.5.4 Sample Configurations

In a permanent and stationary EOC, larger devices such as the display wall and tabletop may already be in place for regular use. For the integration of external agencies such as the Red Cross or utility companies, their representatives may be provided with tablet devices to quickly join in the planning process. Field responder from the EOC may already be equipped with wearable devices, while additional responders may be provided with such devices if available. If this is not possible, responders can access the same tools from the wearable devices using a mobile phone.

Other than permanently situated EOCs, there are many scenarios which use rapidly deployed ad hoc EOCs. Mobile Command Vehicles such as those used by the US Department of Homeland Security are designed for mobile deployment, limiting the available equipment. Temporary EOCs may also be setup in close proximity to incidents, and are often located in public buildings such as community centres. In these types of deployments, many of the previously mentioned uncertainties [19] can have a significant effect on the EOC setup. To account for these variables, EOC-F can operate with minimal equipment, with many components remaining partially operable even in suboptimal environments. In a minimalistic setup, a single tablet can replace the tabletop for collaborative planning.

15.5.5 Usage Scenario

EOC-F is a multi-surface environment (MSE) formed by a combination of the numerous components described above. To better illustrate the use of the various surfaces and spaces within the MSE for emergency response, we walk through a potential scenario involving a train derailment.

A train has derailed in downtown Calgary, where the tracks intersect with several high-traffic areas. Nearby responders from the Police, EMS, and Fire departments are already en route. As the EOC operators prepare to respond, representatives from various organizations, including the government as well as the railroad company, gather in a meeting room to assess the situation. Key tasks are handwritten on a digital whiteboard, and the contents are digitally transferred to the EOC tabletop and mobile devices in the field.

At this point, EOC operators are collecting information from various sources. Photos taken by on-scene responders automatically show up on the wall display map. A social media analyst monitors and filters through social media feeds such as Twitter, pushing relevant information to a feed on the wall display. Information begins to aggregate on the wall display, providing a common operating picture for everyone in the EOC. From the gathered information, it is determined that the derailment has resulted in a large chemical spill. A hazmat team in the field assesses the risk and draws up evacuation areas on a tablet device, sharing the information with the EOC. In response, the police create plans for roadblocks and detours around the incident, then shares the plans with the other agencies by *pouring* their plans onto the tabletop.

To evacuate citizens from buildings near the chemical spill, the fire department needs to know more about the buildings in the area. Using augmented reality enabled tablets, firefighters both in the EOC and on-scene are able to see a 3D virtual model of the incident area. They can determine the height of the buildings, which affects the time required to complete the evacuation. By moving the tablet through the virtual model, responders can easily look inside the buildings to access floorplans.

After considering several plans on the tabletop, a response plan is finalized. GPS sensors in mobile devices carried by field responders allow EOC operators to see where responders are located, directly on the tabletop map. By creating an exclusion zone on the tabletop, all responders within the area are automatically notified of the evacuation and provided with directions for the shortest path out of the area. To direct reinforcements to the incident location, responders are selected on the tabletop and a destination is set on the map. The destination, shortest paths, and ETAs are automatically calculated and sent to responders' mobile devices. Rather than having to ask and confirm addresses or directions over radio, responders can immediately head to the incident by starting navigation on the mobile devices. Because the locations of all responders are known, on-scene responders are aware of when reinforcements will arrive.

During the evacuation and spill containment, responders are working hard to ensure the safety of the citizens. However, it is equally important to keep responders safe, as they encounter unexpected and often dangerous situations during an emergency. Pairing responders' mobile devices with wearable health sensors can help the EOC monitor the safety of responders. If a responder's heartbeat or movement becomes irregular, EOC operators are automatically notified of the discrepancy so that help can be provided if needed.

15.5.6 System Evaluation

EOC-F combines many aspects of emergency management, and it is important to constantly involve end-users so that their requirements and feedback are accounted for. Throughout the iterative process, we frequently provide demos to our industry partners and local emergency management agencies (Fig. 15.6). They compare our system with the ones they use every day, and drive the development of EOC-F with their wants and needs. Role-specific experts such as social media analysts or incident commanders are often invited to help design or trial parts of EOC-F, so we can involve the full spectrum of users. Thanks to the involvement of all these users, EOC-F can be enhanced iteratively based on real user needs and valuable feedback.

15.5.7 Continuing Work

The EOC-F project continues to investigate and create new technologies to support emergency response planning and operation, in collaboration with our industry partner. To support crisis management teams, we will expand our work in

Fig. 15.6 Demo of EOC-F to local emergency response agencies and industry partners, in the Visualization Studio at the University of Calgary

multi-surface technology, analytics and wearable computing approaches. To enhance the handling of future emergencies, we are developing collaboration technology for a posteriori analysis of data gathered during an emergency and feed the lessons learned into training exercises.

While much of our work has focused on the ongoing emergency response, being able to predict what might happen in the event of an emergency is critical. Developed what-if scenarios allow for more effective decision-making on where to deploy resources, manage public safety, and manage the operation of the emergency response. Accurate scenarios also facilitate more effective emergency management training.

In addition to Predictive Emergency Analytics, After Action Review Emergency Analytics is equally crucial to preparedness in emergency response. It is important to be able to review what actions were taken and why decisions were made to help improve upon effective emergency management practices. Keeping a history of decisions, actions, and user interactions is critical to analyzing what happened during an emergency. The records serve to validate decisions made during the emergency, in the event these decisions are reviewed and criticized during the aftermath. Being able to effectively and efficiently analyze this data will give insight into the emergency and help improve the process for any future emergencies.

Part of the effort to collect data during an emergency is to use various sensors embedded in wearable technology. While EOC-F already collects heartrate, location, and movement information, there are many other sensors which can be used to improve safety of responders and situational awareness of EOC operators. Current solutions require responders to actively convey much of this information through radio, while EOC operators listen and record this information manually. This can be very inefficient in many cases. For example, a photo can instantly describe the situation, but cannot be sent through radio. Instead, a less accurate description is given verbally. We will expand on our current use of sensors, with the expectation that automatic logging and analysis of sensor data will lead to greater awareness in the EOC. The automated and consistent logging of data will also benefit subsequent reviews of the response effort.

15.6 Conclusion

The development of EOC-F has facilitated the investigation of how analytics-based, spatially-aware multi-surface environments can support teams managing emergencies in an EOC. We have created a prototype EOC in a multi-surface environment which integrates new technologies to support emergency response. Novel interactions and automated processes support emergency management in time-sensitive emergency situations. Future work to better utilize sensors will provide the information needed to improve the prediction, handling, and review of emergencies. Iterative feedback from end-users will continue to guide the development of EOC-F, enhancing public safety and emergency preparedness through the integration of new technologies.

Acknowledgements We greatly appreciate the continued support and enthusiasm from local emergency management agencies, and their valuable feedback. We thank our industry partner, C4i Consultants, for their knowledge and insight in navigating the emergency management domain. Finally, for their immense contribution to the development of EOC-F, we would like to thank Cooper Davies, Dan Gonzalez, Nicole Loison, Mahshid Marbouti, Alec McAllister, Irene Mayor, Teddy Seyed, Adam Saroko, Yuxi Wang, Jade White, and Dianna Yim.

References

1. 9-1-1 Statistics (2016). National Emergency Number Association (NENA). https://www.nena.org/?page=911Statistics. Accessed 16 Jun 2016
2. Alavi A, Kunz A, Sugimoto M, Fjeld M (2011) Dual mode IR position and state transfer for tangible tabletops. In: Proceedings of the ACM international conference on interactive tabletops and surfaces (ITS '11). ACM, New York, NY, USA, 278–279
3. Azazi A, Seyed T, Botros F, Sabourin D, Chan E, Maurer F (2013) Using multiple kinects to build larger multi-surface environments. In: Workshop on collaboration meets interactive surfaces in ACM interactive tabletops and surfaces 2013 (ITS '13), St Andrews, Scotland
4. Ballendat T, Marquardt N, Greenberg S (2010) Proxemic interaction: designing for a proximity and orientation-aware environment. In ACM international conference on interactive tabletops and surfaces (ITS '10). ACM, New York, NY, USA, 121–130
5. Beaudouin-Lafon M, Huot S, Nancel M, Mackay W, Pietriga E, Primet R, Wagner J et al (2012) Multisurface interaction in the wild room. Computer 45(4):48–56
6. Billinghurst M, Raffle H (2014) The glass class: designing wearable interfaces. In: CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14). ACM, New York, NY, USA, pp 1041–1042
7. Bragdon A, DeLine R, Hinckley K, Morris MR (2011) Code space: touch + air gesture hybrid interactions for supporting developer meetings. In: ACM international conference on interactive tabletops and surfaces (ITS '11). ACM, New York, NY, USA, pp 212–221
8. Brennan J, Krohmer J (2006) Principles of EMS systems. Jones and Barlett, Sudbury, Massachusetts, pp 98–119
9. Burns C, Seyed T, Hellmann T, Sousa MC, Maurer F (2013) A usable API for multi-surface systems. In: Workshop on envisioning future collaborative interactive spaces (BLEND '13). Paris, France
10. Butscher S, Müller J, Weiler A, Rädle R, Reiterer H, Scholl MH (2013) Multi-user twitter analysis for crisis room environments. In: Rester M et al (eds) Collaborative human-computer

interaction with big wall displays—BigWallHCI 2013. pp 28–34. Luxembourg: Publications Office of the European Union
11. Calgary 9-1-1 (2016). The City of Calgary. http://www.calgary.ca/CSPS/PSC/Pages/home.aspx. Accessed 16 Jun 2016
12. Cernea D, Mora S, Perez A, Ebert A, Kerren A, Divitini M, La Iglesia DG, Otero N (2012) Tangible and wearable user interfaces for supporting collaboration among emergency workers CRIWG'12. In: Herskovic Valeria, Ulrich Hoppe H, Jansen Marc, Ziegler Jürgen (eds) 18th international conference on Collaboration and Technology. Springer, Berlin, Heidelberg, pp 192–199
13. Chokshi A, Seyed T, Rodrigues FM, Maurer F (2014) ePlan multi-surface: A multi-surface environment for emergency response planning exercises. In: Proceedings of the ninth ACM international conference on interactive tabletops and surfaces (ITS '14). ACM, New York, NY, USA, pp 219–228
14. Cook KA, Thomas JJ (2005) Illuminating the path: The research and development agenda for visual analytics. No. PNNL-SA-45230. Pacific Northwest National Laboratory (PNNL), Richland, WA (US)
15. Dachselt R, Buchholz R (2009) Natural throw and tilt interaction between mobile phones and distant displays. In: CHI '09 extended abstracts on human factors in computing systems (CHI EA '09). ACM, New York, NY, USA, pp 3253–3258
16. Doeweling S, Tahiri T, Sowinski P, Schmidt B, Khalilbeigi M (2013) Support for collaborative situation analysis and planning in crisis management teams using interactive tabletops. In: 2013 ACM international conference on Interactive tabletops and surfaces (ITS '13)
17. E-Team (2016). NC4. http://nc4.com/Pages/eteam.aspx. Accessed 16 Jun 2016
18. Eagan JR, Lecolinet E, Klokmose CN (2013) What are applications in multisurface environments? In: International workshop on interactive, Ultra-High-Resolution Displays (CHI '13)
19. Fischer JE, Reeves S, Rodden T, Reece S, Ramchurn SD, Jones D (2015) Building a birds eye view: collaborative work in disaster response. In: 33rd annual ACM conference on human factors in computing systems (CHI'15)
20. Groeve TD, Rester M, Atkinson M, Steiner Y, Doherty B, Annunziato A, Galliano D (2013) "CrisisWall"—A Multi-Device, Multi-System Crisis Management Software. In Rester M et al (eds) Collaborative Human-computer interaction with big wall displays—BigWallHCI 2013. p. 23-27. Luxembourg: Publications Office of the European Union
21. Hall E (1966) The hidden dimension. Doubleday, Garden City
22. HoloLens (2016). Microsoft Corporation. https://www.microsoft.com/microsoft-hololens. Accessed 16 Jun 2016
23. Jetter H, Schöning J, Rädle R, Reiterer H, Rogers Y (2013) Colalborative Interactions in Fugure Crisis Rooms. In: Rester M et al (eds) Collaborative Human-Computer Interaction with Big Wall Displays-BigWallHCI 2013. pp 35–39. Luxembourg: Publications Office of the European Union
24. Kim SY, Jang Y, Mellema A, Ebert DS, Collinss T (2007) Visual analytics on mobile devices for emergency response. In: 2007 IEEE symposium on visual analytics science and technology (VAST '07). IEE, Washington, DC, USA, pp 35–42
25. Klann M (2007) Playing with fire: user-centered design of wearable computing for emergency response. In: Jobst L, Markus K (eds) 1st international conference on mobile information technology for emergency response (MobileResponse'07). Springer, Berlin, Heidelberg, pp 116–125
26. Kunz A, Alavi A, Landgren J, Yantaç AE, Woźniak P, Sárosi Z, Fjeld M (2013) Tangible tabletops for emergency response: an exploratory study. In: International conference on multimedia, interaction, design and innovation (MIDI '13)
27. Kyng M, Toftdahl NE, Kristensen M (2006) Challenges in designing interactive systems for emergency response. In: 6th conference on Designing Interactive systems (DIS '06)

28. Maciejewski R, Kim SY, King-Smith D, Ostmo K, Klosterman N, Mikkilineni AK, Ebert DS, Delp EJ, Collins TF (2008) Situational Awareness and visual analytics for emergency response and training. In: 2008 IEEE conference on technologies for homeland security. IEEE, Washington, DC, USA, pp 252–256
29. Mann S (1996) Smart clothing: the shift to wearable computing. Commun ACM 39, 8:23–24
30. Marbouti M, Maurer F (2016) Analyzing issues in social media use during emergency response. In: 15th IFIP conference on e-Business, e-Services and e-Society (I3E 2016), Wales, UK
31. Marquardt N, Greenberg S (2012) Informing the Design of Proxemic Interactions. IEEE Pervasive Computing 11, 2 (April 2012), 14–23
32. Nancel M, Wagner J, Pietriga E, Chapuis O, Mackay W (2011) Mid-air pan-and-zoom on wall-sized displays. In: SIGCHI conference on human factors in computing systems (CHI ‘11). ACM, New York, NY, USA, pp 177–186
33. Nebeling M, Norrie M (2012) jQMultiTouch: lightweight toolkit and development framework for multi-touch/multi-device web interfaces. In: 4th ACM SIGCHI symposium on Engineering interactive computing systems (EICS ‘12), ACM, New York, NY, USA, pp 61–70
34. Nebeling M, Mintsi T, Husmann M, Norrie M (2014) Interactive development of cross-device user interfaces. In: SIGCHI conference on human factors in computing systems (CHI ‘14), ACM, New York, NY, USA, pp 2793–2802
35. Nebeling M, Teunissen E, Husmann M, Norrie MC (2014) XDKinect: development framework for cross-device interaction using kinect. In: 2014 ACM SIGCHI symposium on Engineering interactive computing systems (EICS ‘14), ACM, Ney York, NY, USA, pp 65–74
36. Paternò F, Santoro C (2012) A logical framework for multi-device user interfaces. In: 4th ACM SIGCHI symposium on Engineering interactive computing systems (EICS ‘12). ACM, New York, NY, USA, pp 45–50
37. Piazza T, Heller H, Fjeld M (2009) Cermit: Co-located and remote collaborative system for emergency response management. In: Proceedings of Swedish computer graphics association (SIGRAD ‘09), p. 12–20
38. Qin Y, Liu J, Wu C, Shi Y (2012) uEmergency: a collaborative system for emergency management on very large tabletop. In: 2012 ACM international conference on Interactive tabletops and surfaces (ITS ‘12), pp 399–402
39. Rodrigues FM, Seyed T, Chokshi A, Maurer F (2014) Gesture design and feasibility in emergency response environments. In Workshop on gesture-based interaction design: communication and cognition (CHI ‘14)
40. Russell DM, Stefik MJ, Pirolli P, Card SK (1993) The cost structure of sensemaking. In: INTERACT ‘93 and CHI ‘93 conference on human factors in computing systems (CHI ‘93). ACM, New York, NY, USA, pp 269–276
41. Sakaki T, Okazaki M, Matsuo Y (2010) Earthquake shakes Twitter users: real-time event detection by social sensors. In: 19th international conference on World Wide Web (WWW ‘10). ACM, New York, NY, USA, pp 851–860
42. Savery C, Hurter C, Lesbordes R, Cordeil M, Graham N (2013) When Paper Meets Multi-Touch: a study of multimodal interactions in air traffic control. In: International conference on human computer interaction (INTERACT ‘13), pp 196–213
43. Selim E, Maurer F (2010) EGrid: supporting the control room operation of a utility company with multi-touch tables. In: ACM international conference on interactive tabletops and surfaces (ITS ‘10). ACM, New York, NY, USA, pp 289–290
44. Seyed T, Burns C, King P, Rodrigues FM, Sousa MC, Maurer F (2013) MRI Table Kinect: a multi-surface application for exploring volumetric medical imagery. In: Workshop on safer interaction in medical devices (MediCHI’13), Paris, France
45. Seyed T, Burns C, Sousa MC, Maurer F, Tang A (2012) Eliciting usable gestures for multi-display environments. In: 2012 ACM international conference on Interactive tabletops and surfaces (ITS ‘12). ACM, New York, NY, USA, pp 41–50

46. Seyed T, Sousa MC, Maurer F, Tang A (2013) SkyHunter: a multi-surface environment for supporting oil and gas exploration. In: ACM conference on interactive tabletops and surfaces 2013 (ITS '13), St Andrews, Scotland, pp 15–22
47. Starbird K, Palen L, Hughes AL, Vieweg S (2010) Chatter on the red: what hazards threat reveals about the social life of microblogged information. In: 2010 ACM conference on Computer supported cooperative work (CSCW '10). ACM, New York, NY, USA, pp 241–250
48. Tango (2016) Google Inc. https://get.google.com/tango. Accessed 16 Jun 2016
49. The New WebEOC (2016) Intermedix Corp. https://www.intermedix.com/solutions/webeoc. Accessed 16 Jun 2016
50. Toups ZO, Kerne A, Hamilton WA (2011) The Team Coordination Game: Zero-fidelity simulation abstracted from fire emergency response practice. In: ACM conference on transactions on computer-human interaction (ToCHI '11), vol. 18, issue 4. ACM, New York, NY, USA, p 23
51. Vieweg S, Hughes AL, Starbird K, Palen L (2010) Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In: SIGCHI conference on human factors in computing systems (CHI '10). ACM, New York, NY, USA, pp 1079–1088
52. Vogel D, Balakrishnan R (2004) Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In: Proceedings of the 17th annual ACM symposium on User interface software and technology (UIST '04). ACM, New York, NY, USA, pp 137–146
53. What is the Calgary Emergency Management Agency? (2016) The City of Calgary. http://www.calgary.ca/CSPS/cema/Pages/The-Emergency-Operations-Centre.aspx. Accessed 16 Jun 2016
54. Wobbrock JO, Morris MR, Wilson AD (2009) User-defined gestures for surface computing. In: sigchi conference on human factors in computing systems (CHI '09). ACM, New York, NY, USA, pp 1083–1092
55. Wu A, Convertino G, Ganoe C, Carroll JM, Zhang XL (2013) Supporting collaborative sense-making in emergency management through geo-visualization. Int J Hum-Comput Stud 71(1):4–23
56. Wu A, Zhang XL (2009) Supporting collaborative sensemaking in map-based emergency management and planning. In: ACM 2009 international conference on Supporting group work (GROUP '09). ACM, New York, NY, USA, pp 395–396

Chapter 16
Security in User Interfaces Distributed Amongst Dynamic Sets of Devices and Users

Luca Frosini and Fabio Paternò

Abstract Given the increasing availability of many devices in our daily environments, distributed user interfaces are becoming more and more used. However, they raise many issues related to security, which current frameworks for distributed user interfaces have yet to address adequately. We present a solution for this purpose, able to exploit public key certificates for authentication and encryption as well. The presented solution consists of a reference software architecture for secure distributed user interfaces and a corresponding implementation. We also report on an example application for a city guide supporting collaborative cross-device user interfaces.

16.1 Introduction

Given the proliferation of various types of devices in our daily environments, various frameworks for cross-device interaction have been recently proposed in the scientific community (e.g. [4, 8, 11]) and are receiving increasing attention even in a business perspective. The possibility of dynamically distributing user interface elements across different devices at the same time helps users to collaborate (e.g. to perform collaborative searches), improve group experiences (e.g. in guided tours), perform a task on the most suitable device, or use one device in coordination with others (such as using a mobile device to control a public wide screen).

In order to keep synchronized the parts of the user interface that are distributed across different devices, they must exchange information. This communication can raise some security and privacy issues. For example, the credit card number in a form should not be visible to people other than the credit card owner, even if the

L. Frosini (✉) · F. Paternò (✉)
CNR-ISTI, Via Moruzzi 1, 56124 Pisa, Italy
e-mail: luca.frosini@isti.cnr.it

F. Paternò
e-mail: fabio.paterno@isti.cnr.it

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_16

rest of the form can be. Moreover, the possibility of distributing user interface elements has to be controlled only by the authorized devices.

Analyzing the literature and current practices in both DUIs and security, and taking into account the needs of relevant software companies, we have identified a first list of requirements that effective support for security in Distributed User Interfaces (DUIs) should satisfy:

- Authentication from multiple identity providers, we need a solution to authenticate entities (devices or users) from different organizations. This allows an entity to use its own credentials to operate in multiple applications/contexts without the need to create new ones ad hoc.
- Dynamic identity provider addition/removal, calls for functionalities to easily add or remove an organization from the list of those who can issue credentials that can also be used for DUI applications. This allows application owners to change the authorized identity providers over time, for example for commercial agreements.
- Strong Cryptography mechanisms, to ensure data security in communication we need some strong cryptographic support as well.
- The solution should be inexpensive even when the number of users and devices involved increases, the cost and impact of the security solution must be limited and relatively constant as the number of applications using it grows.
- Authentication between devices even in ad hoc networks and when no internet connection is available, we define this case as offline mode: it means that devices can communicate with each other (e.g. one acts as access point among them) but they do not communicate with the external world. Thus, if one device has no Internet connection it can still communicate with the others devices in the ad hoc network.

In the paper, after related work discussion we introduce an example scenario to highlight the type of issues we want to address. Next, an architecture for secure distributed user interfaces is introduced, followed by the description of a solution to address the main security requirements in this context. We then discuss an example collaborative application and draw some conclusions with indications of future work.

16.2 Related Work

As evidenced by Hong and Landay [7], security and privacy are critical aspects in pervasive and ubiquitous computing.

Some research work has started to address such issues. DeLuca et al. [3] proposed a solution to authenticate on personal devices by using the back (instead of the front) of the input device. In the context of public environments, such as Internet Cafés, Sharp et al. [10] proposed a solution based on the use of a personal mobile as input device to provide private personal information instead of using input peripherals of insecure public devices, such as computers in Internet Café, which can be compromised by hardware or software input loggers. Alt et al. [1] have proposed a solution to provide personal information preserving privacy by using a public display. In such examples the interface distribution is used as a technology to enhance privacy and security. In our case we aim to endow the user interface distribution itself with support for preserving privacy and security. Ghiani et al. [5] proposed a solution to migrate already existing Web applications capable of addressing some security issues such as entering personal private information. That proposal was based on a migration server, while we propose a solution for distributed architectures supporting cross-device access and interaction.

In terms of user interface development frameworks, a few contributions have addressed security issues. Roesner et al. [9] proposed a UI toolkit mechanism to embed pieces of UI elements provided by different sources in a single user interface. The toolkit provides data isolation between different composed pieces of user interface in order to address related security issues, but they do not consider security problems in user interfaces distributed across different devices. Arthur and Olsen [2] proposed the XICE toolkit for securely distributing UIs, which allows developers to specify user interface elements that contain personal private information and cannot be migrated to an unsecure device without the explicit user consent. We propose a solution capable of addressing this situation, but can also securely transfer private information between different devices.

Heupel et al. [6] presented IdeREST: a solution for DUIs to enhance secure and privacy-preserving collaboration. The solution indicates how DUIs could profit from a proof-based anonymous credential system in a collaborative environment. To accomplish this there is an online entity that releases certificate-based proof to the service provider. Their solution provides credentials without revealing user information, but cannot be applied in offline mode, thus it is less general than the solution that we propose.

To summarize, while various recent proposals for supporting distributed user interfaces have been put forward and there is a general agreement that they raise various security concerns, there is still a lack of general, inexpensive, and modular solutions to address them.

The aim of this paper is to analyze the security problems in Distributed User Interfaces (DUIs), and propose a solution that can also be included in current frameworks for DUIs without modifying the underlining distribution protocol among the involved components. The presented solution has the advantage of exploiting public key cryptography methods while avoiding the complexity of deploying the public key and authentication infrastructure. In addition, the cost of this implementation is negligible compared to certification authority-based solutions.

16.3 Example Scenario

In order to introduce the possible security issues, we consider a tourist guide accompanying groups of tourists in a city tour. To enrich the tour the city guide app enables the guides mobile devices to share content on large public displays installed by the tourist office externally or internally to points of interest (see Fig. 16.1a), or with tourists mobile devices (tablet or smartphone) (see Fig. 16.1b).

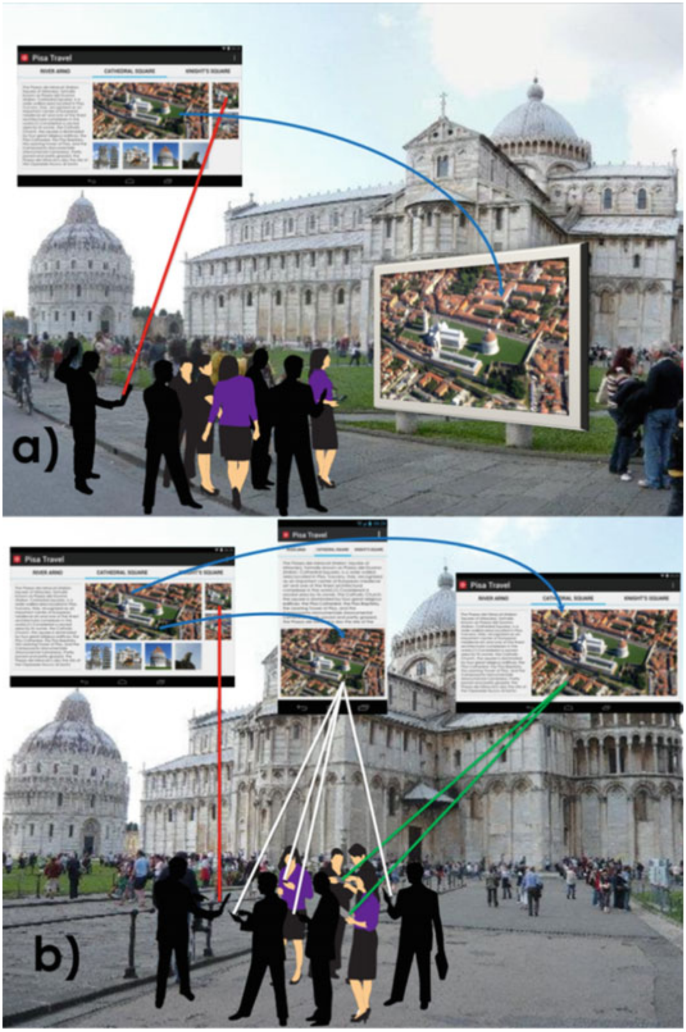


Fig. 16.1 Example scenario for the city guide showing use **a** with a large public display, and **b** amongst mobile devices

The tourist guides are the only authorized entities that can take control of the large public displays, and control the content sharing among devices.

In some case, the tour is in critical indoor places where there is no external Internet connection available. In this case an ad hoc network between participants must be created and this scenario must guarantee the same properties.

In this situation we have different security problems. The users participating to the tour must be authenticated to avoid unwanted participants. The information flowing from/to different devices has to be encrypted to avoid someone intercepting this information. UI distribution changes must be made only by authorized people (i.e. the tourist guides). The public display is a sensible device which must be prevented from abuse. We can summarize the security and privacy issues in four main points:

- Authentication of the involved entities (user or devices);
- Authorization of entities to perform a distribution change request;
- Authenticity of distribution change requests;
- Data Privacy.

In the following we analyze all such points and show how to exploit cryptography techniques in DUIs in order to solve the first three points. Regarding Data Privacy, cryptography is not enough and will be used in conjunction with techniques that control and limit which parts of the UI can be shared, and where.

16.4 A Software Architecture for Secure DUIs

Distributed user interfaces allow users to exploit elements distributed across multiple devices at a given time to access their applications. This implies the need to keep synchronized the state of the components located in multiple devices. Figure 16.2 represents our reference architecture for DUIs. It is composed of the main modules present in Clients with DUI capabilities and the modules present in the Distribution Orchestrator (henceforth Orchestrator).

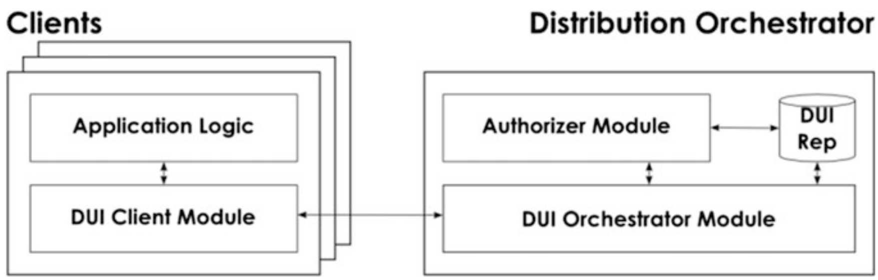


Fig. 16.2 DUI reference architecture

The Orchestrator is responsible for managing distribution update requests coming from Clients, and notifying the distribution updates to all involved clients when the requested update is accepted and processed. Clients and Orchestrator communicate with each other using a DUI protocol.

Our DUI protocol supports three main concepts:

- User Roles: each user can belong to one or more Role(s). Each user is authenticated (see Identity Card section) and then mapped to a Role depending on the information provided in her own identity card.
- Types of Devices: each device can belong to one or more device type, depending on its own capabilities.
- UI parts: the UI part to distribute.

All these three concepts can be differently specified depending on the application considered.

The Orchestrator uses an Authorizer Module, which is responsible for accepting or rejecting the distribution update requests received from Clients. Moreover, the Orchestrator can use a repository (DUI Rep) containing information related to the configurations of user interface distributions (e.g. user interface elements associated with devices types, user roles grants).

In this architecture there is one device that acts as Orchestrator and two or more devices that act as Clients participating in the distribution sessions. The Orchestrator device can also be a Client of the distribution sessions. During a session, the Orchestrator can dynamically migrate from one device to another (Migrating Orchestrator Scenario), for example because the initial Orchestrator device is leaving the session and another one has to continue to orchestrate the distribution. The Clients do not communicate directly with each other, instead any request/notification of distribution change flows through the Orchestrator.

UI Distribution can be changed according to external events (triggered from events generated by one or more device sensors) or through explicit actions by the users. For example, a user can long press on a UI part and, if the user has the associated grants, a Distribution Panel (see Fig. 16.3) is activated to change its distribution across the available devices. The panel allows users to indicate whether the selected UI element should be enabled or disabled or invisible on devices that can be selected by type or role or id.

16.4.1 Security Aspects

First of all, for supporting security the Orchestrator must be trusted. Even if trusted, the Orchestrator must be prevented from accessing any sensitive information that is flowing (e.g. credit card numbers must be readable only by the devices of the credit card owner).

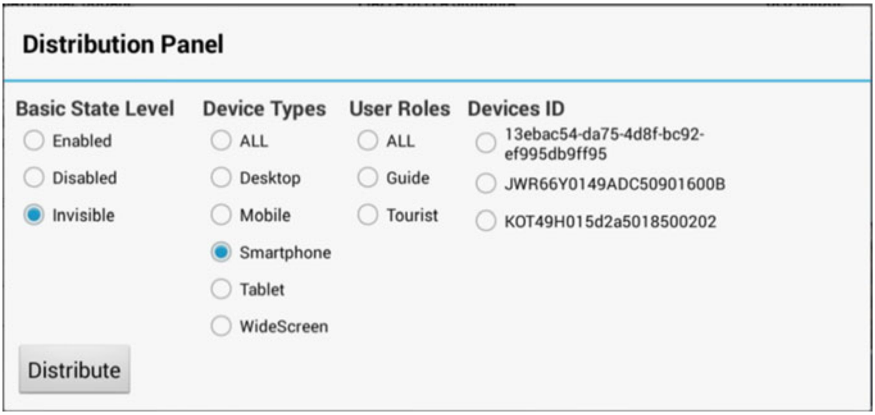


Fig. 16.3 Distribution panel

If Clients and Orchestrator communicate with each other in clear text, anyone can intercept such data flowing. Cryptography helps to solve this problem but it has to be used in conjunction with authentication mechanisms, which guarantee that the messages are exchanged between the desired parties. Moreover, there is a need for a mechanism to prevent an attacker reusing a previous message to change the distribution state.

Indeed, if only a cryptographic system is put in place, it may happen that the Orchestrator receives an encrypted message requesting to participate in a DUI claiming that the device is owned by a given user. However, if the Orchestrator has no mechanism to verify that the device is really owned by that user, the system would provide a false sense of security. Likewise, a Client must be sure it is communicating with the trusted Orchestrator. Thus, security is not just a matter of encryption, as sometimes perceived, but a complete and integrated security strategy must be provided.

In particular, four different properties must be guaranteed at the same time for each message exchanged between Clients and Orchestrator:

- source authentication;
- data integrity;
- privacy of the exchanged information;
- data freshness, in order to avoid delays in distribution commands with undesired effects (e.g. a recent change in the value of a text box not displaying immediately on other connected devices).

To make the architecture secure, a Security Module (Fig. 16.4) must be added to both the Clients and the Orchestrator. In our solution the Security Modules use Public-Key Cryptography techniques to guarantee the above properties because they provide strong security guarantees. They have the drawback of requiring deployment and maintenance of a Public Key Infrastructure (PKI), which is the set

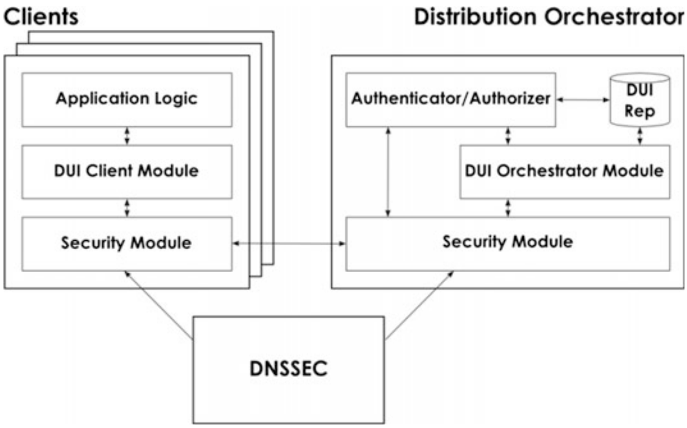


Fig. 16.4 Secure DUI reference architecture

of components used to retrieve and check the correctness of public keys. To address this issue we use the Domain Name System Security Extensions (DNSSEC) as described in the Public Key Infrastructure section.

16.4.2 Public Key Cryptography Properties

In Public-Key Cryptography each entity has a private key KPRIV and a public key KPUB. The first is known only by the owner and the second is public and must be known by third parties that want to communicate privately with the owner of the private key. This type of cryptography has the following important properties:

- Encrypting a message with KPUB, the message can only be decrypted by the entity which owns the KPRIV. In other words, anyone that knows the KPUB of A can send an encrypted message to A, and A is the only one that can decrypt the message.
- Any message encrypted with a private key KPRIV can be decrypted by anyone that knows the KPUB. The encrypted message can only be generated by the owner of the KPRIV.
- KPRIV can be used to sign a message. Verifying the signature of the message ensures that the message has been produced by the entity that has signed it and that it has not been altered.
- Any of these properties can be used in conjunction to achieve greater security.

16.4.3 Public Key Infrastructure

In Public-Key cryptography the main problem is retrieving the public key of the entity we want to communicate with. Public-key cryptography provides strong security guarantees, but has the drawback of requiring deployment and maintenance of the PKI.

To address this issue the most common solution is based on trusting Certification Authorities (e.g. Verisign, Go Daddy) in order to certify the public keys.

By knowing the public key of the certification authority in advance, it is possible to verify that the certificate presented is authentic. To use certification authority-based solutions, each entity has to follow the procedure for identification and has to pay the cost of the certificate, which is often considerable. Moreover, the certificates have a limited validity in time, so renewing the certificate involves further costs. In this respect solutions such as the free Let's Encrypt[1] Certification Authority can only be used for server certificates, so they are not sufficiently suitable in this context.

Certification Authority-based solutions are thus not affordable in terms of costs and overhead due to the identification procedure. Therefore, we have sought a solution that overcomes or reduces the impact of deploying and maintaining the PKI. To this end, we propose a solution that uses DNSSEC, which is a standard extension of DNS that guarantees the data integrity of the information published on DNS and the authentication of the sources that have generated it.

First of all, DNS is a distributed technology, which is well suited to distributed environments, and there are many well-established practices to deploy and maintain a domain. Moreover, it represents an inexpensive solution whose cost is simply that of owning the organization domain.

DNS was initially designed to associate IPs to hostnames. Over the years it has evolved as a distributed way to publish publicly available information. For this reason DNS has different types of records called Resource Records (RR) with different formatting to distinguish and identify the content. One of the standardized DNS records is CERT, which is designed to contain X.509 certificates/CRLs or PGP certificates/revocations and meets our needs well.

16.4.4 Crypto Message for DUIs

In the proposed solution, any message that flows from Clients to the Orchestrator is first signed with the client private key. This allows the Orchestrator to verify the integrity of the message and guarantees that the message comes exclusively from the Client. The signed message is encrypted with the public key of the Orchestrator.

---

[1] https://letsencrypt.org/.

This guarantees that the message can only be decrypted by the Orchestrator, so the communication is reserved.

Vice versa, when the Orchestrator sends a message to a Client, first of all it signs the message with its own private key and then encrypts the message with the public key of the Client.

Using this mechanism source authentication, data integrity and privacy of the exchanged information are guaranteed.

To guarantee data freshness, a Timestamp is inserted by the sender before signing the message. Furthermore, a Universally Unique IDentifier (UUID) is used to protect against reply and known-plaintext attacks (when the hacker needs samples of both the plaintext and its encrypted version in order to access further secret information). Such messages, which we term Crypto Messages, are formatted as follows (see Fig. 16.5):

- Message is the clear text message exchanged between a Client and Orchestrator in an unsecure architecture.
- Unique Message is composed of the Message with UUID and Timestamp.
- Signed Message contains the Unique Message and its own Signature. The Signature is made with the private key of the originator.
- Crypto Message is the encryption of Signed Message made with the public key of the addressee.

For each Unique Message the Orchestrator checks if the received UUID has already been seen. If the UUID has not yet been seen, it is recorded for further checks and the Message is accepted. Moreover, a Timestamp is added. Any messages received by the Orchestrator after a (customizable) amount of time from this timestamp are rejected. This is useful to avoid accepting requests to change the distribution state that were sent too early to be still valid. This can happen in case of huge network latency.

The UUID is needed because it is dynamic and thus it helps to reduce the amount of information that an attacker can use over a known-plaintext attack.

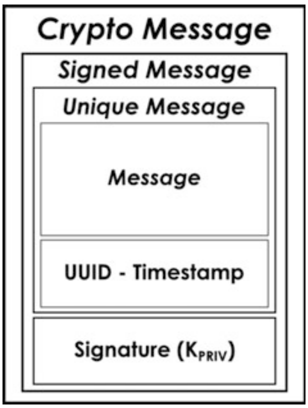


Fig. 16.5 Crypto message format

A known-plaintext attack is a type of attack for cryptanalysis where the attacker has samples of both the plaintext and its encrypted version, which can be used to reveal further secret information.

This use of the Timestamp does not require that the clock be perfectly synchronized but at least that there should not be appreciable difference between device clocks. When a Client notes that its messages are rejected because of request timeout it needs to use a Network Time Protocol (NTP) client to realign its clock.

In terms of implementation, on some platforms (e.g. Android) the clock change is not possible if the running application has not root privileges. In some platform the clock synchronization can be made a priori and/or scheduled (such as on Linux where it can be done at machine startup and daily via cron). For this reason, in our implementation, every time the application starts, instead of changing the clock, an NTP client is used to calculate the difference between the device clock and the one to be set. Whenever a timestamp is generated or checked then the difference is algebraically added to the current clock of the device.

16.5 Security Properties

As indicated in the example scenario description there are four main security requirements that have to be satisfied by the Security Module. In this section we analyze how they are addressed.

16.5.1 Authentication

Any entity involved in user interface distribution has to be authenticated to prevent unauthorized participation. Many of the authentication solutions are based on username and password. Apart from the weakness of this paradigm the entity that verifies this credential often does not coincide with the service provider.

In many environments authentication can be carried out through different sources. To solve this (and other) problems, some protocols, such as SAML, OAuth and OpenID, have been proposed. With these kinds of protocols the identity provider is a separate entity from the service provider. The identity provider verifies the user credentials, and provides the service provider with user identities. This type of solution works well but it does not support offline mode scenarios.

Our proposal, instead, is mainly based on an identity card issued by the organization that the entity belongs to. Each entity involved in the distribution shows its own identity card. If the card is issued by an organization trusted for the application and the entity profile is accepted, then the entity is allowed to participate in the distribution session.

16.5.2 The Identity Card

Each entity that attempts to participate in user interface distribution receives from its organization an XML document (the identity card) containing information related to the entity. Apart from the entity information, the document always contains the public key of the entity it identifies as well.

The document is signed by the organization that releases the identity card. By verifying the validity of the signature of the document the DUI components can be sure that the document was released by the entity that has signed it, and that was not altered (authenticity and integrity). If the document also contains the public key of the entity it identifies, the integrity and authenticity properties of the document guarantee integrity and authenticity of the public key contained. The document is part of the Message, which is encapsulated in a Crypto Message to be sent to the Orchestrator.

We can now understand that a compromised Orchestrator device (e.g. due to an attacker taking silent control of it) could use the identity card of another entity to try to impersonate it. For this purpose, the Orchestrator does not hold the private key of the entity, so it is not able to create a valid Signed Message. The only chance it has is to try a reply attack.

16.5.3 Authorization

As described in the previous section, entities introduce themselves to the Orchestrator through an identity card signed by a trusted-party. The authorizer module (Fig. 16.3) analyzes the identity card information and uses it to identify the rights associated with the entity for the specific application. The mappings between rights and entities types are stored in the database containing the possible user interface distributions (DUI Rep). This authorization solution has the possibility of:

- authorizing a distribution action without using a central authorization center;
- assigning authorization policies to any entities involved in the distribution even without knowing the entity a priori.

16.5.4 Authenticity

Each message exchanged between Orchestrator and Client is encapsulated in a Crypto Message.

The properties guaranteed by the Crypto Message (source authentication, data integrity, privacy of the exchanged information, data freshness) guarantee the authenticity of the message.

16.5.5 Data Privacy

Apart from the aspect of data cryptography in transit, some parts of the UI can contain some personal information that has not to be shared with other involved entities.

To solve the privacy related problem, cryptography is not enough, we have to prevent sending data to entities that should not know the information. To address this problem we introduce three concepts: Public and Private devices; Sensitive Information; and Reserved Information.

Public and Private Devices, Any device that wants to participate to distribution is classified as public or private. We define private a device when its UI is only visible to the owner and the user input is made only by such user. A public device is a device whose UI is visible to any user and, in some cases, the input can be entered by more than one user in parallel or sequentially.

Sensitive Information, Sensitive information is information that a user can share with other private devices but never with public devices. This kind of information is, for example, the email address of participants in a collaborative activity. This type of information must be visible only to the addressed entity, and the distribution orchestrator plays a fundamental role. The Orchestrator receives the sensitive information encrypted with its public key by the sender device, and must share this information only with target devices/users. Then, the Orchestrator uses the public keys of the target devices/users to encrypt such data to share with them. Since the Orchestrator is capable of reading this information it must be a trusted device.

Reserved Information, Reserved information is information that a user does not want to share with other users or any public devices, but only with the others own private devices. For example, the credit card number in a shopping application is a reserved piece of information.

In this case the information that should be communicated must be encrypted before inserting it in the Message. The encryption is made with the public key of the owner. In this way the Crypto Message does not change its format and the reserved information inside the content of the Message will be decrypted only by the entity that owns the private key. The Orchestrator in this case is not capable of decrypting the credit card number so this information remains reserved.

16.6 PKI Deployment

As mentioned above we use a domain secured by DNSSEC as deployment facility to publish X.509 public key certificates.

16.6.1 Public Key Certificate Resolution Schema

Let us suppose that we own a DNSSEC secured domain example.org and we have developed different applications supporting user interface distribution. Each application trusts different entities and organizations. The trusted entities of different application have to be separated. To support this situation under the example.org domain, we create a different subzone with the name of the corresponding application. Suppose we have developed the hello and simple applications, then we have the subzones:

- hello.example.org
- simple.example.org

For each application, two further subzones are created:

- trusted (i.e. trusted.hello.example.org)
- entities (i.e. entities.hello.example.org)

The trusted zone is used to publish CERT records containing the public key of organizations that are accepted for issuing the identity cards. The entities zone is used to publish the public key certificates of users or devices that are trusted a priori, for example the public key of a device that can act as Orchestrator.

This schema has two main advantages:

- the trusted parties do not need to buy a certificate from well-known Certification Authorities, but they can create a self-signed certificate. It is sufficient that the application developer knows this public key and publishes it on DNSSEC.
- The application can dynamically verify which entities are considered trusted parties to issue identity cards. The trusted party can dynamically change during the application life. If an entity is no longer considered trusted the key is removed. If the entity changes the public key, the old one is removed and the new one is inserted.

16.7 Offline Mode Support

In our proposal a device can prepare itself for offline mode. To this aim the device requests all CERT records available in the trusted and entities subzones from DNSSEC, and saves them locally. This operation must be done either before going offline if the situation is detectable, or scheduled at least once a day even when not needed.

DNS has been designed so that the records, even if public are not enumerable (i.e. there is no way to get the full list of the published records), thus in order to support this operation we use a TXT DNS Record named __all_public_keys__ containing the list of all public keys published within the trusted and entities subzone.

16.8 Example Application

Using our DUI framework we have developed some applications. One is the mobile city guide which has been introduced in the example scenario section. This applications has been designed in collaboration with a company which requires the satisfaction of the security requirements presented in the introduction. The mobile version of the application has been developed for Android. The large public display version is Web based.

The application supports guides accompanying groups of visitors who can have either tablets or smartphones. The application shows information supporting the mobile visit. The guide can dynamically decide which components (e.g. relevant content, interactive games) should be visualized on the tourist devices. Moreover, the guide has the right to select one of the public screens positioned in the city.

In this application, it is very important to guarantee that the public screen can be accessed only by authorized guides to avoid abuse.

Tourists who want to attend the city tour can rent a mobile device or request to install the application on their own mobile devices. When tourists join the tour they receive a virtual identity card issued by the tourist guide center containing an id, their name and surname and the role of tourist. The identity card also contains an indication of the session the tourist can participate in. Finally, it contains the user public key.

To obtain the identity card, the first time the application is launched, it requests a coupon code to the user. A handshake is performed with the tourist guide server. During the handshake the mobile device generates its own private and public keys and transmits securely the public key to the tourist guide server. The server finally issues the identity card containing the public key.

The guide device performs the same handshake to obtain the guide identity card, with the guide role associated. This role allows the guide to request distribution changes to the Orchestrator and to use the public large screens. The guide device is also able to act as an Orchestrator to support the visit in offline mode. To allow this possibility the guide device requests the migration of the Orchestrator to his device. Once accepted the old Orchestrator informs all the connected tourist devices to continue using the guide devices. This message contains the information to connect to the enhanced guide device. To recognize that the guide device is not impersonated, the tourist devices retrieve the public key of the guide device from the DNSSEC.

The devices download the local DNSSEC cache the first time before retrieving the id card. Moreover, they are configured to check whether there are any changes in the DNSSEC, concerning its cached records, each time the application starts or as soon as the internet connection became available.

In this way, the tourist devices can join a distribution just presenting the user identity card. Thanks to the identity card signature and the cached signer (tourist office) public key, the guide can authenticated the devices in offline mode.

The large public displays also have their own identity card. The public displays accepts distribution changes only from an authorized Orchestrator to avoid that an attacker can use this wide screen for inappropriate purposes. The public displays are configured to update the cached DNSSEC record periodically (the amount of time can be configured). The displays stops to accepts distribution changes if the cached records are not renewed from 12 h because this implies that the state can be compromised.

In terms of performance, the critical case for this application is when it is in offline mode with the guide device operating as Orchestrator because this can require some cryptography overhead. To test the solution usability in this case we used a Google Nexus 7 tablet as tourist guide device behaving as orchestrator and access point for five devices connected in offline mode. In this situation the application was still responsive and the users did not notice any particular delay.

16.9 Conclusions

The presented solution has been designed to guarantee authentication, authorization, authenticity and data privacy in collaborative distributed user interfaces. It consists in a software architecture for this purpose and a related implementation.

The proposal uses Public-Key Cryptography. The certificates are published on DNSSEC. The use of DNSSEC overcomes the complexity of creating and maintaining a PKI, and moreover reduces costs to simply that of owning a domain. The organization domain can be used if the Top Level Domain (TLD) supports DNSSEC too. At the time of writing more than 75 % of the TLDs are already secured by DNSSEC (http://stats.research.icann.org/dns/tld_report/).

It is expected that each entity with an email address will be able to publish its own public key certificate on DNSSEC (through its own email provider), which will be associated to the email address often used as a Distinguished Name (DN). This will create new opportunities for secure authentication, which will improve the security of the presented solution.

Any entity that wants to participate in a distribution session has to authenticate itself with the Orchestrator by presenting an identity card signed by a trusted party. The exchange of messages is made only between clients and Orchestrator and vice versa. All data exchanged between these entities are encrypted and packed, and finally signed within an UUID and a Timestamp.

In the paper, we also show that privacy is not just a matter of cryptography but also authentication plays an important role. The authentication allows the environment to know the real identities of the involved devices without revealing the user credentials to the orchestrator and allowing the authentication also when in offline mode.

Future work will be dedicated to providing further support to define specific privacy and security policies in collaborative distributed user interfaces.

References

1. Alt F, Kubitza T, Bial D et al Digifieds: insights into deploying digital public notice areas in the wild. In: Proceedings of MUM'11. ACM Press, pp 165–174
2. Arthur R, Olsen DR (2011) Privacy-aware shared UI toolkit for nomadic environments. Softw Pract Exp 42:601–628
3. De Luca A, von Zezschwitz E, Nguyen NDH, Maurer ME, Rubegni E, Scipioni MP, Langheinrich M (2013) Back of device authentication on smartphones. In: Proceedings of CHI'13. ACM Press
4. Frosini L, Paternò F User interface distribution in multi-device and multi-user environments with dynamically migrating engines. In: Proceedings of EICS 2014, pp 55–65
5. Ghiani G, Paternò F, Isoni L Security in migratory interactive web applications. In: Proceedings of MUM 2012. ACM, Article 15, 10 pages
6. Heupel M, Bourimi M, Schwarte P, Kesdogan D, Barth T, Villanueva PG (2013) Enhancing the security and usability of dui based collaboration with proof based access control. In distributed user interfaces. Springer, London
7. Hong JI, Landay JA (2004) An architecture for privacy-sensitive ubiquitous computing. In: MobiSys'04: proceedings of the 2nd international conference on mobile systems, applications, and services. ACM, New York, pp 177–189
8. Nebeling M, Mintsi T, Husmann M, Norrie M Interactive development of cross-device user interfaces. In: Proceedings of CHI 2014. ACM, pp 2793–2802
9. Roesner F, Fogarty J, Kohno T User interface toolkit mechanisms for securing interface elements. In: Proceedings of UIST'12. ACM, pp 239–250
10. Sharp R, Madhavapeddy A, Want R, Pering T Enhancing web browsing security on public terminals using mobile composition. In: Proceedings of MobiSys 2008. ACM, pp 94–105
11. Yang J, Wigdor D (2014) Panelrama: enabling easy specification of cross-device web applications. In: Proceedings of CHI 2014. ACM, pp 2783–2792

Chapter 17
Surface Applications for Security Analysis

Judith M. Brown, Jeff Wilson, Peter Simonyi, Miran Mirza and Robert Biddle

Abstract This chapter relates to human factors in computer security, and how surface technology might support security analysis. This specific domain allowed us to investigate surface application design and development in an established context, and thus learn how the real needs of the domain might best be supported. Throughout, we were fortunate to have partners in industry and government working in the domain who were able to give us advice and feedback. A number of projects were conducted over the span of our research program, each one offering findings that informed later projects. In this chapter, we provide an outline of our work, summarizing each of the main projects and their findings. We cover: (1) a literature review. (2) Ethnographic studies of firstly operators and technicians in seven operations centres, and secondly a team of ten professional analysts working in the security domain; (3) ACH Walkthrough, a collaborative web-based decision-making tool; (4) Ra, a tool that supports rollback, playback and other explorative actions when using web applications like ACH Walkthrough; and (5) Strata, a tool that allows for the annotation of web applications, enabling the work of collaborative teams.

J.M. Brown · J. Wilson · P. Simonyi · M. Mirza · R. Biddle (✉)
School of Computer Science, Carleton University, Ottawa, ON, Canada
e-mail: Robert.Biddle@Carleton.ca

J.M. Brown
e-mail: JudithM.Brown@Carleton.ca

J. Wilson
e-mail: rjp.wilson@gmail.com

P. Simonyi
e-mail: peter@petersimonyi.ca

M. Mirza
e-mail: MiranMirza@gmail.com

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_17

17.1 Introduction

Our group's recent work has concerned software development for surface applications. Other research of ours concentrates on human factors in computer security, so we decided to explore how surface technology might support security analysis. In this chapter, we provide an outline of our work, summarizing each of the main projects and their findings, while also referencing publications that provide extensive detail. We conclude our paper with a summary of themes that emerged across all the projects.

Our research on security analysis has focused on work to understand information related to security, including both computer security and security in a more general sense. The computer security work involved incident reports from users and managers, and automatic reports from a variety of tools such as intrusion detection and log tracking systems. Such work is commonly carried out in security operations centres (SOCs). The more general security work resembled intelligence analysis where information relating to a variety of potential threats to security are studied. Such work is typically secretive, both in national security intelligence, and also business intelligence work.

In all this work, the key activity is sensemaking: "the process of searching for a representation and encoding data in that representation to answer task-specific questions" [52]. Sensemaking in security work presents some particular challenges because of the adversarial nature of the context. Some key information will be missing, some information will be unreliable, and some irrelevant. Moreover, due to the adversarial element the information may be intentionally deceptive: other actors may be concealing information, and providing misleading or irrelevant information. Throughout, analysts will be required to make judgments either individually or collectively. The process of analysis is not straight-forward.

Several techniques and technologies have evolved to address challenges in security analysis. For example, Heuer and Pherson [31] describe a number of "structured analytic techniques" to assist intelligence analysts. The book *Illuminating the Path* [59] illustrated how "Visual Analytics" can be used to provide technological support for the sensemaking process, e.g., by using information visualization to expose patterns that might otherwise go unnoticed. Our work drew on these ideas, but specifically looked at *collaborative* security analysis, and how surface computing might help.

Surface computing is likely to become commonplace in some domains such as entertainment and education. However, we also expect large surfaces will serve a primary role in supporting collaborative work. Meeting rooms and team environments will be designed to feature large surfaces. These large surfaces, while being a key to enabling more collaborative computing environments, will typically work in concert with other display devices in mixed-display environments, where both individual and team devices are used together to support collaborative work. We believe large displays and mixed display environments (combinations of large displays, tablets, smart phones and other types of surfaces) will become ubiquitous in office environments of the future. For example, this is the proposed role for the Microsoft Surface Hub [43]. Analysis for the business case of this approach appears compelling [48]. In this emerging and novel context, application software, and especially application interfaces, must be explicitly and purposefully designed and developed to support surface computing for collaborative work.

The main sections of this paper are:

- Review of Surface Computing for Collaborative Analysis
- Field Studies of Security Analysts at Work
- ACH Walkthrough: Software to Support Security Analysis
- Ra: Support for Web Application Interaction History
- Strata: Annotation for Web Applications.

17.2 Review of Surface Computing for Collaborative Analysis

Our first step in this sequence of projects was to conduct an extensive survey of the area. We covered a wide range of topics including the literature specifically on the topic, relevant theory, significant interaction design issues, and the underlying technologies and development platforms. The survey was published as a 140 page book by Morgan and Claypool [10]. Within the broader context of collaborative analysis work we particularly discussed co-located analysis work in the security domain which is typically either network security or intelligence work. Our book [10] describes the current research in this space, and provides a perspective on it based on our own experiences and expertise. In the paragraphs below we cover our main findings providing illustrative references.

We began by reviewing the underlying technology for surface interaction by Han [27], Haller et al. [26], Spindler et al. [57], and others, especially looking at large surfaces and novel methods for interaction by researchers such as Jacob et al. [37], Wigdor and Wixon [64], and Buxton [14].

We identified research on surface technology issues that are particularly important to analysis work. For example, Tuddenham, Davies and Robinson show how document flow issues may impact security analysis work [60]. Additionally, individual work on digital artifacts using laptop and workstation computers in theory should be compatible with digital tabletops and digital wall displays so that digital artifacts don't have to be transformed into analog (paper) artifacts to be taken to a meeting. In practice, however, the problem of moving digital artifacts seamlessly between surfaces has not yet been resolved.

Also important is that human communication is rife with indexical references, i.e., pointing, which involves verbally or physically indicating something, as highlighted in work by Genest and Gutwin [23]. In artifact-rich environments, such as sensemaking, so behaviour is common and saves much time. Other issues in collaborative security analysis arise when multiple display environments (including both small and very large displays) are used, especially when diverse kinds of displays are being used together as explored by Wallace et al. [61, 62] and Song et al. [56]. Finally, we identified easy text entry and interactor identity issues as important unresolved challenges in the domain [42].

We found that, in general, interaction design for surface computing presents novel challenges that are not easily solved by mechanisms used for traditional desktop interaction design, a topic explored by Jacob et al. [37], Wigdor and Wixon [64], and Buxton [14]. Menus and scrollbars are less critical in the context of large surfaces, and new approaches to pointing, selecting, and hovering are required. Gesturing is the emerging approach, and is still evolving [38, 66].

Andrews' Endert's and North's research has clearly shown there are many advantages for large displays for individuals [2]. These include cognitive benefits, increased productivity, reduced errors, and greater satisfaction. We believe these benefits to individuals often carry over into collaborative situations. Research on groups and teams, however, is much newer. Early results by Isenberg et al. and Anslow et al. are by and large positive [3, 36], but also indicate that it is very important that surface applications be carefully designed [67]. For example, to increase situation awareness in contexts where groups are collaborating loosely, the research shows that it is very important to reduce the amount of information that is shared to no more than what is required [61, 62]. Other research indicates the positioning and arrangement of displays can impact collaboration. In mixed-display environments the research shows that it would be important to be clear about the most important objectives of the collaboration so that choices about display devices and functionality can be made with these considerations in mind.

Understanding analysis work is not easy. Designers and developers of tools often have undeclared assumptions about what analysis work is, and these assumptions can easily become embedded in the tools, resulting in a rupture between the work at hand and the tools to accomplish the work [58]. Theories have been applied to aid understanding of individual analysis work, primarily based on understanding cognition. However, increasing amounts of data and larger and more complex analyses have led to a need for collaborative analysis. Collaborative artifact-mediated work can be understood from a variety of theoretical perspectives. In particular, we reviewed Distributed Cognition, Evolutionary Psychology, Attention mechanisms, Group situation awareness, and Cultural-Historical Activity Theory (See Sect. 5.4 in [10]). However, collaborative work, such as complex collaborative work in specialized domains, can be challenging to understand and predict, particularly where new technology presents unfamiliar opportunities.

With respect to software architecture and development there is, and will continue to be, some turbulence as technology standards and design best practices emerge and become established. It is very important for designers to understand this, as the challenges for developers are much greater than those presented by familiar and well-understood WIMP (windows, icons, menus and pointer) interfaces. Diversity of toolkits and libraries may make cross-platform development problematic until the advantages of interoperability influence the market. Similarly, heterogeneity of data sources and formats may present challenges.

Additionally, few multi-touch surfaces have the means to identify the source of gestures when several collaborators are interacting with a single screen [42]. However, the novel ubiquity and low cost of tablets and smartphones offers an excellent opportunity to provide the identity of collaborators in mixed-surface environments, as well as additional modes of interaction beyond touch. Further, tablets and smartphones as additional devices for collaboration can offer opportunities for private exploration, offline data manipulation and preparation, and interactions requiring personalization or authority.

Some technologies have already dealt with multiplicity and diversity at the infrastructural level, particularly web-based frameworks, and seem to be a good starting point for collaborating across multiple devices (See Sect. 6.2 in [10]). However, there remain differences in how gestures are shared with browsers within each of the main handheld operating systems, and so it may be worth designing for a mix of browser-based and native code.

There are also deeper issues. The challenge of sharing application state across multiple devices gives rise to an important question of the identity and "ownership" of objects. It is important to draw appropriate distinctions between actual objects and inferred or proxy objects. The mutability (the ability of objects to be changed) of shared objects must follow a logic that meets mutually shared goals of participants.

Our survey left us optimistic that large surfaces and mixed-display environments seem well poised to support co-located collaborative analysis work. However, it was clear that design for surface applications in the analysis domain requires a system perspective. Surface computing is only as useful as its application software, and applications for collaborative analysis work need careful study of the domain, and carefully designed interfaces and software. Further, surface computing environments need appropriate accommodation and infrastructure, which also needs to be designed. In this context it is important to design with an eye to end-user interaction, end-user experiences, *and* the broader environment, which would include team interactions and the physical aspects of the workplace.

17.3 Field Studies of Security Analysts at Work

The next step in our research program was to conduct field studies. Especially in the domain of security analysis, access to professionals can be very difficult to obtain, and our partnerships with industry and government organizations were critical.

We conducted a number of studies in two related domains. In a first set of studies, we carried out observations and interviews of operations centres. In this set of studies, there were seven sites involved. The sites represented a variety of industry and government contexts from financial transaction processing to healthcare support. Our study involved many hours and days of observations, and interviews across a range of workers and stakeholders. In total we shadowed 129 individuals for a total of 250 h of observations, conducted 38 interviews, and facilitated 11 information-gathering meetings with executives of Operations Centers. Our analysis of the data used Grounded Theory, and the results showed new patterns of work that have evolved similarly across the workplaces we studied, offering new insight about how these workplaces might be better supported with technology. We particularly focused on designing for collaborative incident response work and its unplanned nature [12].

In a second set of studies, we focused on analyst teams conducting in-depth projects to explore specific issues of interest. Our main study in this work involved observing a team of 10 professional analysts over a 4 day project. The project itself was a proxy based on open data, rather than on real, and therefore sensitive data, but was designed by one of the senior analysts as representative of their real work. Our analysis of the data used Culture-Historical Activity Theory, especially the work of Engeström on collaborative work [19–21]. In particular, we used a model of collaboration based on Engeström's work, which identifies three types of collaboration: reflective communication, cooperating, and coordinating (in decreasing order of their strength) [21].

We will elaborate more on the second set of studies, because of the impact they had on later stages of our research.

The analysts' workflow, as they described it, and as we observed it, was as follows: Step 1: Reading background material. Step 2: Brainstorming issues. Step 3: Selecting most important issues from the top 30 issues. Step 4 & 5: Investigating issues. Step 6: Producing a 'bowl of knowledge' (multiple analyses, each one on an identified issue).

Our study found that in the early direction-setting stages of the analysis process (steps 1, 2, 3 and 4), the analysts collaborated closely by both cooperating and engaging in reflective communication. We illustrate this in Fig. 17.1.

However, after these initial steps which were completed on the first day, despite the fact the analysts were working on the same general topic, and using the same

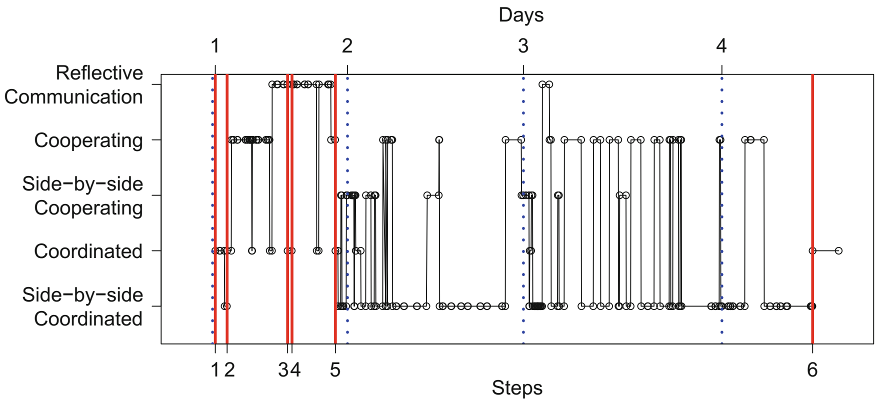


Fig. 17.1 The process of collaboration across time. *Solid lines* show steps in the analysts' process. *Dashed vertical lines* show days. The types of collaborative events are ordered from strongest (reflective communication) to weakest (side-by-side coordinated work). *Source* Brown et al. [11]

data set, we principally saw them working side-by-side and independently (again, see steps 5 and 6 in Fig. 17.1). The most common form of collaboration (cooperation) we saw in this phase was when one analyst requested technical help from another (this represented a significant number of the cooperating events seen in step 5). However, there were a couple of significant events of cooperation that were not of this type.

Significantly, at one point, one analyst produced a large poster showing results of her work, whereupon others were keen to comment and find connections to their own work.

We also observed another analyst applying a structured analysis technique called "Analyses of Competing Hypotheses" (ACH) which drew the attention of two other senior analysts who used their displays to pursue detailed investigations relating to this analyst's analysts. One of the displays used was a smartphone display and the other was a regular-sized display.

The collaborative work of the three intelligence analysts led us to explore ACH as an initial collaborative application for large displays. ACH is a technique developed by Heurer [29], and supported by software. The main idea is that an analyst considers several alternative hypotheses that might explain a set of evidence. They assess the data for credibility, relevance, and then consistency with each hypothesis. These factors are then used to build a mathematical model, which helps the analysts to avoid confirmation biases in favour of a stronger process that emphasizes disconfirming and eliminating a broad range of hypotheses.

Our analysis of the data, both observations and interviews, led to a number of interesting findings. One relates to "Process Productivity". As noted above, analysts collaborated more in early stages, and much less so later on. We observed that the early stages were done with whiteboards, posters, and brainstorming, where collaboration was explicitly supported. In the later stages there was much less support, and we felt that better support would facilitate and encourage more collaboration.

Another finding related to "Process Outcomes". This was related to the first finding, but we also realized that with only low levels of collaboration, the process involved little cross-checking, discussion of coverage, integration of knowledge, and comparison of results. We speculated that better support for collaboration would not only improve the productivity of the team, but also the quality of their outcomes.

Finally, we identified possibilities for better "Learning within the Process". In activity theory it is well-established that important learning occurs in cycles of externalization and internalization as team members interact. In the activity we observed, more support could have been put in place to increase the likelihood of individual and team learning, two secondary outcomes of strong collaborative practice. In the collaborative event we observed, very few team benefits ensued except when team members shared and reflected on their techniques during their presentations at the end of the project. There were also minimal individual benefits (although a few analysts learned new tools on their own, individuals seldom explicitly learned from each other).

Throughout the study, we saw the analysts use all kinds of surface-like artifacts, including whiteboards, posters, and notebooks, and well as certain software applications which ran on workstations set up with single or dual displays. Occasionally, we saw one analyst use her smartphone. Our conclusion was that there were important opportunities for surface computing to improve the collaborative analysis process. In particular, we felt that application software for large touch-surfaces might well support analysis techniques used later in the process, such as ACH. This kind of support might thus improve Process Productivity, Process Outcomes, and Learning within the Process.

17.4 ACH Walkthrough: Software to Support Security Analysis

In this section we report on the design and implementation of a surface application to support co-located security analysis. The field study of a team of security analysts suggested that the "Analysis of Competing Hypotheses" process (ACH) would benefit from collaborative support because the consideration and judgement would both be assisted by team discussion. We found calls for increased collaboration by authorities in the intelligence analysis world. Heuer and Pherson [30] suggest that their collection of structured methods can support collaboration and reduce cognitive bias. Hackman [25] concurs and emphasizes that collaboration both improves outcomes and contributes to the development of individual and team skills. Tools like ACH improve analysis work by reducing the impact of analysts' cognitive biases. ACH in particular is meant to reduce confirmation bias, where analysts will unknowingly focus on the evidence that supports their pet hypotheses rather than evaluating all evidence fairly.

We reviewed other versions of ACH software, namely the version developed for individual analysts at PARC [49], and two versions designed for collaboration, namely Globalytica Think Suite's Team ACH [24], and Open Source ACH [13]. We created extensive requirements for a collaborative version of ACH using surface technologies. The main requirements were that:

1. A collaborative version of ACH should enable part of a larger process where analysts alternate between individual work on an ACH and collaborative work on an ACH;
2. Analysts should be able to easily view evidence documents while working on an ACH analysis. We speculated that a mixed-display environment would support this best;
3. Collaborative ACH work should be enabled by a walkthrough process whereby members of the team take on roles that would strengthen the analysis, while they walked through all aspects of the analysis and checked or extended its content.
4. There should be built in support for exploring the strength of an analysis.

We focused on requirement 3 & 4. We saw the walkthrough support as an important part of the tool, since a fair number of users were new to ACH. Our requirements also introduced both a new collaborative practice as well as a surface application.

Fig. 17.2 ACH walkthrough in use: running with synchronized data on a large multi-touch screen, on a laptop computer, and on a tablet

Together these would aim to improve an ACH analysis by enabling face-to-face discussions about the attributes of the analysis, e.g., its completeness, its correctness, and so on. Our application software, "ACH Walkthrough" (ACH-W) accomplishes all these goals as a functional prototype. Figure 17.2 shows the software in use. The data set we used to illustrate the software is from publicly available material to investigate the collapse of ENRON Corporation [46].

While ACH Walkthrough can be used for ACH analysis generally, we especially intend for it to be used for a collaborative review, where a small team of analysts work together. In particular, we suggest an approach similar to that suggested by Wharton et al. called the "Cognitive Walkthrough" [63], where a team walks through steps, discussing and executing each step together, each team member contributing from their perspective. Recall that in our field study, we saw a need for *reflective communication*. We suggest that our walkthrough technique will provide strong support for reflective communication. In particular, when reflecting, analysts should discuss the overall direction of the work, the quality of the work, and the methods they are using to achieve their common goal.

As well as a collaborative review, we suggest that ACH analysis involves some work best done by analysts working independently. For example, this might be most appropriate for searching through documents and identifying evidence, and even for many initial assessments of credibility, relevance, and consistency with hypotheses.

Accordingly, we suggest that the best overall strategy for ACH is to alternate between independent work and collaborative reviews facilitated by ACH Walkthrough. The software allows analysis data to be transferred back and forth to spreadsheets.

The collaborative walkthrough is structured into a series of steps, where each is a step in an ACH analysis, together with discussion relevant to that step. To increase the value of the discussion, we suggest that team members adopt roles. For example, one analyst could play the role of a particular expert or organization, and represent that perspective in the discussion. This facilitates a diversity of perspectives in the discussion, and increases the possibility that critical issues will be identified. Heuer and Pherson [30] discuss the advantages of role-play in intelligence analysis, along with related techniques such as devil's advocacy and "red team" analysis.

In the walkthrough, our multiple device architecture also supports multiple perspectives on the data. As illustrated in Fig. 17.6, several devices can be used simultaneously with different views (each view is on a different 'tab' in a traditional tabbed display), and any changes made to the data are instantly synchronized. It would be possible, for example, to have two large touch displays, so that one could be used to consider consistency ratings (explained below), and the other could be used to browse related evidence documents. At the same time, individual analysts could check other tabs on the analysis using tablets or smartphones.

17.4.1 UI Design

ACH Walkthrough is a client-server web application, and it requires login with a userid and password on a project-by-project basis. Within a project, the software supports many ACH analyses, each with hypotheses, evidence items, and the scoring of these following Heuer's model. The UI presents several tabs, where each tab supports one functional aspect of the ACH process. We felt that a tabbed design was consistent with Heuer's step-wise process whereby the user's attention is deliberately tunneled through a structured process.

In addition to the basics of ACH analysis, the software provides several innovative features to leverage surface computing to support collaboration. These include large-scale touch controls, suitable for small groups, some innovative touch controls we call "fretboards", and a visualization technique called "parallel coordinates" applied to ACH data. We also provide "Walkthrough" facilitation to help groups systematically review an ACH analysis. Finally, we use an innovative multi-device approach which allows several devices to be used simultaneously.

Fretboards: In ACH, there are several steps that involve entry of a quantitative score: credibility and relevance of evidence items, and consistency of evidence with hypotheses. Instead of using numeric entry, we designed a new touch control we called the *Fretboard*. The name refers to the fingerboard on a stringed instrument, with lines that mark positions for certain musical notes. Our fretboards allow touch and drag interaction to position an indicator, showing the appropriate

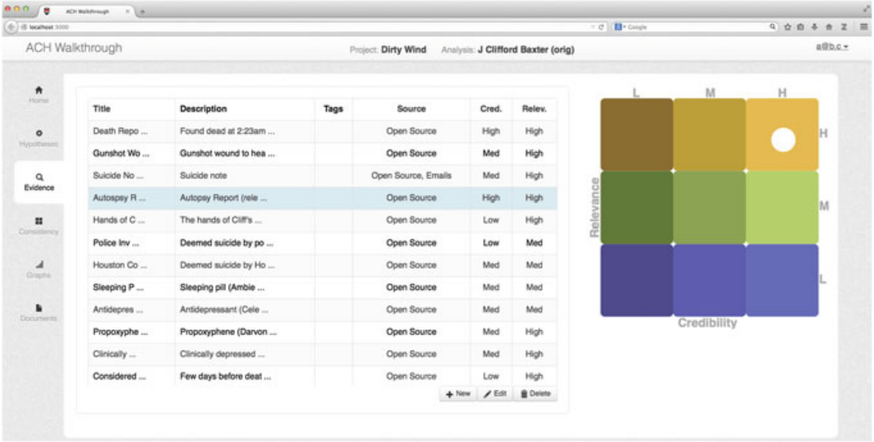


Fig. 17.3 Evidence tab: showing list of considered evidence with title, description and source. Touch control *fretboard* at *right* allows setting of credibility and relevance simultaneously

quantity along one or two frets/dimensions. This makes the entry highly visible to the group, allows spatial reasoning, and avoids use of the keyboard (see Figs. 17.3 and 17.4).

Walkthough Advice: In our field study, we identified a need to better facilitate strong collaborative activities such as those involved in joint review. To support this, we leverage ideas from a kind of software inspection technique called the "Cognitive Walkthrough" [63], hence the name of our tool being ACH Walkthough. The technique involves members of the group selecting roles to play in the review, and then the group stepping through the analysis together discussing each step. This supports a diversity of ideas, and avoids "groupthink". To support this, our tool has "walkthrough notes" that appear and give guidance, as seen in the lower half of Fig. 17.4.

Parallel Coordinates Visualization: In our field study and in later exploration of ACH analysis, we found that people wanted to consider the overall patterns in rating evidence for credibility and relevance, and in scoring of hypotheses for consistency. This enabled the analysts to assess the overall strength of their work thus far. To support this in our tool, we added a visualization of the ACH analysis using the visual formalism known as a "Parallel Coordinates". We considered alternatives [65], but settled on this visualization for its fit to task. Parallel Coordinates is an established visualization technique [34] to aid exploration of diverse data, and the technique has been advocated especially in the context of cybersecurity [16]. See Fig. 17.5 for an example.

Multiple Devices: One important collaborative characteristic in our software does not involve any specific element in the UI. By leveraging the MeteorJS automatic synchronization of data across connected clients, multiple screens/users are updated in near real-time, as illustrated in Fig. 17.6. This means that, at a meeting,

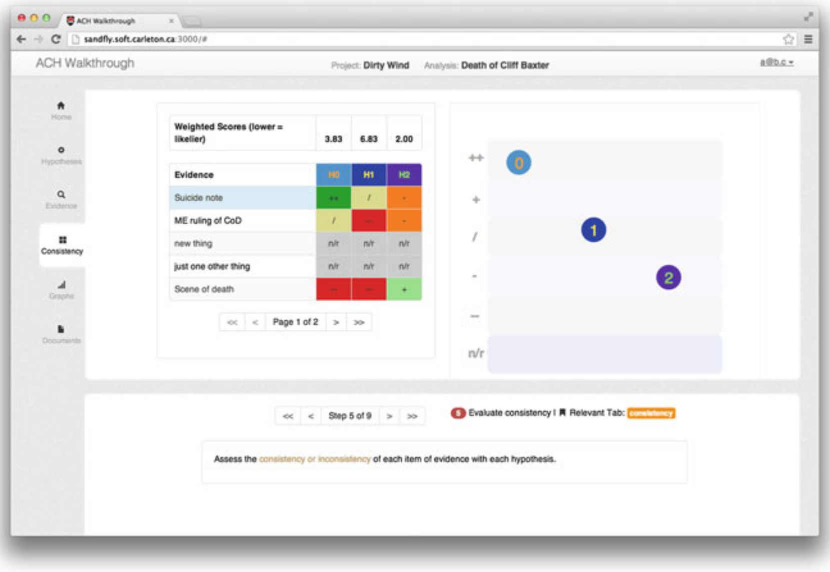


Fig. 17.4 Consistency tab: showing the main ACH table at *left*, where the *rows* show the evidence items, and the *columns* show the hypotheses. The touch control *fretboard* at *right* allows the setting of consistency ratings. The Walkthrough Advice Panel is shown for step 5

several screens can be used for the same ACH analysis, where changes made on any device are reflected on them all. Multiple large screens may be used, or small tablets. This facilitates parallel work in a collaborative context.

17.4.2 Software Implementation

A web-based approach was taken to enable deployment across many platforms with sufficiently powerful and standards-compliant web browsers, and without any need for complex software installation. As with most web applications, the overall system depends on a central server, with a certain amount of code loaded onto the browser (client) while the software is running. The ACH-W software relies, however, on processing that occurs on both the server and the client. This client application is delivered and updated without interruption or the need for client-side installation. In many cases the server can even be modified and restarted without the client application losing its place. This approach enables many useful features, such as no data being stored on the client machine when the program is not in use, and the ability for simultaneous use of the software for the same analysis by different devices, supporting remote collaboration.

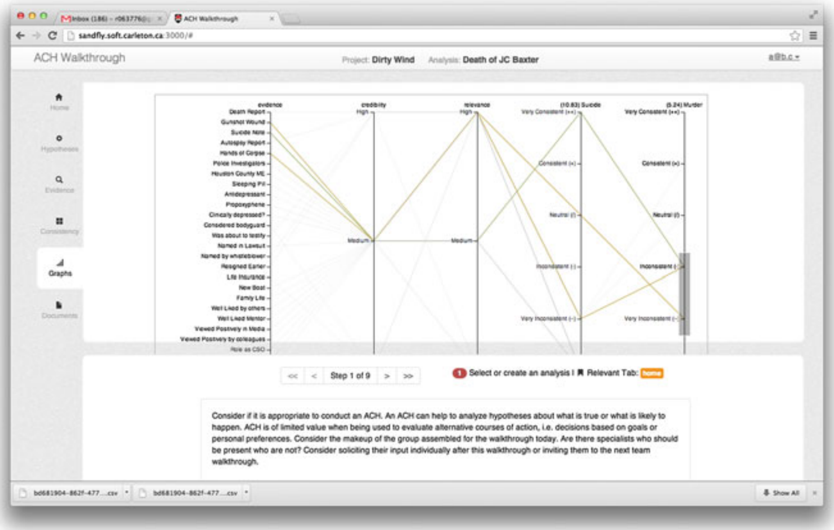


Fig. 17.5 Graphs tab: showing "Parallel Coordinates" graph visualization for the ACH analysis. Notice the selection box on the *rightmost* hypothesis axis, restricting selection to the evidence items inconsistent with that hypothesis. This was selected by "brushing" (a term defined by Becker and Cleveland [4] for selection of data within a visualization) that, in this particular case, supports Heuer's goal of seeking highly *diagnostic* evidence

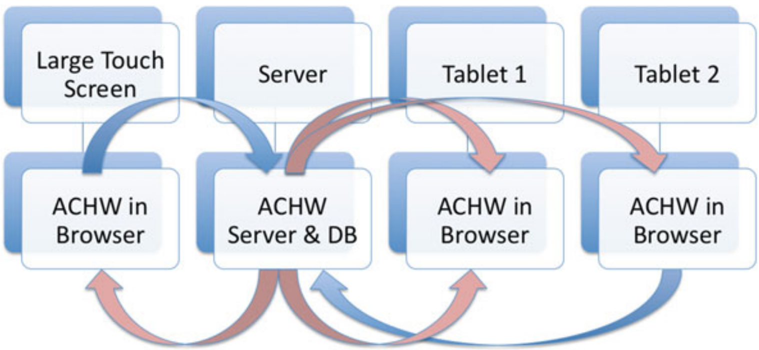


Fig. 17.6 Multiple device flow: in ACH Walkthrough, any number of devices of various kinds can be used to work on the analysis, and to make changes simultaneously and independently. The changes flow to the server, and thence to any other devices working on the same analysis

Our software was written primarily in the JavaScript programming language, using standards-compliant language software running in both servers and clients. We use several important external, but open-source, libraries in our implementation.

17.4.3 Evaluation

For our evaluation of ACH Walkthrough we had access to two vital resources. The first resource consisted of senior members of the group from the field study (e.g., the senior analyst who acted as the team's client), and the second resource was a panel of professors at an American university ('the demo panel') for whom the client requested we give an extended hands-on demo. Both groups provided extensive and helpful feedback.

One set of issues identified concerned the visualization elements in ACH Walkthrough, especially the parallel coordinates display, and the interaction afforded by "brushing" on the axes.

Figure 17.5 shows the first version of a plot for ACH-W, and an important issue should be immediately apparent. The problem can be seen when examining the number of lines between the first and second axes (left to right) and the apparent loss of detail as lines in subsequent gaps overlap. This problem results from the fact that the data points are not floating point values but instead are categorical (the first axis) and ordinal (the remaining axes). This loss of information can be corrected by using curved lines, as shown in Fig. 17.7.

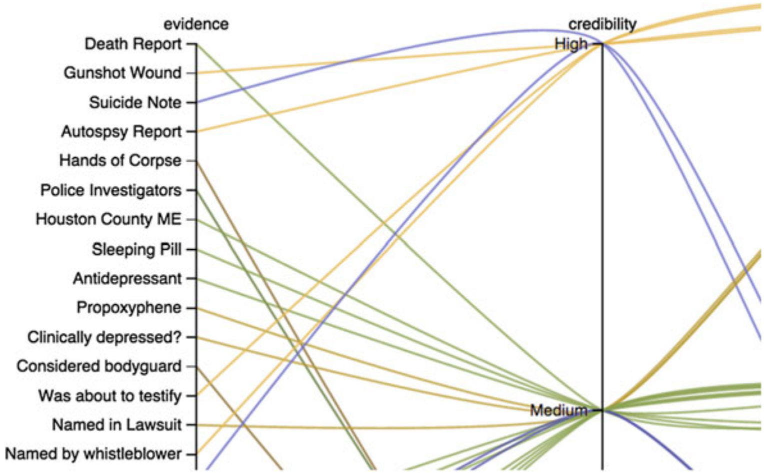


Fig. 17.7 ACH-W parallel coordinates showing improvement with *curves*

Brushing supports a surprising range of interaction tasks, especially as users become familiar with the meaning of the graph's dimensions. Users new to parallel coordinates graphs might at first be drawn to visual clusters and reinforcing trends across the display, and indeed in many domains this is the main strength of parallel coordinates. However, in the particular case of analysis work like ACH, the real power comes from drawing one's attention to *individual* evidence items that fall within meaningful regions of the graph and then taking the time to consider one's evaluations from fresh perspectives.

Interaction with parallel coordinates supports this kind of diagnostic reasoning by making it easy to select items that help rule out a given hypothesis. The user can create a brush that selects evidence items with ratings of *inconsistent* or *very inconsistent* along a particular hypothesis axis, which draws the analysts' attention to the evidence associated with only the highlighted lines. If the analysts had previously defined brushes for high credibility and relevance, they would quickly find evidence items requiring the greatest consideration.

Our informal usability sessions revealed opportunities for refinements of interactive features. Our internal testing using brushing and parallel coordinates had shown it offered powerful analytic value, but in user testing we learned that it does depend on some prior awareness of the brushing as well as a certain level of patience and attention to detail. We had assumed that most users would have encountered interactive displays in web forms, but for several users (particularly those new to parallel coordinates), the availability of brushing was not immediately obvious. It may have gone against their expectations if they assumed that the visualization was merely a static aggregation of data.

Without cues from experienced users, our testers did not attempt to apply any brushes. In our current implementation of ACH-W there aren't any obvious interaction cues for newcomers. In fact there is only one type of discoverable affordance and it is offered to mouse users when hovering the pointer over an axis. Unfortunately this feature assumed that hovering could even take place. Users of touch interfaces lack the ability to hover, and so they miss out on interaction cues altogether.

This issue became apparent through a usability test where the participant was helpfully thinking aloud and found himself stuck on one of the walkthrough steps. It was only the novelty of the technique that caused a problem. Once he was shown the availability of the brush feature, its meaning was readily apparent.

However, even when users understood brushing, they did not immediately grasp its ability to help seek evidence that could *disprove* their favourite hypothesis, a task that is central to reducing cognitive bias. One possible enhancement for first-time users might be to introduce the feature of brush-based filtering by offering a list of pre-set selections based on ACH-specific tasks (e.g. filter irrelevant items, confirm diagnostic items for hypothesis n, then $n + 1$, find counter-evidence for hypothesis n, etc.) and then instruct the user to walk through each of these presets. Also, the initial rendering of the parallel coordinates graph could briefly show animated selections on each axis that quickly unfold until they encompass their full range and then leave behind affordances for the user to adjust (see mockup in Fig. 17.8).

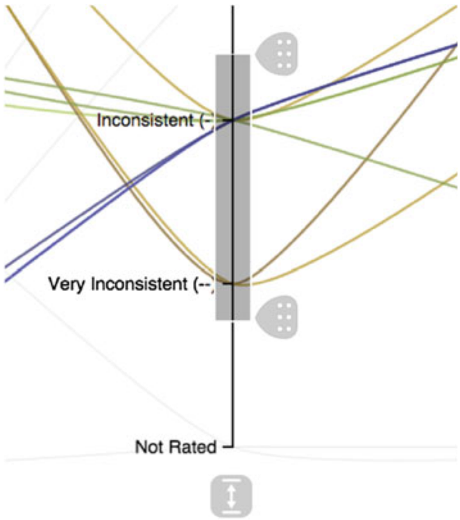


Fig. 17.8 Mockup of three possible affordances indicating touch points. With two of them the brushed area can be extended

Finally, a number of senior researchers from the demo panel expressed concerns with the process of ACH in its present form. Their concerns fell into two broad categories: psychological, especially whether ACH avoids cognitive bias, and mathematical, about the nature of the underlying mathematical model. These issues are both interesting, but as they do not relate specifically to our software, we will not elaboarate further.

Comments on the software features of ACH Walkthrough, however, are our concern. One commenter was concerned that the two digits of precision used in presenting scores against hypotheses in the Consistency Tab and the Graphs Tab might mislead the user into perceiving a mathematical distinction between closely ranked scores. The scores use a simple formula developed with Heuer in the construction of the Xerox PARC version of ACH, and we chose to reproduce this formula. Future versions will represent a more coarse representation of the overall score, or may eliminate the numeric score entirely. They might instead use a visualization that fosters appropriate attention to the similarity, rather than the minor differences, between hypotheses.

Similar to this concern was a comment on the immediate feedback of the change in score provided while manipulating the ratings on the Consistency tab. The reviewer believed the immediate feedback might actually encourage confirmation bias rather than fight it. This was an interesting concern that could form the basis of a future experimental review. Design of such an experiment could prove difficult to achieve however, particularly given the various other natural sources of confirmation bias present. It would also be difficult to produce a baseline from which to establish the presence of an effect. This was left as another potential avenue for future research.

Overall, our experience with ACH Walkthrough was positive, but the interaction design and software are still at the stage of functional prototype. The next steps should be a more controlled usability study, ideally with professional analysts, and a real problem suitable for analysis. At the same time, our early feedback was often accompanied by suggestions for new features. The most commonly requested features are in the list below. The first two items on this list inspired the next project in our work, which we present in the next section.

1. Versioning and merging of versions to support returning to a previous state and merging security analysis work
2. Roll-back and play-back functions to support asynchronous collaboration and exploration
3. Improved support for integration with external data sources to support evidence gathering
4. Bidirectional links between evidence and precisely tagged supporting documentation to support evidence browsing
5. Voice input for hypotheses and evidence to ease the burden of typing
6. Colour customization for rating system (from a colour-blind evaluator)

17.5 Ra: Support for Application Interaction History

In the previous sections, we have described how our field study suggested that collaborative security analysis would be assisted by large surface tools, and we then presented such a tool, ACH Walkthrough. Both when observing usage of ACH Walkthrough, and when seeking feedback, two additional features seemed especially worthwhile exploring: versioning and merging of versions, and roll-back and play-back functions. We therefore set out to explore how such features might be provided. We developed an add-on for web applications, such as ACH-W, to support interaction history, and present our prototype, Ra, in this section.

While tools like ACH are designed to address issues such as confirmation bias, another cognitive process that can interfere with effective analysis is *satisficing* [55], in which an analyst will stop when they have reached an answer that seems "good enough". On its own, this is rational and acceptable as long as the threshold is set right. The problem is that software may impose additional costs to further exploration—at worst, further exploration requires starting all over again—and that lowers the "good enough" threshold. This is related to the problem of *premature commitment* from the Cognitive Dimensions of Notations framework [5]. If an analyst has reached a solution but wants to try something else, they must decide whether it's worth the effort to just get back what you had if the "else" isn't any better. Without a system for storing interaction history, the user is constrained to repeat the steps to achieve the old solution, or else execute the inverse of all actions taken since then. This may be a significant impediment to exploration.

All users of complex software make decisions that they may later wish to change. Software can support this need to revisit past decisions by keeping past versions of the application's state that the user can go back to. There are several mechanisms for maintaining and presenting this history. Since early in the history of desktop computing in the 1970s and 80s, most user applications have provided users with an "undo" command to revert the most recent change. But not all uses of "undo" occur because of mistakes. Kirsh and Magli [39] divide (non-erroneous) user interactions into two categories: *pragmatic* actions are those that actually move the user closer to their goal (e.g., error recover, recall or recap) and *epistemic* actions are those that help the user learn about their situation, exploring to gather information that is either "hidden or hard to compute mentally". So interaction history systems should be designed to support epistemic interaction as well as pragmatic actions such as error recovery.

Touchscreens make epistemic interaction more compelling. Lee et al. [41] argue that touchscreens enable a kind of directness that is even more immediate than the Direct Manipulation described by Shneiderman [54], since Shneiderman was assuming the use of a mouse and keyboard. Large touchscreens also enable new kinds of co-located collaboration possibilities [9]. Sharing a touchscreen is much easier than sharing a keyboard and mouse. Touch interfaces are changing the kind of software we make, and the new types of applications need to support epistemic interaction.

We wanted to develop a system to provide users with access to all their historical interaction states, including those that would be discarded by a traditional stack-model undo system. Such a system should encourage more epistemic interaction by allowing users to return to known good states after exploring and to reduce premature commitment and the urge to satisfice by freeing users from the risk of losing good work while investigating other options. We want to make software tools that better support data analysis and other kinds of nonlinear tasks that are hard to automate; we want risk-free exploration.

In furtherance of these goals, we developed a prototype library called Ra, pictured in Fig. 17.9.

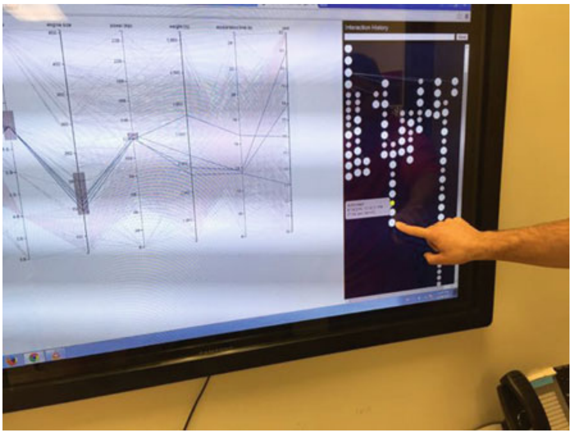


Fig. 17.9 A study participant using our prototype software Ra (the *dark sidebar*) with an interactive data analysis tool

17.5.1 Visualizing History

Software has many different methods for handling interaction history and exposing it to users. We reviewed a large number of approaches, but the ones that seemed most general were those from software source code version management. Some modern version control software, such as Git and Mercurial, store the history of a project as a directed acyclic graph (DAG). Tools for working with these systems will often display the history as a DAG as well. In software projects, the structure is a DAG because most work ends up in the final product. When developers work in parallel, they are usually not working on alternatives; they will merge both lines of development together.

In Ra, we represent the history as a tree. (This is also the data structure used internally.) We expected that the exploratory behaviour we want to encourage would result in mostly dead ends, or multiple different results for presentation or comparison, rather than some unification of most of the work. Merges seem like a desirable feature in some cases, but their usefulness may not be worth the extra complexity. It is unlikely that Ra could perform merges automatically, and there is no obvious way for a user to direct the merge of two snapshots of internal application state. This is in contrast to merging source code, which can often be done automatically, whereas merging application state would require understanding the (usually not human-readable) representation of that state.

The tree visualization in the Ra sidebar is inspired by the visualizations for version control systems, and the traditional visualization of trees in computer science. New nodes are added below when the state is changed (by say, a user interacting with the web application). New nodes are added to the right of the parent node if the user wishes to explore alternative actions from the parent state. The parent node represents the previous state.

17.5.2 Implementation

The general technical goal is to capture snapshots of the running state of a Web application, and then be able to load snapshots without too much delay. There are several ways this could be accomplished, each with its own drawbacks.

For ease of prototyping, we chose to implement Ra as a JavaScript library, to be included in the Web application with some (but preferably minimal) supporting application changes. We wanted Ra to be non-invasive enough that it can be added to an existing application without restructuring the whole program.

The central part of keeping required changes to the host application localized is the use of *Proxy* objects. The newly-finalized ECMAScript 2015 Language Specification [18] (ECMAScript 6) introduces them, though prominent JavaScript engines such as SpiderMonkey in Firefox [44] implemented versions specified in drafts of the specification well before the final publication. A Proxy imitates an existing object, and it can intercept almost all interaction with that object. In the specification [18], a

Proxy object is defined as an "Exotic Object", meaning that it is not required to display normal JS object semantics. For example, immediately after setting a property on a regular object, retrieving the same property must return the previously stored value (unless an exception was raised); Proxy objects are not required to act this way. The Proxy object has a special handler function that can override the normal object semantics. Following the same example, retrieving a property value as `obj.prop` would call a function provided when the proxy was created, and the expression would evaluate to the return value of that function. The function can usually return any value it chooses, although there are some more complicated edge cases requiring the semantics of certain features such as non-writable properties to be respected [18, 45].

This general approach to program augmentation is based on much earlier work by Noble, Biddle, and Groves [47].

An application using Ra substitutes Proxy objects created by Ra for the objects that hold its state. When all objects that hold state in the application are actually Proxy objects managed by Ra, the application code continues to interact with Ra implicitly when it uses those objects, yet all other code can continue to use the objects as if they were the real state objects. This allows us to update state objects on demand, wherever they may be inside the application at the time, and whoever may have references to them. From the perspective of the application code, when the user loads a saved snapshot, the state objects immediately become the saved values, without requiring the application to actually make any changes. This is accomplished by setting all the traps in the Proxy to return the value from the current saved state object instead of the original. If the application was already written in an object-oriented style following the Model-View-Controller (MVC) pattern [40], with state stored as properties of long-lived objects, then the state objects do not have to be tracked through their entire lifecycle; to support Ra, changes are needed only where state objects are *created*. Additionally, since the "objects storing application state" that Ra needs correspond to objects in the Model component of MVC, all the state objects are already identified and ready to be replaced by proxies.

Our proxy-based approach corresponds very nicely to traditional MVC or three-tiered application architectures, since the Model component keeps the state objects isolated from the other code. However, the state object requirements are impractical in some programming paradigms and architectural styles (or lack thereof) used in JavaScript. Storing state in the web application domain object model (DOM) is a common technique that is problematic for Ra. Trying to recover the important state from the DOM from Ra's position would be complicated and error-prone at best. Moreover, some applications keep state in closures, in variables local to a function but available to any other functions that are lexically inside the function. Programs written in the functional paradigm generally rely on this rather than mutating objects. It is common to use closures to avoid adding properties to the global object in top-level code, and some store state in variables in that scope. There is also a well-known pattern for "private members" in code trying to emulate Java-style object-oriented programming by using closures to restrict access to variables, since variables cannot be updated from outside their scope.

One test we conducted with Ra involved a parallel coordinates application similar to one of the design elements in ACH Walkthrough. We found that the parallel coordinates code presented several of the problems we anticipated. The code uses the D3 framework [8], which maintains listeners on the state objects in the model component, and the code was written in a partly functional style with significant state variables in closures. Restructuring the application to meet Ra's requirements would have been a large undertaking comparable to rewriting the application.

We developed a mediation mechanism to allow an application to use Ra without significant restructuring when it can't meet the state object requirements. It puts more responsibility for managing state on the application, so this may be of limited practical value in an application with complex state. However, it was sufficient for the parallel coordinates application. The Ra API is extended to include *priests*, which are special objects provided by the application that act as interpreters between Ra and the application state. Objects that store state but do not meet Ra's requirements are still marked with a call to ra.stateObject on creation, but a priest name can be supplied as well, as in the call `ra.stateObject(b, "brush")`. Instead of returning a Proxy object wrapping `b`, this will return `b` itself. Ra will then delegate responsibility for monitoring, saving, and restoring that object to a priest registered with the given name.

17.5.3 Ra User Experience

When Ra is part of a Web application, it adds a sidebar, shrinking the available application space. Ra does not try to intercept or manage user interaction with the application part, so aside from being narrower, the application works exactly as it would without Ra.

As the user uses the application, Ra records the state of the application when it changes. We call the recorded state a "snapshot". These are shown as nodes in the tree visualization in the sidebar, where each snapshot follows from its parent in the tree, and later nodes are shown below earlier nodes ("time flows down").

The user can return the application to a previous state from the sidebar. Tapping or hovering with the mouse brings up a balloon popup for each node, as shown in Fig. 17.10, from which the saved state can be loaded. The node in the tree that represents the current application state is marked in yellow. When the user returns to a previous state, they can still interact with the application—making different decisions this time. Instead of replacing the history from that point forward, as a traditional undo system would, Ra starts a new branch in the tree, as shown in Fig. 17.10, so both timelines are available.

Ra can record states for different reasons, and these get different glyphs for the nodes in the tree visualization. There are three types of node, as illustrated in Fig. 17.11:

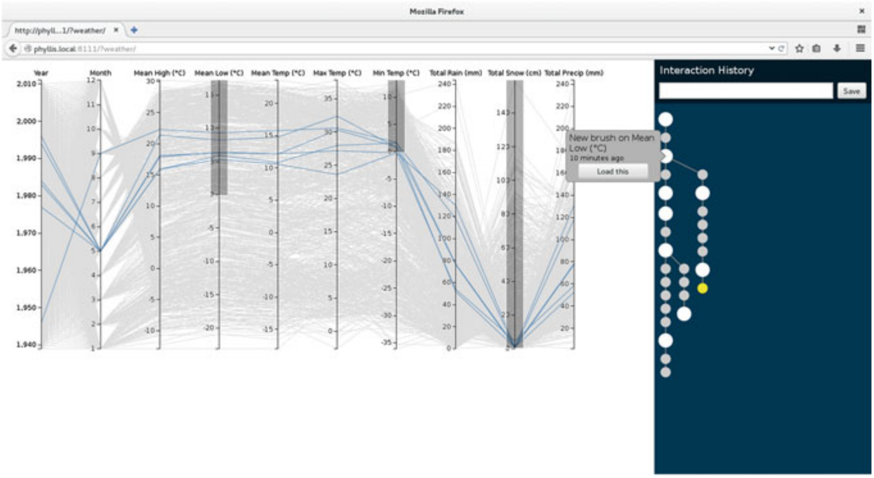


Fig. 17.10 The parallel coordinates application (showing the months Ottawa had at least some snow and where the temperature never dropped below zero—a rare occurrence). The balloon popup for a node is showing the label and timestamp. The user has already returned to that state and started a new branch; the "Load this" button would let them do so again

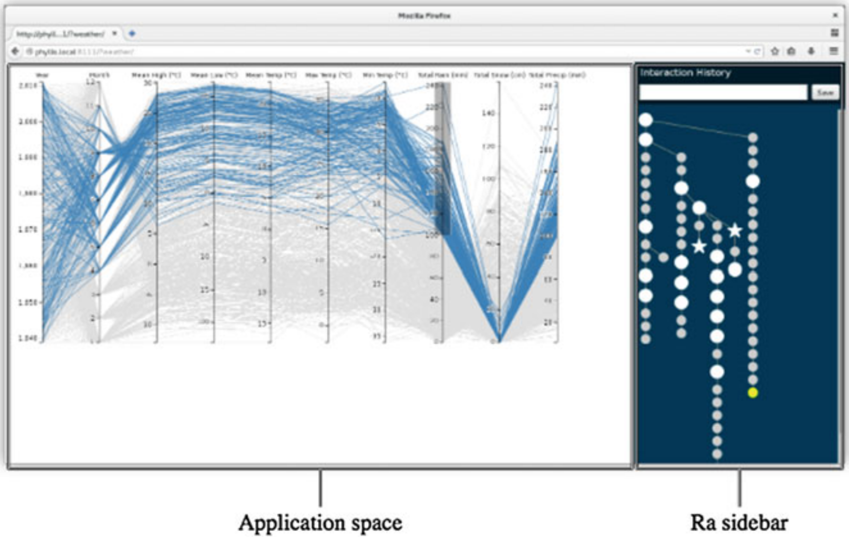


Fig. 17.11 Ra divides the page to make room for a sidebar

- Automatic snapshots are shown as small dots. They are all given the same default label ("autosave") because no semantic information about the state is available. Ra creates them automatically when it detects that the state has changed, though this is rate-limited to at most one per second, and the other snapshot types supersede automatic snapshots.

- App-suggested snapshots are shown as bigger, brighter dots. The application can tell Ra to create one of these snapshots when it is in a state the user is likely to want to return to, which is why these nodes are more prominent. The application provides the label for these snapshots. This kind of snapshot depends on the support of the application, and some applications may not create any. Our parallel coordinates application uses these to mark the creation of new brushes (selections).
- Starred snapshots are shown as stars. They are created explicitly by the user. They may have a label provided by the user. To make a starred snapshot, the user enters the desired label in the textbox in the sidebar, then presses the "Save" button. Note that in previous versions of Ra, starred snapshots had the same appearance as app-suggested snapshots.

We performed a usability evaluation in which participants used a simple puzzle and an interactive data analysis tool with Ra. Our experience implementing Ra and our observations from the study revealed several important themes. Perhaps most interestingly, we observed three kinds of history tasks. This categorization is not directly about the user's intent, for which there would surely be more than three categories, but the relationship between the state the user was in (old) and the state to which the user went (new). The three kinds of task we observed were "oops-undo", "undo-retry", and "undo-review-redo".

Correcting mistakes In the "oops-undo" case, the user has made a mistake recently, or tried to perform an action but the computer did something unexpected. The old state was clearly wrong; the user does not expect to need it again, and perhaps it should be hidden from view. This is the case for which the traditional undo was designed for, and it is reasonably well-suited to it.

Trying alternatives In the "undo-retry" case, the user wants to try some alternative, usually starting from further back in history than the oops-undo case, or from a parallel timeline. New work will be based on the new state, but the user may not be certain that the old state should be discarded; the old state may still be useful.

Ra was intended to support this task in particular. Traditional undo mechanisms force the user to give up one branch to work on another, which requires the user to commit to a decision before they see the result; they may have to resort to manual version control (saving the file separately for each experiment) or make a decision with incomplete information. Ra allows the user to keep any number of parallel alternatives without the extra costs of saving and managing alternatives in files.

Comparing versions In the "undo-review-redo" case, the user just wants to *look at* a previous version of their work. It might be to compare two alternatives, or to copy a particular piece of a previous solution, or even to remind themselves of what *not* to do. The old state is still the working copy where new editing work will happen; the new state is not something the user wants to keep.

Using traditional undo for this task is particularly dangerous because any accidental edit will discard the redo stack, leaving the user in a state they intended to abandon. Ra supports it better, since all versions are accessible, but there are features that could improve the experience, such as some way to keep track of the current *working* branch separately from the version being viewed. We did not think of this task when initially designing Ra, but our user studies suggest this should be better supported.

Reviewing sequences of past states without editing them may also help other people understand the final state. For example, Farah and Lethbridge [22] developed a linear timeline for reviewing the development of software engineering models. In the field of intelligence analysis, the system could be used for a kind of traceability, allowing analysts to review the decisions that led them to a conclusion. Ra could emphasize this capability by making it easy to explore the path from a state to the root of the tree—that is, the work that went into a selected state, ignoring parallel timelines.

Overall we were satisfied that our prototype implementation of Ra demonstrates utlility with strong potential for supporting security analysis. Key concepts in sensemaking are exploration and iteration, and avoiding bias that can easily occur when satisficing.

17.6 Strata: Web Application Annotation

In several steps of our research, we have observed professionals in co-located collaborative security analysis work. One common kind of behaviour is annotation, whether of documents, or on whiteboard diagrams. Where an application was displayed on a large screen, whether television or projection, it was common to see people using paper or whiteboards to make sketch duplications of key parts, and then annotate these. Often users tended to point and gesture to elements on the screen, as if they were marking up the content on the display itself. This behaviour is well-understood, and not only supports note-taking for future reference, but also constitutes "cognitive tracing", whereby people make notes to help them think about the task at hand while it is underway [53].

We therefore decided to explore explicit support for this behaviour, and developed an add-on for web applications to support annotation and easy screen capture. This would allow the users collaborating over the display to annotate the web application which they are interacting with, in addition to saving and retrieving previously saved annotations. In this paper we present the technology choices and interaction design of our prototype,"Strata', see Fig. 17.12.

The value of markup on documents has been included in various contexts including in modern PDF reader applications such as Adobe's Acrobat and Apple's Preview which provide users with annotation capabilities on PDF documents.

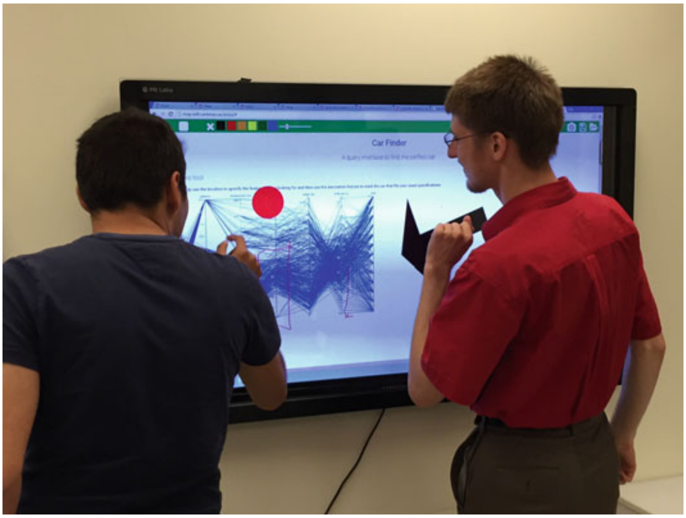


Fig. 17.12 Strata: co-located users collaborating over a large touchscreen display using Strata to mark up a sample “car finder” web application

Annotation capabilities have also been included in Microsoft’s Edge browser in their Windows 10 operating system, which provides markup tools for web pages such as free hand drawing, highlighting, and text based notes. There are also various web browser extensions which allow the user to mark up web pages, such as the Hypothesis extension [32].

Annotations have also been the subject of various studies in academia. Denoue and Vignollet proposed a very simple implementation by storing annotations on the client using extended URLs and avoiding the server all together [17]. Alternatively, Sodhi Chatti describes a “transparent whiteboard” overlay approach to creating and storing annotations; the annotations would be formed so that it is self-contained which would therefore allow the annotations to be stored anywhere (either the server or client) [15]. Finally, Beryl Plimmer explores putting the web page into an iframe and then overlaying Adobe Flash-based annotations on top of the frame and tagging annotations with metadata associated with the user. The annotations are then stored in a database, retrieved at anytime, or shared with various users [51].

In summary, earlier work focused largely on the value of annotations on documents using a mouse-driven interface and regular computer monitors, while our focus is on the value of annotations on web applications (and not documents) as a mechanism for facilitating collaboration amongst users over large touchscreen based displays. The importance of annotations for interactive systems has long been suggested by Thomas Green as “secondary notation” in his identification of “cognitive dimensions” of complex systems [6]

Our goal was to implement a JavaScript based add-on that can be included in any web application to provide users with mark up capabilities. Therefore when we developed the prototype of the Strata system, we decided that the project should meet the following requirements:

1. The system should enable creating annotations on top of web applications.
2. Annotations generated should not obstruct the content on the display. The content should remain accessible and manipulatable even if there is markup overlaid on top.
3. Both touch and mouse based input should be supported by the system, since there may be times when users would prefer using a mouse even on a touchscreen computer.
4. Multitouch drawing capabilities and gestures must be included in order to enable multi-user collaboration.
5. The system should leverage multitouch gestures in order to ease usability and facilitate exploratory interaction within the system.
6. The system should be able to save and load annotations so that users can revisit them at a later time or share them with other users.
7. The system should integrate seamlessly with a web application without requiring several re-writes by a web application creator wishing to include the system.

Strata was designed as a library that could be added on to any web applications, rather than creating an extension which requires users to modify their web browser. This allows web application developers to easily integrate the Strata system with any existing web application in order to gain access to markup capabilities and enhance the collaborative aspect of their web application.

Initial prototypes of Strata were developed using HTML Canvas. However, this technology was abandoned in favour of Scalable Vector Graphics (SVG, also in HTML5), because the HTML5 Canvas element would overlay over the content and would thus prevent the user from interacting with the elements underneath. Our design is that users can toggle between interacting with the application itself, or with the annotation as a "layer" (hence the name "Strata").

Since the system is intended to work on large touchscreen devices, the system supports a multitude of features including multi-touch input and gesture recognition. Strata is designed with both of those features in mind, it leverages the Interact.js library, which provides unified mouse and touch events thus allowing for development on both touch screens and mouse based personal computers. In order to support multitouch, Strata leverages Interact.js's [1] "pointerId" attribute to assign a newly created pencil object to each finger thereby mapping each pencil object to a finger therefore allowing for drawing using multiple fingers. Multitouch is essential not only because a single user would expect it, but also since the system is intended to be used on large touch screens, it would likely be used by multiple simultaneous users. Another advantage of using Interact.js is that it provides support for gestures; any object with that matching class would recognize gestures, including pinching to resize an element, and a rotate gesture, both of potential use in annotation.

To demonstrate Strata and conduct preliminary usability testing, we applied it to a simple car finder web application, which allows users to explore choices for cars

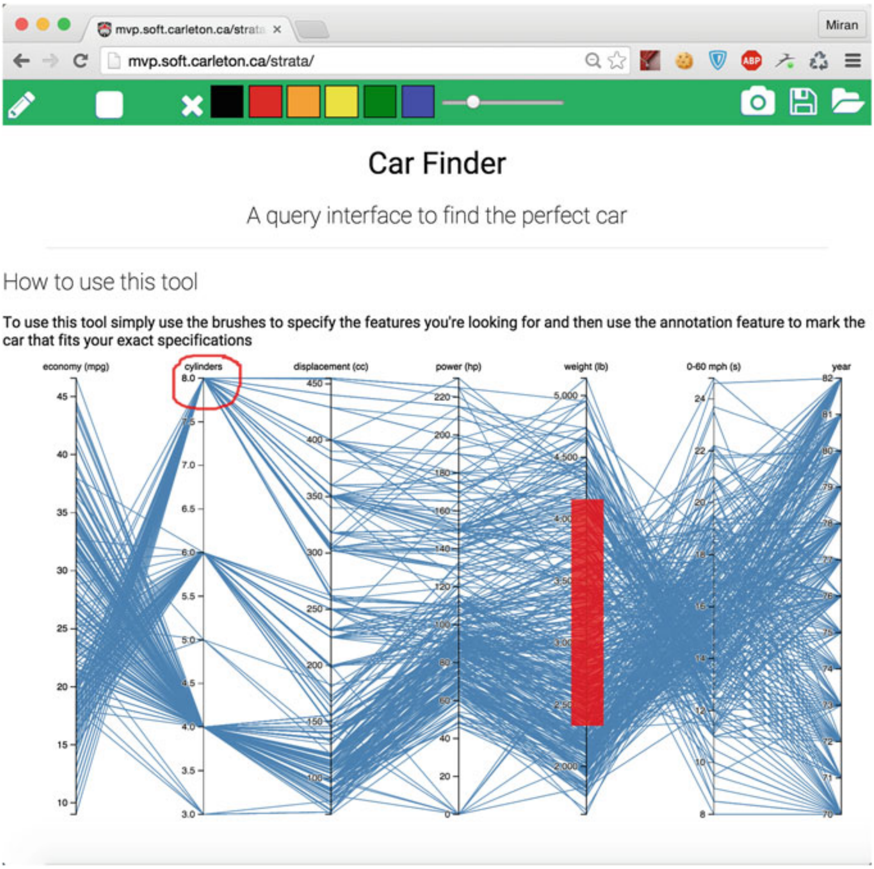


Fig. 17.13 Annotation: Strata annotations have been used to mark up the sample web application using freehand "pencil" drawings and a rectangle shape

based on economy, power, etc. This application again uses parallel coordinates visualizations [33] and is based on the D3 visualization library [7, 28] we used in ACH Walkthrough. This visualization shows data attributes of cars on several parallel axes and allows individual elements to be selected by "brushing" on the axes.

The application interface is shown in Fig. 17.13 where the car finder is in the main part of the screen and the Strata add-on interface is shown as a tool bar across the top. The toolbar contains the pencil tool and options for stroke and colour.

Document annotation tools such as those found in PDF viewers provide a similar interface for interaction. Users can begin interacting with the Strata system by pressing the pencil icon to enable drawing mode. Once the drawing mode is triggered, the web application will no longer become accessible so that the users can mark up the screen without fear of accidentally selecting text or interacting with the underlying program. The web application's functionality can be resumed once the drawing mode is disabled.

Once in drawing mode, the user has various options on the toolbar including clearing the paper, setting the colour and the stroke size options for the freehand drawing "pencil" tool, or selecting a shape such as a rectangle. These options can also be changed at any time by using context menu options.

The Strata add-on provides users with various ways of annotating web applications. They have access to both freehand drawings as well as shapes, including arrows, rectangles and ellipses. A context menu is used to create new shapes as demonstrated. Since it is a contextual menu, it changes the options as appropriate to the shape, based on where it was triggered. Triggering the context menu over an element will bring up the styling options for that particular element, otherwise triggering it over the web application will bring up element creation options which allows users to add rectangles, circles and so on.

The Strata toolbar presents users with various options, including saving the annotations as a JSON file. The resulting JSON file can then be loaded at any later time by invoking the load function through the Strata toolbar or shared with other users who can then load the annotations, view them and possibly add or remove elements from them. The system also includes the ability to save a screenshot of the annotations which can be invoked by clicking the camera icon in the Strata toolbar. Once the screenshot functionality is invoked, a screenshot of the web application (including any markup) will be taken and sent to a private image gallery.

One final design feature we implemented was to explore the possibility of "semantic annotations". By this, we mean a mechanism that can allow applications to present hooks so that Strata can do smart annotations using those hooks (delegating the markup to the web application). In this way, for example, a rectangular element in the application interface can trivially be annotated with a transparent but coloured rectangle to highlight the element. Similarly, circular elements could be detected and annotated with highlighting circles, and writing could be highlighted following the flow of the lines of text.

Future work on the project includes formal usability testing in an ecologically valid context where we have people collaborating on real work using the system. We also believe that it would be important to investigate the value of a web extension-based architecture in the future in order to be able to utilize the Strata overlay in any web application without the developers having to include support themselves.

17.7 Discussion and Conclusion

In this chapter we have reviewed a number of our projects on surface computing for security analysis. We began with a survey of related work, and then conducted field studies. We developed ACH Walkthrough, a surface computing application to support analysis work, and then two add-ons, one to support interaction history, and another to support annotation. This work was carried out over several years, and involved several projects we did not address in detail here.

Reviewing the work as a whole, several themes stand out. One suggested by our literature review, and confirmed in our field study, is simply that large surfaces, whether whiteboards, posters, or large computer displays, really do facilitate collaboration. Small surfaces are hard for multiple people to see, and are perceived as personal, making joint use seem invasive.

A second theme is more specific to security analysis. The work involves large amounts of data that is typically incomplete, unclear as to relevance, and can even be intentionally deceptive. Yet making determinations and recommendations must still be done, because security always involves risk. Together, this has led to analysis processes that have several kinds of filtering, assessment, and iteration, for example as described by Pirolli and Card [50]. Our field study of professional analysts suggested that this process would be improved in several ways by better collaboration, for example using large surfaces. This is also consistent with results found by Isenberg et al. [35] in their study of students doing intelligence analysis. However, we also learned that it was unrealistic, and almost certainly unhelpful, to expect analysts to work in close collaboration all the time. Much of the work required intense focus and concentration, and was best done alone for periods of time.

In our work developing and testing our surface computing tool for security analysis, ACH Walkthrough, a cluster of themes emerged. One was the importance of guided collaboration, where our walkthrough steps helped users follow the ACH process. Another was that it became clear that the work involved "epistemic " interaction. This has been identified by Kirsch and Maglio [39] as supporting speculative actions, done to explore possibilities. We realized this was the principle underlying our fretboards and parallel coordinates visualizations. At the same time, we appreciated the need for analysts to take away results, work alone, and bring back new ideas.

All this led to the identification of new ways to better support this kind of work. Interaction history support, such as provided by Ra, can help epistemic interaction because it frees the analysts to explore alternatives, while allowing easy return to previous states. Annotation of application states can be supported by software like Strata, which allows collaboration around application software, while making notes on the results for later review.

In summary, we found that surface computing has a strong relevance for security analysis, especially in how it can support collaborative epistemic interaction, and this can be improved by support for guidance, interaction history, and annotation. These are promising new directions for software design.

References

1. Adeyemi T Interact.js. http://interactjs.io/. Accessed 21 Aug 2015
2. Andrews C, Endert A, North C (2010) Space to think: large high-resolution displays for sensemaking. In: Proceedings of the 28th international conference on human factors in computing systems. ACM, New York, NY, USA, CHI '10, pp 55–64. doi:10.1145/1753326.1753336
3. Anslow C, Marshall S, Noble J, Biddle R (2013) Sourcevis: collaborative software visualization for co-located environments. In: 2013 first IEEE working conference on software visualization (VISSOFT). IEEE, pp 1–10
4. Becker RA, Cleveland WS (1987) Brushing scatterplots. Technometrics 29(2):127–142. http://www.jstor.org/stable/1269768
5. Blackwell A, Green T (2003) Notational systems–the cognitive dimensions of notations framework. HCI models, theories, and frameworks: toward an interdisciplinary science
6. Blackwell AF, Britton C, Cox AL, Green TRG, Gurr CA, Kadoda GF, Kutar M, Loomes M, Nehaniv CL, Petre M, Roast C, Roe C, Wong A, Young RM (2001) Cognitive dimensions of notations: design tools for cognitive technology. In: 4th international conference on cognitive technology: instruments of mind. Springer, London, UK, UK, CT '01, pp 325–341. http://dl.acm.org/citation.cfm?id=647492.727492
7. Bostock M Parallel coordinates. http://bl.ocks.org/mbostock/1341021. Accessed 21 Aug 2015
8. Bostock M, Ogievetsky V, Heer J (2011) D^3 data-driven documents. IEEE Trans Vis Comput Graph 17(12):2301–2309
9. Brown J, Wilson J, Gossage S, Hack C, Biddle R (2013a) Surface computing and collaborative analysis work. No. 19 in Synthesis lectures on human-centered informatics. Morgan & Claypool. doi:10.2200/S00492ED1V01Y201303HCI019
10. Brown JM, Wilson J, Gossage S, Hack C, Biddle R (2013b) Surface computing and collaborative analysis work. Synthesis lectures on human-centered informatics. Morgan & Claypool
11. Brown JM, Wilson J, Biddle R (2014) A study of an intelligence analysis team and their collaborative artifacts. School of Computer Science Technical Report TR-14-04. Carleton University
12. Brown JM, Greenspan S, Biddle R (2016) Incident response teams in it operations centers: the t-tocs model of team functionality. Cognit Technol Work 1–22
13. Burton M ACH:A free, open source tool for complex research problems. http://competinghypotheses.org/. Accessed 03 Sept 2014
14. Buxton B (2011) Integrating activity theory for context analysis on large display. In: Human input to computer systems: theories, techniques and technology. Available online, chap 4. http://billbuxton.com/inputManuscript.html
15. Chatti MA, Sodhi T, Specht M, Klamma R, Klemke R (2006) u-annotate: an application for user-driven freeform digital ink annotation of e-learning content. In: Sixth international conference on advanced learning technologies, 2006. IEEE, pp 1039–1043
16. Conti G (2007) Security data visualization: graphical techniques for network analysis. No Starch Press
17. Denoue L, Vignollet L (2002) Annotations in the wild. In: ECAI 2002 workshop on semantic authoring, annotation and knowledge markup
18. Ecma International Draft specification for es.next (ecma-262 edition 6). http://wiki.ecmascript.org/doku.php?id=harmony:specification_drafts. Accessed 17 March 2015
19. Engeström Y (1992) Interactive expertise: studies in distributed working intelligence. In: University of Helsinki research report, no. Research bulletin 83 in HELDA—The Digital Repository of University of Helsinki. http://hdl.handle.net/10224/3666
20. Engeström Y (2000) Activity theory as a framework for analyzing and redesigning work. Ergonomics 43(7):960–974. ISI:000088268400012
21. Engeström Y (2008) From teams to knots: activity-theoretical studies of collaboration and learning at work. Cambridge University Press, Cambridge, UK; New York:, chap Teamwork between adversaries: coordination, cooperation, and communication in a court trial. http://www.loc.gov.proxy.library.carleton.ca/catdir/enhancements/fy0729/2007028833-b.html,

http://www.loc.gov.proxy.library.carleton.ca/catdir/enhancements/fy0729/2007028833-d.html, http://www.loc.gov.proxy.library.carleton.ca/catdir/enhancements/fy0729/2007028833-t.html

22. Farah H, Lethbridge TC (2007) Temporal exploration of software models: a tool feature to enhance software understanding. In: Proceedings of the 14th working conference on reverse engineering. IEEE Computer Society, WCRE '07, pp 41–49. doi:10.1109/WCRE.2007.49
23. Genest A, Gutwin C (2011) Characterizing deixis over surfaces to improve remote embodiments. ECSCW 2011: Proceedings of the 12th European conference on computer supported cooperative work, 24–28 September 2011. Aarhus Denmark, Springer, London, pp 253–272
24. Globalytica Globalytica software tools: TemACH. http://www.globalytica.com/thinksuite-html/. Accessed 03 Spt 2014
25. Hackman JR (2011) Collaborative intelligence: Using teams to solve hard problems. Berrett-Koehler Publishers
26. Haller M, Leitner J, Seifried T, Wallace JR, Scott SD, Richter C, Brandl P, Gokcezade A, Hunter S (2010) The nice discussion room: integrating paper and digital media to support co-located group meetings. In: Proceedings of the 28th international conference on Human factors in computing systems. ACM, New York, NY, USA, CHI '10, pp 609–618. doi:10.1145/1753326.1753418
27. Han JY (2005) Low-cost multi-touch sensing through frustrated total internal reflection. In: Proceedings of the 18th annual ACM symposium on user interface software and technology. ACM, New York, NY, USA, UIST '05, pp 115–118. doi:10.1145/1095034.1095054
28. Heer J, Bostock M, Ogievetsky V (2010) A tour through the visualization zoo. Commun ACM 53(6):59–67
29. Heuer RJ (1999) Psychology of intelligence analysis. Center for the Study of Intelligence
30. Heuer Jr RJ, Pherson R (2010) Structured analytic techniques for intelligence analysis. CQ Press. http://books.google.ca/books?id=ruGUQQAACAAJ
31. Heuer Jr RJ, Pherson RH (2010) Structured analytic techniques for intelligence analysis. CQ Press
32. Hypothesis Hypothes.is. https://hypothes.is/. Accessed 21 Aug 2015
33. Inselberg A (1997) Multidimensional detective. Information Visualization, 1997. Proceedings, IEEE Symposium on, IEEE, pp 100–107
34. Inselberg A, Dimsdale B (1990) Parallel coordinates: a tool for visualizing multi-dimensional geometry. In: Proceedings of the 1st conference on visualization '90. IEEE Computer Society Press, Los Alamitos, CA, USA, VIS '90, pp 361–378. http://dl.acm.org/citation.cfm?id=949531.949588
35. Isenberg P, Fisher D, Morris M, Inkpen K, Czerwinski M (2010) An exploratory study of co-located collaborative visual analytics around a tabletop display. In: 2010 IEEE symposium on visual analytics science and technology (VAST), pp 179–186, doi:10.1109/VAST.2010.5652880
36. Isenberg P, Fisher D, Paul SA, Morris MR, Inkpen K, Czerwinski M (2012) Co-located collaborative visual analytics around a tabletop display. IEEE Trans Vis Comput Graph 18:689–702. doi:10.1109/TVCG.2011.287
37. Jacob RJ, Girouard A, Hirshfield LM, Horn MS, Shaer O, Solovey ET, Zigelbaum J (2008) Reality-based interaction: a framework for post-wimp interfaces. In: Proceedings of the twenty-sixth annual SIGCHI conference on human factors in computing systems. ACM, New York, NY, CHI '08, pp 201–210. doi:10.1145/1357054.1357089
38. Kammer D, Wojdziak J, Keck M, Groh R, Taranko S (2010) Towards a formalization of multi-touch gestures. In: ACM international conference on interactive tabletops and surfaces. ACM, New York, NY, USA, ITS '10, pp 49–58. doi:10.1145/1936652.1936662
39. Kirsh D, Maglio P (1994) On distinguishing epistemic from pragmatic action. Cognit Sci 18(4):513–549. doi:10.1207/s15516709cog1804_1
40. Krasner GE, Pope ST et al (1988) A description of the model-view-controller user interface paradigm in the smalltalk-80 system. J Object Oriented Program 1(3):26–49

41. Lee B, Isenberg P, Riche N, Carpendale S (2012) Beyond mouse and keyboard: expanding design considerations for information visualization interactions. IEEE Trans Vis Comput Graph 18(12):2689–2698. doi:10.1109/TVCG.2012.204
42. Marquardt N, Kiemer J, Ledo D, Boring S, Greenberg S (2011) Designing user-, hand-, and handpart-aware tabletop interactions with the touchid toolkit. In: Proceedings of the ACM international conference on interactive tabletops and surfaces. ACM, New York, NY, USA, ITS '11, pp 21–30. doi:10.1145/2076354.2076358
43. Microsoft (2016) Microsoft surface hub. https://www.microsoft.com/microsoft-surface-hub
44. Mozilla Developer Network ECMAScript 6 support in Mozilla. https://developer.mozilla.org/en-US/docs/Web/JavaScript/New_in_JavaScript/ECMAScript_6_support_in_Mozilla. Accessed 17 March 2015
45. Mozilla Developer Network Proxy. https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Proxy. Accessed 17 March 2015
46. Multiple Contributors (2011) Enron email dataset. https://www.cs.cmu.edu/~./enron/
47. Noble J, Groves L, Biddle R (1995) Object oriented program visualisation in tarraingim. Austral Comput J 27(4):138–149
48. Owens S (2016) The total economic impact of microsoft surface hub: cost savings and business benefits enabled by surface hub. Forrester Research, Inc
49. Palo Alto Research Center (2010) Analysis of competing hypotheses software version ach2.0.5. http://www2.parc.com/istl/projects/ach/ach.html
50. Pirolli P, Card S (2005) The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. Proc Int Conf Intell Anal 5:2–4
51. Plimmer B, Chang SHH, Doshi M, Laycock L, Seneviratne N (2010) iannotate: exploring multi-user ink annotation in web browsers. In: Proceedings of the eleventh australasian conference on user interface, vol 106. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, AUIC '10, pp 52–60. http://dl.acm.org/citation.cfm?id=1862280.1862289
52. Russell DM, Stefik MJ, Pirolli P, Card SK (1993) The cost structure of sensemaking. In: Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems. ACM, pp 269–276
53. Scaife M, Rogers Y (1996) External cognition: how do graphical representations work? Int J Hum Comput Stud 45(2):185–213
54. Shneiderman B (1981) Direct manipulation: a step beyond programming languages. SIGSOC Bull 13(2–3):143. doi:10.1145/1015579.810991
55. Simon HA (1956) Rational choice and the structure of the environment. Psychol Rev 63(2):129
56. Song P, Goh WB, Fu CW, Meng Q, Heng PA (2011) Wysiwyf: exploring and annotating volume data with a tangible handheld device. In: Proceedings of the 2011 annual conference on Human factors in computing systems. ACM, New York, NY, USA, CHI '11, pp 1333–1342. doi:10.1145/1978942.1979140
57. Spindler M, Martsch M, Dachselt R (2012) Going beyond the surface: studying multi-layer interaction above the tabletop. In: Proceedings of the 2012 ACM annual conference on human factors in computing systems. ACM, New York, NY, CHI '12, pp 1277–1286. doi:10.1145/2208516.2208583
58. Suchman L (1994) Plans and situated action: the problem of human-machine communication, 4th edn. Cambridge University Press
59. Thomas JJ, Cook KA (2005) Illuminating the path. IEEE Computer Society Press
60. Tuddenham P, Davies I, Robinson P (2009) Websurface: an interface for co-located collaborative information gathering. In: Proceedings of the ACM international conference on interactive tabletops and surfaces. ACM, New York, NY, USA, ITS '09, pp 181–188. doi:10.1145/1731903.1731938
61. Wallace JR, Scott SD, Stutz T, Enns T, Inkpen K (2009) Investigating teamwork and taskwork in single- and multi-display groupware systems. Personal Ubiquitous Comput 13:569–581
62. Wallace JR, Scott SD, Lai E, Jajalla D (2011) Investigating the role of a large, shared display in multi-display environments. Comput Support Cooper Work 20:529–561

63. Wharton C, Bradford J, Jeffries R, Franzke M (1992) Applying cognitive walkthroughs to more complex user interfaces: experiences, issues, and recommendations. In: ACM conference on human factors in computing systems (CHI)
64. Wigdor D, Wixon D (2011) Brave NUI world: designing natural user interfaces for touch and gesture, 1st edn. Morgan Kaufmann Publishers, San Francisco, CA
65. Wilson J, Brown JM, Biddle R (2014) Interactive parallel coordinates for collaborative intelligence analysis. School of Computer Science Technical Report TR-14-05, Carleton Uni
66. Wobbrock JO, Morris MR, Wilson AD (2009) User-defined gestures for surface computing. In: Proceedings of the 27th international conference on human factors in computing systems. ACM, New York, NY, USA, CHI '09, pp 1083–1092. doi:10.1145/1518701.1518866
67. Yuill N, Rogers Y (2012) Mechanisms for collaboration: a design and evaluation framework for multi-user interfaces. Trans Hum Comput Inter (TOCHI)

Chapter 18
Collaboration Around an Interactive Tabletop in Rehabilitation Settings

Mirjam Augstein, Thomas Neumayr, Renate Ruckser-Scherb and Sabine Dielacher

Abstract Acquired brain injury, mostly after a stroke or an accident, is a hard cut in a person's life and often followed by a long process of rehabilitation with many ups and downs. Therapy can be perceived as monotonous due to the (therapeutically necessary) repetitive nature of the tasks. Therapy nowadays does not only involve conventional settings but often additional computer-based exercises that allow the computer to take over time-consuming routine tasks. In addition to the time factor, computer-assisted therapy can lead to higher patient motivation. Especially, when computer-based rehabilitation allows for collaborative settings, a positive effect on motivation can be noted. Collaboration can be easily facilitated with tabletop computers because they can be interacted with by multiple people in parallel. Modern tabletops can process a high number of concurrent interactions and the user interfaces can be designed in a way that allows for relevant (interactive) elements to be aligned towards different directions. This chapter presents an approach towards rehabilitation using an interactive tabletop in collaborative settings, covering the therapeutic motivation behind as well as aspects related to interaction design and modalities.

M. Augstein (✉) · T. Neumayr
University of Applied Sciences Upper Austria,
Softwarepark 11, 4232 Hagenberg, Austria
e-mail: mirjam.augstein@fh-hagenberg.at

T. Neumayr
e-mail: thomas.neumayr@fh-hagenberg.at

R. Ruckser-Scherb · S. Dielacher
Neuromed Campus Kepler Universitätsklinikum, University of Applied Sciences
for Health Professions Upper Austria, Niedernharterstr. 20, 4020 Linz, Austria
e-mail: renate.ruckser@fhgooe.ac.at

S. Dielacher
e-mail: sabine.dielacher@fhgooe.ac.at

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_18

18.1 Introduction

Acquired brain injury (ABI), e.g., through stroke or accident, is mostly followed by a long process of rehabilitation in which skills that have (temporarily) been lost are re-trained [35]. Affected skills are broad and comprise cognitive abilities (like memory or attention) as well as motor or visuo-constructive skills. Computer-based training is often part of the rehabilitation program (e.g., RehaCom [21]). For instance, computers can take over time-consuming tasks that otherwise would have to be done manually by the therapist, e.g., computation of statistics or changing levels of difficulty. The first is mainly problematic because it is costly in terms of time. The second additionally involves a motivational challenge; in conventional therapy, in case a task turns out to be too difficult for a patient, therapists often have to search for an easier one in their therapeutic material (often folders containing a huge number of task descriptions). This process is in many cases perceived as demotivating by the patient (see e.g., [14]). A computer-based change of the level of difficulty can be significantly faster and more unobtrusive.

Recently, a multi-disciplinary team consisting of therapists, software developers, interaction, graphic and object designers and assistive technology experts, has developed fun.tast.tisch.,[1] an interactive tabletop system supporting the rehabilitation process. The system comprises a number of therapeutic exercises (so-called "modules") that are all based on tasks as used in conventional therapy. The tabletop setting has several additional advantages compared to conventional computer-based settings. Many conventional therapeutic tasks involve a (traditional) table that can be easily replaced by a tabletop computer. The table is perceived as a familiar object by the patients and the horizontal orientation of the surface allows for physical objects to be placed and arranged on the table which is necessary in many therapeutic tasks (especially when training motor or visuo-constructive skills). Another important advantage of the tabletop setting is that tabletop-based interactive tasks can be worked on collaboratively. A therapist can instruct and assist the patient without taking over full control of the interaction process (as it would be the case in conventional computer-settings utilizing keyboard and mouse). Further, multiple patients can work together to solve therapeutic tasks. While this kind of setting raises additional challenges (e.g., it might no longer be easily possible to distinguish between the individual patients' contributions), it yields high potential regarding (i) motivational aspects (often, patients become tired of therapy after a while and a collaborative setting usually is a welcome change) and (ii) effectiveness and efficiency factors (multiple patients participating in a therapy session can lead to a higher number of sessions per patient while keeping the number of sessions per therapist constant).

[1]The project's name "fun.tast.tisch" originates in a wordplay in German language: "Tisch" is the German word for table, "tasten" is a German verb that can be translated with feeling by touch. Thus, the project's name suggests having fun with a touch-based interface on a table. Additionally, the name "fun.tast.tisch." sounds similar to the German word "fantastisch" which means "fantastic".

This chapter describes the motivation behind fun.tast.tisch. and provides information on the therapeutic background in Sect. 18.2. Section 18.3 summarizes related work in the fields of (i) interactive systems in rehabilitation settings and (ii) collaborative settings in tabletop interaction. The design and development process is described in Sect. 18.4. Collaborative therapeutic exercises are dealt with in more detail in Sect. 18.5. Section 18.6 describes the specific challenges related to collaboration around an interactive tabletop and discusses interaction and user interface considerations. Finally, Sect. 18.7 summarizes user perception, strengths and challenges of collaborative tabletop settings in the rehabilitation process.

18.2 Therapeutic Background

This section describes the motivation behind tabletop-based rehabilitation from therapists' point of view.

18.2.1 Neuro-Rehabilitation After Acquired Brain Injury

ABI refers to brain damage resulting from an impact to the brain out of vascular and traumatic etiology. This can be stroke, traumatic brain injury, brain tumor or anoxia [35]. Over 80 % of persons with ABI are left with residual deficits in motor control, decline in cognitive and emotional functioning, social disability, inability to care for themselves and a decrease in community participation. The severity of problems can vary from mild to severe, depending on the location and nature of their injury [35]. Many activities of daily living (ADL) require movements of the upper limb, so the rehabilitation of the upper extremity is crucial. Current exercises are reaching for objects, holding, manipulating, placing and releasing them. Patients have to perform repetitive movements that focus on increasing muscle strength and endurance, range of motion and coordination. For some patients this is unexciting, which does not motivate them to give their best [4].

ADL also require cognitive skills yet ABI often entails cognitive impairment. Attention, (especially short-term) memory, sequencing, problem solving, visuo-constructive skills and other skills could be affected. Cognitive rehabilitation aims to improve impaired skills or to enable patients to use compensatory strategies. The ability of our nervous system to reorganize its structure, connections and functions in response to training is called neuronal plasticity or brain plasticity. Other areas of the brain than prior to the injury take over functions of damaged ones. The extent of neuronal plasticity depends on the task, environmental factors as well as motivation and attention [15]. Maximizing the effect of plasticity and functional reorganization is a main aim of neuro-rehabilitation [13]. Learning theories influence rehabilitation programs and force intensity and repetition of exercises as well as task-oriented exercises. Skills retrained step-by-step will transfer to improved functional performance [14].

18.2.2 Collaborative Rehabilitation Settings

In conventional therapy, group-based rehabilitation is indicated whenever social and communicative skills have to be improved. A collaborative rehabilitation setting can support face-to-face patient interaction like working or playing together around tables and interact with each other. Often group interventions help to maximize client effort, induce positive emotional changes, foster self-understanding and help patients to regain more quality of life [22]. Group-based rehabilitation can be very motivating and many patients are more willing to spend time performing their exercises [17].

Boisselle [12] describes elements which must be fulfilled in rehabilitation programs in order to let patients benefit from a collaborative setting:

- Communication during group activities encourages patients to share their feelings and difficulties. They speak about their rehabilitative experiences and create bonds with other patients. This can improve their emotional state.
- Cooperation gives patients motivation from others, who are in a similar situation. They feel the support of others and can decrease feelings of helplessness. Using collaborative activities encourage patients to learn from each other.
- Immersion allows patients to forget about their trouble and focus on an activity. If patients become competitively immersed in an activity they often work longer and try harder to be better than their competitors. Supporting onlookers can encourage and motivate the patient.

There are also economic advantages in collaborative rehabilitation settings: Group-based therapy can reduce health-care costs if multiple patients participate in one therapy session. Thus one therapist can provide a higher number of sessions per patient. Please note that some patients, in particular those with impaired cognitive skills, may be deflected by a collaborative setting. Therapists thus have to choose the type of setting very carefully.

18.2.3 Tabletop-Support in the Rehabilitation Process

In movement rehabilitation, a high frequency of therapeutic sessions is necessary. This may lead to monotonous, uninteresting situations during exercising. Yet patients have to be inspired to perform to the best of their abilities. Computer-based therapy is an important component for optimizing the rehabilitation especially of the upper limb; it helps to give a high number of repetitions, high frequency and high intensity of exercises up to the determined performance limit of the patient. It provides acoustic and visual prompts and feedback and encourages patients by using dynamic animations and effects. Multi-touch tabletop technology has the potential to enhance patient motivation and compliance because it is highly interactive and immersive and it supports natural methods[2] of user interaction [4].

Tabletop technology supports intuitive and natural interactions; it moreover tolerates rough movement skills and imprecise manipulation. People with motor coordination difficulties benefit from this because it allows individualized and unmediated control over the interface. Multi-touch tabletops allow direct manipulation of real objects on the surface. This can support a wide range of user interactions such as multi-finger touch, hand gesture and manipulation of objects, e.g., reaching, grasping, lifting, moving and placing [20]. This provides a broad repertoire for upper limb movement rehabilitation. In combination with games it is possible to increase patient engagement. The possibility to directly manipulate data or objects on the screen is more intuitive for many patients and has a very strong appeal to them [1]. Patients with cognitive impairment often show problems when handling intermediary devices such as a keyboard or a computer mouse [26].

Multi-touch tabletops support interaction among co-located patients and provide opportunities for collaboration and training of group work skills. The table allows face-to-face contact and social interaction among multiple patients. Being positioned around the tabletop reminds patients of the social setting of board games or coffee parties at a traditional table [2]. The flexibility of tabletop activities makes it easier for therapists to adapt and gradually modify activities for each patient [2] and it helps to objectively measure and track patient progress [4].

18.3 Related Work

This section presents related work in the areas of tabletop-based rehabilitation and collaborative settings in general tabletop interaction.

18.3.1 Interactive Surfaces in Rehabilitation Settings

Interactive surfaces are applied in settings for motor, cognitive and social rehabilitation to impact brain plasticity and recovery. Mumford et al. [29] developed a virtual tabletop workspace for upper-limb rehabilitation and tested it with two patients suffering from traumatic brain injury. Participants showed improvement in movement accuracy, efficiency and bi-manual dexterity. A suite of multi-touch tabletop applications that address the needs of patients and therapists were developed by Annett et al. [4]. Patients found their application engaging and exciting to use; therapists were able to tailor activities to meet individual patient needs and performance measurements could be recorded.

---

[2] In the context of interaction, "*natural* refers to the user's behavior and feeling during the experience rather than the interface being the product of some organic process" [36].

The use of tangible objects is useful in rehabilitation—especially if fine-motor skills should be trained. They can also be a powerful tool in cognitive rehabilitation, particularly when associated with ADL. Jung et al. [25] designed an ADL-mediated cognitive rehabilitation system called E-CORE (Embodied Cognitive Rehabilitation) to combine movement and cognition through the use of tangible objects and a tabletop interface. A system developed by Leitner et al. [26] applies three concepts for rehabilitation exercises for the fields of visual impairments, visual perception problems and training of fine motor skills on the basis of tangible tabletop interfaces. This system features physical objects that are manipulated on a tabletop surface. In a more recent study, Annett et al. developed a participatory design with therapists to get a more "patient-friendly" system. The evaluation of the system shows that there is a need for customization and flexibility in the software as well as for supporting a variety of activity types. The design of activities impacts the success of technology-assisted rehabilitation more than the utilization of technology itself [3].

Research indicates that games can stimulate enjoyment in patients (e.g., reported by Li et al. [27]), enhance their learning and provide safe task conditions. They are intrinsically motivating and arouse interest, so they are a perfect complement to conventional therapy. Duckworth et al. [18] examined how computer game mechanics may leverage interactive technology to increase patient engagement and social interaction. Hancock et al. [24] present a game-based approach that also relates to art therapy. Collaborative, cooperative and competitive modes of interaction among patients have the potential to stimulate a high level of interest and enjoyment and are intrinsically motivating. Group-based rehabilitation can facilitate social interaction [19]. This corresponds to findings by Annett et al. [2] who described the benefit of multi-user interaction around a multi-touch tabletop.

18.3.2 Collaborative Settings in Tabletop Interaction

This section discusses selected representative related work in the general field ob tabletop-based collaboration (intentionally not specifically related to the rehabilitation area). Scott et al. [32], for instance, offer a systematic description of what tasks interactive tabletops should support in a collaborative setting and compare them to traditional paper-based table settings. They investigated along eight different criteria like interpersonal interaction, transitions between different interactions, or the use of physical objects. Based on this comparison they analyzed which activities by then were sufficiently supported through the technology and which still lacked support. Their criterion "support the use of physical objects" is also a major focus in the fun.tast.tisch. project and we implemented the handling of physical objects like it is suggested by Scott et al. For instance, we used objects "not previously enhanced by technology" in some modules and account for additional (irrelevant) objects placed on the table.

Other work by Scott et al. [31] deals with different types of areas (so-called "territories") on a tabletop surface that are relevant for collaborative settings: personal, group, and storage areas. They conducted observational studies in traditional collaborative tabletop settings based on pen and paper and showed how these areas are traditionally established through a process of spontaneous order (e.g., areas directly in front of a person are usually regarded as personal). Through our design phases we found it highly practicable to keep in mind which areas of our screen design should be taken up by personal interactions of therapists and patients, which parts could serve as shared spaces and which could be used to store patients' impaired arms or hands but also physical objects that are currently unused.

Silva et al. (see [34]) describe the design and evaluation phases of a collaborative game set on an interactive tabletop they designed for users with autism. The main focus here lies on the support of often inhibited collaboration abilities in this target group through especially designed collaboration patterns. E.g., users have to collaborate to store an item in a virtual box. While user A presses a button to open the box, user B moves the item into the box. A similar functionality can be observed in our Shopping module (see Sect. 18.5) where all participating users have to simultaneously touch a shared interactive element whenever they are ready to start (see Fig. 18.1 (left)).

Recently, Granda et al. (see [23]) conducted experiments with students who worked on database design activities on multiple tabletops in a classroom. A main focus of this study was to attribute individual contributions to their originators in order to be better able to assess individual performance and decrease phenomena like social loafing. In the study, each participant had a unique pen (identifiable through its color by the participants and through infrared markers for the technology). In our Shopping module we use a similar approach where patients use colored physical tokens that are additionally tagged so that the hardware can identify an activity's originator.

18.4 A User-Centered Design and Development Process

We believe that software purposed to support the rehabilitation process should be particularly well tailored to the target groups' needs and thus relies strongly on the thorough user involvement during development. This section discusses the user-centered design and development process as experienced in the fun.tast.tisch. project that consists of the steps described in the following paragraphs for all modules (also see [9, 11]).

The first step is the *creation of a module description*, usually a text explaining the motivation and therapeutic background (including suitability for collaborative settings) behind the module, in some cases enhanced by rough sketches that enable better understanding by stakeholders not familiar with the rehabilitation domain (e.g., graphic designers or software developers). Module descriptions are exclusively created by domain experts.

In order to bridge the gap between the textual description and the more formal basis for an interaction design, the second step comprises the *derivation of a hierar-*

chical task analysis (HTA) as described by Sharp et al. [33] which is also done by an expert in the rehabilitation domain. The HTA breaks down tasks into sub-tasks, sub-sub-tasks etc., ending up in a hierarchical structure that can be visualized in a table or diagram. The third step comprises the *creation of an interaction design and mock-ups* and focuses on a visual description of the interaction process as later followed by therapists and patients working with the module. As the interaction design is crucial for later user acceptance (users need to quickly understand and learn interaction patterns), this step comprises a cognitive walkthrough with all relevant stakeholders. Within the fourth step, *design prototypes* are created that involve the final visual presentation of all interactive and static elements that appear in any step of a user's later interaction with the module. The design prototypes are stored as static images. The fifth step results in the first interactive version of a module, a so-called *interactive prototype* that comprises all of the module's functionality from a software development perspective.

The sixth step covers the *evaluation* of the new module. Evaluation comprises different kinds of user tests as well as larger-scale studies in a more controlled setting in a rehabilitation clinic. User tests with patients (conducted by the therapists in the fun.tast.tisch. team) take place for all modules and aim at gaining stakeholder feedback in realistic (close to real-world therapy) settings. Larger-scale studies mainly aim at the evaluation of user acceptance regarding the overall system and are conducted with selected modules. Some modules are additionally evaluated in an efficacy study, i.e., regarding therapeutic effect in comparison to conventional intervention.

The process is an iterative one, i.e., steps can be repeated based on the feedback and insight gained in following ones. The overall design and interaction concept, i.e., the part of the design and interaction that are identical for all modules and affect the whole system (such as menus, buttons, actions like going back to the main menu, pausing or stopping a module or adjusting the level of difficulty) were dealt with very early in the project, before the development of the first module.

18.5 Collaborative Therapy Tasks for a Tabletop System

Collaboration in fun.tast.tisch. and in tabletop-based rehabilitation in general can take three different manifestations. First, there is a permanent collaborative setting that involves a patient and a therapist. Here, the system aims at (i) allowing the therapist to configure the modules for the patient and control the overall process (including starting or pausing a module) while (ii) supporting the patient in the therapy process by letting him/her solve therapeutic exercises. The system should be able to handle these processes concurrently, i.e., a therapist should have an opportunity to reconfigure a module during a patient's interaction with it without taking over full control of the interaction process. Second, patient-patient collaboration can be enabled by the system explicitly, allowing for automated assignment of activities to their originators in order to log their therapeutic progress. Third, there can be additional patient-patient collaborative settings that focus on the social engagement aspect and do not require identification of activity originators. A theoretically possible fourth collaborative setting involving one patient and more than one therapists is not realistic in practice due to the therapists' tight schedules and increased therapy costs (while not necessarily increasing therapeutic effect). This constellation is thus not further discussed in this chapter.

The fun.tast.tisch. system supports the first kind of collaboration in all of its 16 modules, thus the following paragraphs describe in more detail only the second and third kind on the basis of some representative modules. Only 4 modules are not suitable for being operated with collaboratively (by multiple patients) at all (mostly for therapeutic reasons).

Explicit patient-patient collaboration is supported by the *Shopping* module. It can be used to train memory skills but predominantly should help patients to be better able to assess their own memory abilities. First, patients specify how many items they want to buy (these have to be memorized). Next, a digital shop with up to 25 products is displayed and the patient has to choose the ones memorized before by placing a (tagged) physical token (a "shopping coin") on them. The module can be worked with collaboratively by up to three[3] patients. In this case, each patient receives their own products to memorize and their own physical tokens to pick them from the shared shop. Patients recognize their coins by different colors that are also used in the digital parts of the user interface. The exercise is successfully finished when all patients have placed their individual shopping coins on the correct goods. The module can only be started collaboratively after all involved patients agree to have finished memorizing their goods. They then have to concurrently touch a start button as shown in Fig. 18.1 (left). Patients are also motivated to discuss their memory strategies (a list of memory strategies is presented to the therapist who discusses them with the patients and selects the one(s) used (mainly for statistical purposes)).

Implicit patient-patient collaboration is e.g., supported by the *Tangram* module that is used to train visuo-constructive skills and requires patients to use seven puzzle pieces, so-called "Tans" to combine them to form a shape displayed on the table display (see Fig. 18.2 (right)). If the patient has solved a shape correctly, this is automatically recognized by the table. *Tangram Stories* is a module which in principle works exactly like Tangram but is enhanced by a storytelling component to mainly appeal to children as a target group. In the module *Spatial Cognition* therapists place physical objects (simple geometrical ones or complex ones like a tower or a house) on the surface. These objects are tagged and therefore recognized by our system concerning their shape and orientation. Next, up to four different view perspectives are generated and patients have to choose the correct one, either from their own or

[3]The number of patients is limited to three because four people can take place at the table (one at each side of the table) and one of the short table sides is reserved for the therapist (all interface elements that need to be operated by the therapist are aligned towards this side). In case all patients sit in a wheel chair, only two patients can collaborate (as the wheel chair can only be positioned at the long side of the table).

Fig. 18.1 Three patients collaboratively starting a task via touch (*left*) and collaboration in a multi-patient setting using tangibles (*right*)

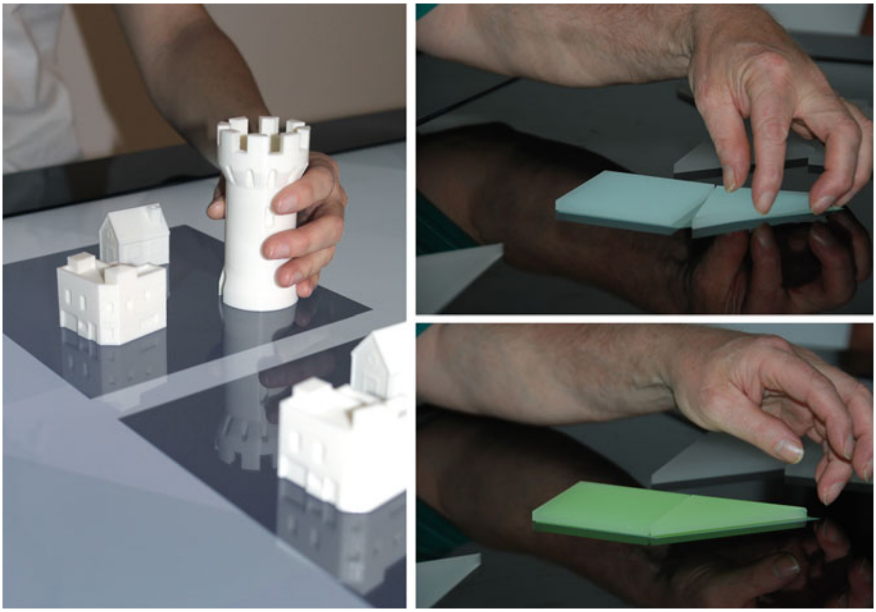


Fig. 18.2 Two different kinds of tangibles as used in fun.tast.tisch.: three-dimensional objects (*left*) and semi-transparent, flat ones (*right*) that can be used for visual feedback (*bottom right*)

another perspective. *Window Washing* mainly appeals to motor skills and mobility of affected arms and hands. Here the whole area of the surface is used to display an image that seems to be covered with a layer of digital dirt which has to be cleaned by patients. To do so, patients wipe the display with their hands, arms, sleeves or an especially constructed physical object. To solve the *Match the Pairs* exercise, patients have to select two same-colored circles at the same time by bi-manual touch out of a set of many items (see Fig. 18.3 (left)). The pairs then disappear and one round is finished when no pairs remain. *Spot the Difference* is used to train cognitive skills. The system presents two images one of which was manipulated to contain five

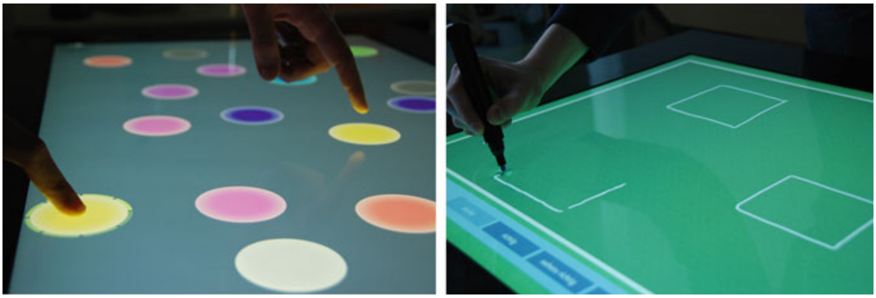


Fig. 18.3 Touch-based (*left*) and pen-based (*right*) interaction with a tabletop in fun.tast.tisch

"mistakes". The differences have to be identified by touch. For all of these modules, a higher number of patients can lead to faster problem-solving and mutual encouragement as well as an additional positive competitive aspect.

18.6 Input Modalities and Interaction Considerations

Different methods and modalities can be used when interacting with a tabletop system in general that all involve specific strengths and limitations. In the therapeutic context, these characteristics might be altered based on therapeutic implications. Most tabletop systems allow for interaction via touch and via tangible objects. An additional (less common) input method relies on interaction with a pen. The following sections discuss different ways of interacting with a tabletop system (also see [6–8]) as well as their implications on collaboration. Figures 18.2 and 18.3 show the three interaction modalities used in the fun.tast.tisch. system. Table 18.1 [7] shows patients' perception regarding the different kinds of touch input, tangibles and pen usage.

Table 18.1 Users' interaction activities in the fun.tast.tisch. system and how they were perceived (agreement between ++ and −−) [7]

Interaction activity	Perception by users		
	Intuitive	Motivating	Easy to handle
Simple touch	++	o	++
Delayed touch	+	o	+
Two-handed touch	+	+	++
Selection via tangibles	−	+	++
Arrangement of flat tangibles	++	++	+
Arrangement of 3D tangibles	++	++	+
Tracing a line with pen	++	++	o

18.6.1 Touch Interaction

Touch input has become most popular with the rise of natural user interfaces and growing popularity of smart phones and tablets. It is also the most widely used input method in tabletop systems. The tabletop hardware we use in fun.tast.tisch. (Microsoft PixelSense) is based on optical analysis of the surface image and supports a high number of touch points that can be recognized and processed concurrently which in general allows for multi-finger gestures as well as collaboration of several users. In the fun.tast.tisch. system, touch is used for the navigation through the main portal and menus of the modules. Navigation through menus via touch is straightforward in most cases—a simple tap selects a control. However, as menus take a lot of space and even more space is needed for most therapeutic exercises it is not possible to show the main menus all the time. Thus, in order to allow for adaptation of settings during run-time without fully interrupting the therapeutic exercise, we came up with a tangible-based interaction technique for adhoc-configuration (as described in the next section). Additionally, as the main navigation is hidden most of the time there must be a control that leads back to the navigation. In fun.tast.tisch. we solved this using a two-element menu that requires a user (here: therapist) to touch two controls at the same time. This is called "two-handed touch" and was introduced to avoid unintended activation of a control.

From patients' perspective, touch is also used for interaction with the system in different therapeutic exercises. Simple touch (i.e., one-handed tap) is well suitable especially in cases where the therapeutic exercise focuses on other than motor skills. For instance, if an exercise focuses on cognitive abilities (such as attention), simple touch interaction is usually sufficient. In fun.tast.tisch., we additionally introduced the so-called "delayed touch" method that requires a user to hold a touch for a predefined time before the system reacts. This kind of touch is mainly relevant when patients should be kept from trying to solve tasks on a trial-and-error basis (e.g., in the *Spot the Difference* module). It can however also be used to account for interaction difficulties of people with tremor that might often cause an unintended touch. The two-handed touch used to activate the therapist's menu was also introduced for patients, mainly for purposes of training motor skills and controlled, coordinated two-handed movement (e.g., in the *Match the Pairs* module, see Fig. 18.3 (left)).

Touch interaction has one significant drawback when it comes to collaborative settings and activities should be linkable to individual persons. In fun.tast.tisch., there are various (touch-based) exercises that can be solved collaboratively by multiple patients, however none of them allows unambiguous assignment of activities to people. Here, the collaborative aspect is mainly purposed to increase the motivational and social engagement aspects. In case an identification of an activity's originator is necessary or at least preferable, it would be an option to introduce a convention-based approach. For instance, different patients could be assigned different colors or shapes and the exercise could tag interactive digital objects with these colors or shapes to assign sub-tasks to individuals. This kind of collaboration however does not allow for spontaneous collaborative problem-solving and task separation. An alternative solution approach would be the assignment of touch interaction based on the orientation of the touch points and the seating arrangement around the tabletop. Recognition of single fingers on optical touch screens has been researched before (e.g. Dang et al. [16] discuss hand distinction for multi-touch tabletops). However, even if these approaches are capable of distinguishing hands for two-handed interaction, several conflict cases were reported also (e.g., when hands are close to each other or one hand is beneath another). In case of several people collaborating these conflict situations are likely to happen which is why we deemed approaches based on an analysis of finger orientation too unreliable for therapy situations. Another approach based on hand-contour analysis is discussed by Schmidt et al. [30]. While this approach performs well regarding user identification the authors argue that it is not suitable for the association of touches with the originating users (e.g., because the whole flat hand needs to be placed on the screen which is done in common interaction with digital objects). An approach that would technically work for fun.tast.tisch. is presented by Marquadt et al. [28] who succeed in distinction of hands, handparts and users. However, this approach requires a user to wear tagged gloves which is again not appropriate in the therapy setting.

18.6.2 Interaction with Tangible Objects

Interaction using tangible objects as input elements arose with the emergence of horizontal interactive displays (such as tabletops). In fun.tast.tisch., we use tangibles for several purposes. First, we introduced a plexiglass cylinder as control for the therapist that can be placed on the display at any point of time during the interaction and brings up a slimmed-down menu. This cylinder is used to unobtrusively reconfigure (e.g., change the level of difficulty of) the module the patient is currently working on at run-time. Second, many exercises rely on physical objects for therapeutic reasons (especially motor and visuo-constructive skills can be trained well by grasping, moving, placing or flipping real objects). In most cases it is important for the system to be able to locate and identify these objects. Third, physical objects can also be helpful regarding therapeutic aspects but without effect for the system (e.g., to help a user keep a hand or arm in a steady position). The following paragraphs discuss different kinds of tangible objects we use in fun.tast.tisch.

Tagged, three-dimensional objects as shown in Fig. 18.2 (left) are used for all settings where the system needs to be able to uniquely identify the objects and locate them precisely on the screen.[4] Three-dimensional objects can additionally be very helpful for therapeutic tasks related to spatial perception, orientation or other related skills. In addition to the three-dimensional ones, *flat objects* can be useful for tasks that traditionally work with similar objects (e.g., pieces of cardboard) like those in the *Tangram* module. The material of these objects can further be exploited to pro-

[4]Here, objects are identified by the built-in analysis and recognition of the standard 256 byte tags offered by PixelSense.

vide visual feedback to the user by highlighting the area under it so that the object appears to shine in a specific color. In fun.tast.tisch., this kind of feedback is only used positively (i.e., it is shown after a user has correctly positioned and arranged the objects, see Fig. 18.2 (right)). In order to allow for the shine-through feedback while still allowing the table to recognize the objects reliably, the material must be semi-transparent. Because of the shine-through feedback and because some specific shapes might need to be flippable (so that the user can use it from both sides), it might not be possible to tag these objects. In fun.tast.tisch., an own algorithm was introduced that recognizes shapes on the screen based on an image recognition analysis (taking into account lengths, angles and aspect ratio). The third kind of tangible as used in fun.tast.tisch., is not relevant for the system but used for therapeutic reasons only. For instance, such an object can be used in the *Window Washing* module where the user's hand is fixed to a *fully transparent flat plexiglass panel* shaped like a glove. This object is not recognized by the system but helps patients to keep their hands on the table.

In general, tangible interaction involves several advantages compared to purely touch-based input. First (and most relevant for collaborative settings), an activity with a tagged tangible object is perfectly assignable to its originator. Thus, in a collaborative therapy setting multiple patients could interact with their designated tangible objects, collaboratively and spontaneously solving tasks while the system could still keep track of the individual's progress. This is, for example, the case in the *Shopping* module (see Fig. 18.1 (right)). Second, as mentioned above, tangibles can be used for ad-hoc configuration without having to bring up the main navigation menus (and thus interrupting the current interaction flow). Third, for some tasks, performing them using tangible objects feels closer to reality for the users.

18.6.3 Interaction with Pens

Interaction with a pen-like object could in general be subsumed under interaction with tangibles. However, in case the pen acts like a real one (thus being used for writing or drawing), this kind of interaction can be treated separately. For tabletops that use optical technology like the MS PixelSense, the most reliable way of pen recognition is using pens that release infrared radiation. Pens are used in fun.tast.tisch. for a single-patient drawing task in the so-called *Complex Skills* module (see Fig. 18.3 (right)). However, pens could also play a role in collaborative therapy settings, e.g., for drawing a picture together. Further, pens could also be used in tabletop-based therapy for training hand-writing skills. Here, however, the tabletop setting might not yield much benefit, compared to conventional ones. For collaborative settings, results of pen-based activity cannot be linked to users per se. Basic distinction could be done using different pens that release their infrared light in different shapes (as the pen tip is recognized by the shape of the released infrared light).

18.7 Discussion and Conclusions

This section summarizes impressions and subjective findings related to user acceptance of a tabletop-based rehabilitation system. For the detailed results of two user studies that took place in a rehabilitation hospital, see [5, 10]. Additionally, [6] discusses some results of a usability-study with non-target-group users. Table 18.2 [7] presents factors related to tabletop-based therapy and their importance for patients and therapists. As shown in the table, one factor specifically important for patients is the possibility of multi-patient exercises. According to the feedback of many patients that took part in our user studies, collaborative tasks can be highly motivating and in many cases significantly increase fun and social engagement.

The user tests and studies have also shown that reliable, predictable and error-free behavior of the tabletop system is a most decisive factor regarding user acceptance

Table 18.2 Importance of factors related to the design and development and overall perception of a tabletop system for rehabilitation for patients and therapists (between ++ and −−)

Factor	Importance for ...	
	Patients	Therapists
Motivating feedback	++	++
Varied exercises	++	++
Configurability of exercises	o	++
Adjustable task difficulty	o	++
Therapeutically approved exercises	o	++
Therapeutic effect	+	++
Multi-patient exercises	++	o
Sophisticated design	++	o
Unconventional input modalities	++	−
Recognizability of design and interaction elements	+	+
Modern technology	++	o
Screen size	+	+
Error-free software	+	++
Reliable hardware	o	++
Easily transportable hardware	−−	+
Accessibility for wheelchair patients	++	++
Adjustable table height	o	++
Surface constitution (disinfectable)	−−	++
Availability of statistics	+	++
Availability and quality of support	−−	++
Availability and quality of tutorials	−−	+
Low expenses for hard- and software	−−	++

(for both, patients as well as therapists). Especially for patients, experience with the system was extremely frustrating if an exercise had been solved correctly but it was not recognized by the system (i.e., the user did not receive positive feedback immediately). For therapists, the motivational aspect was not most important related to software bugs, but the time factor; the therapy schedule is usually extremely tight, thus there is no time for restarting the system, trying to solve software problems etc. without leading to reduced therapy time for the patient.

As the experience with the fun.tast.tisch. project has shown, the introduction of tabletop-based therapeutic intervention in the rehabilitation process bears high potential. First, presumed that there are no major technical issues, the interaction with the tabletop is perceived as a welcome change in the rehabilitation process. Most patients describe their experience with the system as well as the device itself with positive attributes like “fun”, “motivating”, or “exciting”. For therapists, the tabletop setting involves additional advantages like the opportunity of intervening in the task solving process unobtrusively and without having to interrupt the patient’s interaction.

Limitations regarding rehabilitation with an interactive tabletop and collaborative settings in this domain are mainly related to the identification of an interaction’s originator which is only easily possible with the use of tangible objects. Additional challenges that came up in the concrete case of the fun.tast.tisch. system are related to the light-sensitivity of the hardware and its inability to reliably recognize and process a high (here, >15) number of tags concurrently are specific to MS PixelSense and thus are not critical for collaborative interactive tabletop settings in rehabilitation in general. Other challenges involve the acceptance of therapeutic institutions which in many cases relies not only on the acceptance of the system and the ideas behind but also on the budget that is available for therapeutic material.

References

1. Alcañiz M, Abarca V, Lozano JA, Herrero N (2009) Analysis of multitouch technology for neurorehabilitation. Off J Int Assoc CyberPsychol Train Rehabil 2(3):183–190
2. Annett M, Anderson F, Bischof WF (2010) Hands, tables, and groups make rehabilitation awesome! In: Proceedings of the 15th annual CyberTherapy and CyberPsychology conference, pp 3–8
3. Annett M, Anderson F, Bischof WF (2016) Activities and evaluation for technology-based upper extremity rehabilitation. In: Hu F, Zhang T (eds) Virtual reality enhanced robotic systems for disability rehabilitation, Chapter 15. Hersey, pp 307–338
4. Annett M, Anderson F, Goertzen D, Halton J, Ranson Q, Bischof WF, Boulanger P (2009) Using a multi-touch tabletop for upper extremity motor rehabilitation. In: Proceedings of the 21st annual conference of the Australian computer-human interaction special interest group, pp 261–264
5. Augstein M, Neumayr T (2014) Challenges and opportunities of fun.tast.tisch. in interactive rehabilitation settings. In: Proceedings of the 2nd international workshop collaboration meets interactive surfaces. Walls, Tables, Tablets, Phones, co-located with ITS2014. Dresden, Germany

6. Augstein M, Neumayr T (2014) Touch and tangibles—interacting with fun.tast.tisch. results of a study with non-target-group users. In: Proceedings of the 2014 Workshop on tactile/haptic user interfaces for tabletops and tablets (TacTT), held in conjuction with ITS2014. Dresden, Germany
7. Augstein M, Neumayr T, Karlhuber I, Dielacher S, Altmann J, Öhlinger S (2015) Tabletop-supported rehabilitation with fun.tast.tisch.: lessons learned. In: Proceedings of the 2015 international conference on interactive tabletops and surfaces. Funchal, Madeira, Portugal, pp 283–288
8. Augstein M, Neumayr T, Karlhuber I, Dielacher S, Öhlinger S, Altmann J (2015) Training of cognitive performance in complex tasks with a tabletop-based rehabilitation system. In: Proceedings of the 2015 international conference on interactive tabletops and surfaces. Funchal, Madeira, Portugal, pp 15–24
9. Augstein M, Neumayr T, Ruckser-Scherb R, Karlhuber I, Altmann J (2013) The fun.tast.tisch. project—a novel approach to neuro-rehabilitation using an interactive multiuser multitouch tabletop. In: Proceedings of the 2013 ACM international conference on interactive tabletops and surfaces. St. Andrews, United Kingdom, pp 81–90
10. Augstein M., Neumayr T, Schacherl-Hofer I (2014) The usability of a tabletop application for neuro-rehabilitation from therapists' point of view. In: Proceedings of the 9th ACM international conference on interactive tabletops and surfaces. Dresden, Germany
11. Augstein M, Neumayr T, Schacherl-Hofer I, Öhlinger S (2013) Iterative user-centered design for interactive tabletop interfaces in neuro-rehabilitation. Mensch & Computer 2013: Interaktive Vielfalt., Interdisziplinäre FachtagungBremen, Germany, pp 309–312
12. Boisselle A (2016) Considerations for the end-user: standards, collaboration, and participation. In: Hu F, Lu J, Zhang T (eds) Virtual reality enhanced robotic systems for disability rehabilitation, Chapter 1. Hersey, pp 1–12
13. Cameirão M, Badia BiS, Oller ED, Verschure PF (2010) Neurorehabilitation using the virtual reality based rehabilitation gaming system: methodology, design, psychometrics, usability and validation. J Neuroeng Rehabil 7(48):1–14
14. Cicerone KD, Langenbahn DM, Braden C, Malec JF, Kalmar K, Fraas M, Laatsch L, Harley J, Bergquist T, Azulay J, Cantor J, Ashman T (2011) Evidence-based cognitive rehabilitation: updated review of the literature from 2003 through 2008. Arch Phys Med Rehabil 92(4):519–530
15. Cramer SC, Mriganka S, Dobkin BH, O'Brien C, Sanger TD, Trojanowski JQ, Rumsey JM, Hicks R, Cameron J, Chen D, Chen WG, Cohen LG, deCharms C, Duffy CJ, Eden GF, Fetz EE, Filart R, Freund M, Grant SJ, Haber S, Kalivas PW, Kolb B, Kramer AF, Lynch M, Mayberg HS, McQuillen PS, Nitkin R, Pascual-Leone A, Reuter-Lorenz P, Schiff N, Sharma A, Shekim L, Stryker M, Sullivan EV, Vinogradov S (2011) Harnessing neuroplasticity for clinical applications. Brain 134(6):1591–1609
16. Dang CT, Straub M, André E (2009) Hand distinction for multi-touch tabletop interaction. In: Proceedings of the 4th ACM international conference on interactive tabletops and surfaces, pp 101–108
17. De Weerdt W, Nuyens G, Feys H, Peter V, Van de Winckel A, Nieuboer A, Osaer J, Kiekens C (2001) Group physiotherapy improves time use by patients with stroke in rehabilitation. Aust J Physiother 47(1):53–61
18. Duckworth J, Bayliss JD, Thomas PR, Shum D, Mumford N, Wilson PH (2014) Universal access in human-computer interaction. In: Universal access to information and knowledge: 8th international conference, UAHCI 2014. Lecture notes in computer science 8514. Tabletop computer game mechanics for group rehabilitation of individuals with brain injury. Springer International, pp 501–512
19. Duckworth, J., Mumford, N., Caeyenberghs, K., Eldridge, R., Mayson, S., Thomas, P.R., Shum, D., Williams, G., Wilson, P.H.: Universal Access in Human-Computer Interaction. Access to Learning, Health, Well-Being, Lecture Notes in Computer Science 9177, chap. Resonance: An Interactive Tabletop Artwork for Co-located Group Rehabilitation and Play, pp. 420–431. Springer International (2015)

20. Duckworth J, Thomas PR, Shum D, Wilson PH (2013) Design, user experience, and usability. In: Health, learning, playing, cultural, and cross-cultural user experience: second international conference, DUXU 2013, Lecture notes in computer science 8013. Designing co-located tabletop interaction for rehabilitation of brain injury. Springer, Berlin, Heidelberg, pp 391–400
21. Fernández E, Bringas ML, Salazar S, Rodríguez D, García ME, Torres M (2012) Clinical impact of rehacom software for cognitive rehabilitation of patients with acquired brain injury. MEDICC Rev: Int J Cuban Health Med 14(4)
22. Gauthier L, Dalziel S, Gauthier S (1987) The benefits of group occupational therapy for patients with parkinson's disease. Am J Occup Therapy 41(6):360–365
23. Granda RX, Chiluiza K (2015) Supporting the assessment of collaborative design activities in multi-tabletop classrooms. In: Proceedings of the 2015 Asia-Pacific conference on computer aided system engineering
24. Hancock M, ten Cate T, Carpendale S, Isenberg T (2010) Supporting sandtray therapy on an interactive tabletop. In: Proceedings of the SIGCHI conference on human factors in computing systems (CHI10), pp 2133–2142
25. Jung J, Kim L, Park S, Kwon GH (2013) Converging clinical and engineering research on neurorehabilitation. In: E-CORE (Embodied COgnitive Rehabilitation): a cognitive rehabilitation system using tangible tabletop interface. Springer, Berlin, Heidelberg, pp 893–897
26. Leitner M, Tomitsch M, Költringer T, Kappel K, Grechenig T (2007) Designing tangible tabletop interfaces for patients in rehabilitation. In: Proceedings of the conference and workshop on assistive technologies for people with vision and hearing impairments: assistive technology for all ages. Granada, Spain
27. Li Y, Fontijn W, Markopoulos P (2008) A tangible tabletop game supporting therapy of children with cerebral palsy. In: Proceedings of the 2nd international conference on fun and games, pp 182–193
28. Marquadt N, Kiemer J, Ledo D, Boring S, Greenberg S (2011) Designing user-, hand-, and handpart-aware tabletop interactions with the touchid toolkit. In: Proceedings of the 6th ACM international conference on interactive tabletops and surfaces, pp 21–30
29. Mumford N, Duckworth J, Eldridge R, Guglielmetti M, Thomas P, Shum D, Rudolph H, Williams G, Wilson PH (2008) A virtual tabletop workspace for upper-limb rehabilitation in Traumatic Brain Injury (TBI): a multiple case study evaluation. Virtual Rehabilitation 175–180
30. Schmidt D, Chong MK, Gellersen H (2010) Handsdown: hand-contour-based user identification for interactive surfaces. In: Proceedings of the 6th nordic conference on human-computer interaction: extending boundaries, pp 432–441
31. Scott SD, Carpendale S, Inkpen KM (2004) Territoriality in collaborative tabletop workplaces. In: Proceedings of the 2004 ACM conference on computer supported cooperative work, pp 294–303
32. Scott SD, Grant KD, Mandryk RL (2003) System guidelines for co-located, collaborative work on a tabletop display. In: Proceedings of the 8th European conference on computer supported cooperative work, pp 159–178
33. Sharp H, Rogers Y, Preece J (2007) Interaction design: beyond human-computer interaction, 2nd edn. Wiley
34. Silva GFM, Raposo A, Suplino M (2013) Par: a collaborative game for multitouch tabletop to support social interaction of users with autism. In: Proceedings of the 5th international conference on software development and technologies for enhancing accessibility and fighting info-exclusion
35. Turner-stokes L, Pick A, Nair A, Disler PB, Wade DT (2015) Multi-disciplinary rehabilitation for acquired brain injury in adults of working age. Cochrane Database Syst Rev 2015(12)
36. Wigdor D, Wixon D (2011) Brave NUI world. Designing natural user interfaces for touch and gesture. Morgan Kaufmann

Chapter 19
Visual to Non-visual Collaboration on a Dynamic Tactile Graphics Display

Jens Bornschein and Denise Prescher

Abstract In this article, a collaborative workstation for sighted and visually impaired users is proposed. It can be used for creating tactile graphics in a collaborative manner. The workstation consists of a classical drawing application extended with tools supporting the design of tactile graphics as well as a non-visual interface to the drawing application. As a result, blind users get both auditory and tactile feedback from the workstation through a dynamic planar tactile pin-matrix device. We also introduce supporting features as well as discuss problems in collaboration. Ultimately, we provide a set of recommendations for building a collaborative system with these different interface modalities.

19.1 Introduction

Access to graphical content for visually impaired readers, e.g. blind people, is normally obtained by a verbalization of an image. This description is only one possible interpretation of the content of an image. Besides, it is an indirect way to present this information to the user. A more direct way to gain non-visual access to pictorial information is to present the image as a tactile graphic.

A tactile graphic is a representation of an image that can be explored through touch. To create a tactile graphic, raised structures are added to a solid base, such as a sheet of paper or plastic (see [1]). The raised structures can be added manually by hand, such as building collages of different materials or scratching tactile structures into a baseplate. Tactile graphics can also be produced by embossers. Embossers press pins from behind into the paper, producing tactually perceptible dots on the front. Other production methods use a special paper that will expand when heat is

J. Bornschein (✉) · D. Prescher
Human-Computer Interaction, Technische Universität Dresden, Nöthnitzer Straße 46, 01187 Dresden, Germany
e-mail: jens.bornschein@tu-dresden.de

D. Prescher
e-mail: denise.prescher@tu-dresden.de

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_19

applied, called microcapsule paper. This microcapsule paper can be printed with a normal printer. If a heat lamp is placed over it, dark printed areas will heat up more than light ones, revealing the tactile structures.

To reduce the complexity of the tactile structures, graphics can be enriched with additional verbal information. These combinations of tactile and auditory feedback are called audio-tactile graphics. It can be realized by using a tactile graphic hardcopy as overlay for touch-sensitive input devices. By touching a tactile element, additional information from a corresponding source file is given to the user through auditory output (e.g. text-to-speech). As a result, the use of Braille text in tactile graphics can be significantly reduced while increasing the information content of the graphic.

To produce a tactile graphic image, a sighted transcriber has to create an equivalent to the graphic while considering the special needs of a blind reader. This typically means the original graphic has to be reduced in complexity and reorganized. In addition, the image must be rendered in grayscale, as only grayscale graphics are useful for tactile production methods. This is not an easy task. Most transcribers have problems in preparing appropriate and well usable tactile adaptations of images. This is often because the sighted transcriber does not exactly understand the differing information capturing processes and capabilities of their visually impaired clients. As a result, sighted transcribers often produce inadequate or unusable tactile graphics.

In particular, it is difficult to render a three-dimensional object into a tactual representation for blind readers. Congenitally blind people have difficulties in identifying a basic 3-D object out of a deformed two-dimensional perspective projection. For example, a wheel of a bicycle can be rendered into an elliptic shape if it is presented as a front-side view. However, a reader who does not have conceptual knowledge of visual perspective will not immediately identify this ellipse as a wheel, since this reader would expect a wheel to be rendered as a circle. This same phenomenon happens with the presentation of overlapping or hidden objects [1].

Another major problem in tactile graphics is the complexity of the images. Complexity not only refers to the amount of information but also to its readability and understandability. The more complex the image, the more difficult it is to find, distinguish and identify the objects in it. Therefore, keeping graphics as simple as possible (simplicity) is one of the main recommendations of common guidelines for tactile graphics [2]. The question then becomes how simple an image needs to be in order to be rendered as a tactile graphic. This decision requires a lot of practice and experience, but is the key for achieving high quality tactual representations. Even skilled transcribers sometimes have problems as they are not objective testers for their work and are not part of the user group for their products. Therefore, there is a lack of quality management when producing these tactile images. Often a graphic is revised after getting feedback of users after supplying them an insufficient version but not in advanced [3].

Looking at the product development technique of user-centered design, the propagated advantage is the final product is more suitable to the users' expectations, needs, and preferences. To reach this objective, a multi-cyclic development process is necessary. User evaluations of prototypes have to be applied frequently during development. This helps to identify problems at an early stage and to receive continuous user feedback about the evolving product.

For the process of creating tactile graphics, such a user-centered approach is not commonly practiced. This can be attributed to the fact that early production levels of tactile graphics are not easy to produce and distribute. Normally, a tactile hardcopy has to be produced, which can be expensive and time-consuming.

The main objective of the Tangram project is to make the process of producing tactile graphics more efficient and to increase the quality of the resulting tactile materials. Within this project, the Tangram workstation is developed. This workstation not only includes an extended graphics editor for sighted drawers of tactile graphics, but also a direct tactile interface through a refreshable, two-dimensional tactile pin-matrix display with touch sensor capabilities. Using this workstation, a collaborative team consisting of a sighted graphic producer and a visually impaired lector can build tactile graphics in a more user-involved manner.

The main idea of the workstation presented here is enabling a sighted graphic transcriber to transform graphical content into a tactual representation to his best knowledge. Afterwards a reviewer—optimally a visually impaired expert with significant Braille skills—will be consulted to give well-grounded feedback about the transcription. If some concerns exist, both the graphic transcriber and the reviewer can try to overcome the shortcomings in a collaborative manner.

In the following sections, the Tangram workstation, as well as the available collaborative functionalities, are described. A collaborative evaluation scenario for the workstation is also described. Furthermore, based on an evaluation of the workstation in a collaborative scenario, user and developer experiences, and literature, some recommendations for supporting collaborative environments between a visual and non-visual user are given.

19.2 Related Work

Bringing a blind and a sighted person together for collaboration is not new. Even in the case of quality management, this approach is used extensively (e.g. in case of proofreading Braille-translated texts from books or newspapers). To enable a blind user to participate actively in a beneficial and meaningful way, a sufficient presentation of information has to be provided [4]. To allow for successful collaboration, the user's awareness of the activities of all other participants is important [5]. This applies to blind users in particular.

In most cases, collaboration between sighted and blind people is utilized in teaching environments, not for productive working scenarios. In the screen reader *Jaws for Windows*, a tandem function is available. This function allows a teacher to guide a student by remotely accessing the student's instance of the screen reader [6]. In this scenario, it does not matter if the teacher is a sighted or a blind user. As this demonstrates, classical screen readers for making graphical user interfaces accessible to blind users should be designed to support collaboration. Such investigations were even done for the usage on tactile pin-matrix devises on the example of collaborative e-learning environments [7]. A blind user should not work isolated with a computer and his understanding of the user interface should be similar to that of his sighted colleagues [8].

In the *McSig* system [9], a sighted teacher trains a blind student calligraphy to enhance his skills in writing signatures. The workstation used in the study contains of a touch screen for the trainer and a force feedback device, called Phantom, providing the student with haptic output. The Phantom force feedback device is often used for non-visual collaborative approaches. For instance, several systems for learning math, geometry or science utilize this system for exploring graphics and charts [10–14].

The Phantom force feedback device can also be used for drawing tasks. In the *AHEAD* system, the Phantom is used to explore raised line graphics as well as to draw them [15]. With this system, a second user can use the mouse to draw or guide the blind user using the Phantom device within the drawing. Audio-tactile elements can be created and explored as well. However, not all blind users like to be guided by their sighted partner.

With the *BPLOT3* system, a blind user can draw images by entering textual commands in a plotter-control-language as well as by touch input [16]. A synchronized graphical user interface to the console interface allows a sighted user to check and edit the work of a blind user. In *BLPOT3* no tactile feedback of the tactile graphic is provided until it was distributed as a hardcopy. In this system, the sighted user plays the role of the lector for the blind user since only the visual user can check for errors in the graphic.

19.3 The Tangram Workstation—Collaborative Creation of Tactile Graphics

For the Tangram workstation, the popular and freely available open-source project *LibreOffice*[1] is chosen to produce and edit graphics. The *LibreOffice* suite includes classical office applications for creating text documents, spreadsheets, presentations, and a drawing application called *Draw* to create vector graphics. In contrast to pixel-based images, images in vector format are easier to export in different sizes as their output size can be changed without quality loss. They can also be easily modified due to their object-based nature and, therefore, they are well suited for the creation of tactile graphics. Furthermore, digital based tactile graphics have to be

[1]https://libreoffice.org.

simple as they are usually grayscale or binary. Additionally, the object-based approach of vector graphics makes it easy to enrich them with additional textual information and allows for a distribution as audio-tactile graphics.

19.3.1 Interfaces to the System

The sighted user interacts with the graphical user interface (GUI) of *LibreOffice Draw* in ways common to most software systems that implement GUIs. As input modalities, he uses the mouse pointer and the computer's keyboard. Interactions are controlled by the office system, which also handles plausibility checks, error handling, and logging. A special toolbar extends the classical GUI for the sighted drawer (see [17]), which contains elements for easily accessing the general properties of the graphic's objects, such as position and size. There are also special tactile image creation support functionalities included in the toolbar. These functionalities include predefined tactile filling patterns, line styles, and other macros to support the creation of tactile images. A special dockable-window dialog also provides access to some special properties for audio-tactile distribution, such as title and description input fields for the currently selected graphical objects. This makes it easy to enrich the image with additional information about specific elements.

Beyond the basic interface with its extensions, a non-visual interface was added. The non-visual interface provides the blind user with audio-tactile output as well as different ways of input. For the tactile output a two-dimensional dynamic pin-matrix system called the *BrailleDis 7200* [18] is used. The display consists of 120 columns and 60 rows of piezo-electric actuated pins. Binary tactile images with a resolution of about 10 dots per inch (dpi) can be displayed. The display area itself is touch-sensitive, with a touch resolution of about 5 dpi. 36 hardware buttons are arranged around the presentation area (see Fig. 19.1). On the left and the right side of the device, two cursor-keypads with five buttons are installed. Underneath the

Fig. 19.1 Tangram workstation with the non-visual interface for the blind user (tactile pin-matrix device, *left side*) and the visual interface for the sighted user (*right side*)

display, a long navigation bar is available. On the top, twelve buttons are placed which can be used as a Braille keyboard for entering text. This device gives the blind user access to more than 60 different functionalities for exploration, editing, and supporting collaboration. Text editing functionalities are also available through using the Braille-keyboard-like function keys behind the display area.

To support the sighted user in getting a better understanding of what is tactually presented, a visualization of the tactile output is displayed on the screen. With this tool, which was used previously as a debugging tool for development, the tactile output with a readable substitution for Braille-text and a visualization of the touch-sensory data (current hand movements of the blind user) are presented.

19.3.2 Non-visual Data Access and Information Presentation

As the sighted user is using the GUI, and since the system is built for collaboration, utilizing the GUI also for the blind user, for example by sharing the same focus or input controls, would hinder the sighted partner in using the application efficiently. Therefore, a separated and independent input/output (IO) interface has to be built. Accessing the pure pictorial information of the sighted graphic transcriber's drawing is relatively easy. To accomplish this task, the screen is mirrored with a frequency of about 10 Hz and transformed into a binary image representation that is shown on the tactile display (see Fig. 19.2). This screenshot is downscaled to the requested size by applying a changeable zoom-factor where one pixel corresponds to one tactile pin. On this downscaled image, a filter is applied converting the lightness of a colored pixel into a binary value for the pin. Light pixels will result in lowered pins and dark pixels will result in raised ones.

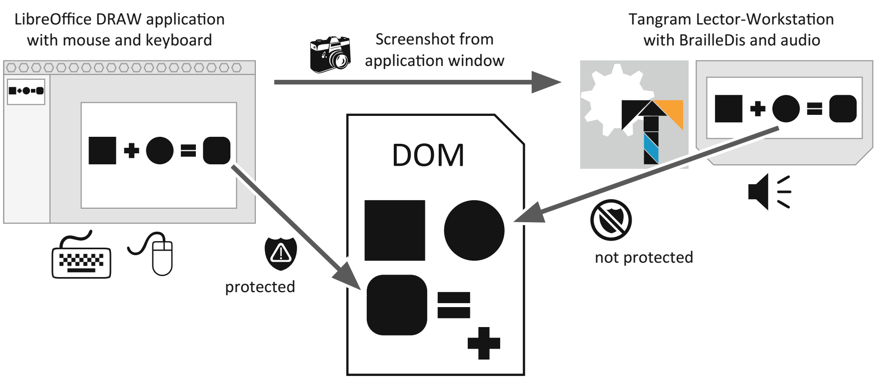


Fig. 19.2 Synchronization of the visual (*left*) and the non-visual interface (*right*) of the Tangram workstation

The blind reader can freely explore this representation through zooming and panning operations. He can pan in a two-step manner by using the navigation bar control element on the front side of the device to navigate in four directions. The first step is performing a small panning of only a few pins translation; the second performs a page jump (the visible content is panned without any overlapping to the previous view). A similar two-step approach is implemented for the zoom functions. Two rocker switches, on the left and right side of the device, perform an increasing when pressing it upwards and a decreasing when pressing it downwards. The left rocker performs small zoom operations and the right one a 3× zooming.

A different level of detail of the graphical information is available at different zoom increments. With a small zoom factor, the reader gets an overview of the whole document and its layout. With a high zoom factor, detailed structures, as well as small objects, become visible. A special zoom factor, called 'print zoom', is also implemented. With 'print zoom', images appear in the same size as they would be on a DIN A4 print-out. This allows the blind reader to discern the final resulting dimensions and appearance of the image. Furthermore, while for the sighted user text elements are presented in normal ink-print letters, text is replaced and presented in tactile Braille for the blind reader. This is only possible for the print zoom, as Braille always needs to be in a certain size. Otherwise, the text-replacement would cover underlying graphical elements and would destroy the layout for the blind user.

Getting access to the object structure is more challenging. This information is obtained from two different sources of *LibreOffice*. The first is through the accessibility interface. Classical screen readers, such as *JAWS,* use this interface for gathering information and controlling the GUI. The second is accessing the native document object model (DOM). This allows for information collection and manipulation of the document. A synchronization of both options is used by the Tangram system to guarantee valid and powerful non-visual access.

The tactile user interface (TUI) is separated into several specialized regions (compare Fig. 19.3b). This is necessary to provide the user a reliable and effective interface. The regions are immobile and display different kinds of information. On

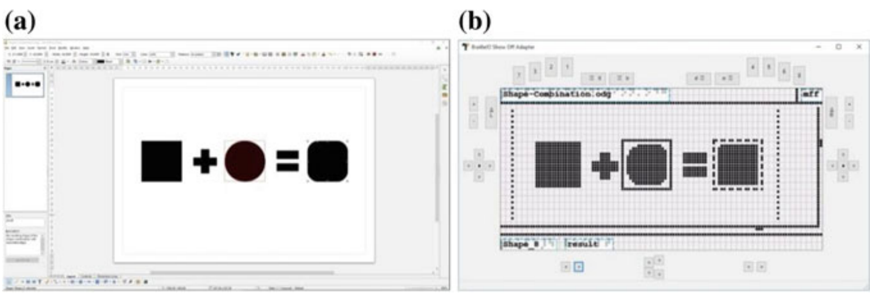


Fig. 19.3 Focus awareness within the Tangram system: graphical user interface of the sighted user (**a**) and output on the pin-matrix device for the blind user (**b**)

top there is a header region, which displays document information such as the document's title. On the right side of this region, there is a small region displaying the current interaction mode of the TUI using three Braille letters. The main space of the TUI is utilized for presenting the tactile image. At the bottom of the display area, there is a detail region where system status information, as well as detail information about objects, can be displayed in Braille. When the user requests an object, information about the requested object is also given by auditory text-to-speech output.

The touch-sensitivity of the device enables the user to directly interact with the displayed images on the TUI through using gestures. Pointing at an object allows the user not only to get more information (e.g. ID, title, and description), but also allows the user to manipulate the object. By touching an element on the pin device, the system tries to match the touch position with the corresponding position on the screen and requests the accessibility interface of *LibreOffice* for the object at the requested pixel position, which is then returned to the user.

Gestural input on tactile interfaces has a disadvantage: since a user's information absorption is realized by the sense of touch, and the hands are used to explore the content, at the same time, the touch is used as an input modality. This contradiction and double assignment leads to the so-called 'Midas touch effect'. Originally founded for gaze interaction, the Midas Touch Effect describes the situation of a system not to be able to distinguish between the intention to explore and the intention to interact [19]. To prevent the system from always interpreting touch sensory data as input commands, a mode switch for gestural interaction has to be applied. If the user wants his touch to be interpreted as a gesture, the system has to become aware of this intention. In the Tangram system, the user must press a button to perform a gesture in order to avoid this effect. However, this solution leads to a setback: only one-handed gestures are possible.

19.3.3 Non-visual Graphic Manipulation

As mentioned before, the blind reviewer can access graphical, textual and additional enriched information of the graphic for giving a precise feedback about the quality of the graphic presented on the device. Beyond pure exploration, the blind reviewer can also manipulate objects actively and independently. The blind user can also check some adaptations in layout on his own to find the most promising one. He can also rearrange the image composition for a better understanding of the image.

To do so, several manipulation options exist, although they are limited in scope. These include changing the following properties of elements: (1) position on the drawing sheet, (2) object size in two dimensions, (3) rotation, (4) filling style, and (5) style of the outline. When manipulating the filling style and the outline, a palette of nine predefined filling patterns and three line styles are available to choose from. All of them are tested to be robust and well usable in the context of tactile graphics and are also available for the sighted user through the toolbar. Edit textual values and enrichments, such as title and description tags of objects, is possible through the non-visual interface, too.

To manipulate an object, the user has to send a request either through touch or by tabbing through the document scene graph—which corresponds to the visible elements of the DOM. After selecting an object, a manipulation mode has to be chosen by scrolling through a 'rotating menu' in which all the five mentioned editing modes are selectable in a carousel menu metaphor (see [17]). To switch through the modes, the central button of the right cursor-keypad has to be pressed. After selecting an editing mode, the direction buttons of the cursor-keypad can be used to manipulate the object intuitively. For example, the object is moved to the right by pressing the right button, the vertical size is increased by pressing the up button, and the vertical size is decreased while pressing the down button.

To make manipulations recognizable on the tactile device, as well as fine-grained enough to enable even small changes, the step size for changing the position or size of an object is related to the current chosen zoom level of the TUI. Herby, the step size of the changes is set to a value that adapts the element in a way that, at least, a change of one pin is recognizable by the user.

The system does not handle conflicts if both users are editing the same. It will not lock any objects to prevent the partner from editing the element. The last stored or done manipulation overrides the previous one. In case of graphical object manipulation, this is not a substantial problem, as the non-visual manipulation is step based and the visual and non-visual representation is updated immediately for both users. As a result, the sighted user sees the changes of the blind partner as well. It is more problematic if the sighted user holds the element for a direct manipulation via a mouse button press. If he releases the object from his manipulation state, the edits of the blind user will be overwritten. At the moment, such conflicts have to be resolved by the communication between the two partners and thoughtfulness.

The blind user can also enter text by switching his interface into a Braille mode. In this mode, he can use the Braille-keyboard buttons behind the display area to enter key combinations that represent single Braille-characters. These are transformed into standard text and added to the object's properties. This is a common way for blind users to create text, for example on a Braille-typewriter.

19.3.4 Support of Collaboration

In contrast to cooperation, where all participants are working separately, a collaborative environment necessitates direct interaction, including discussion and information exchange. Therefore, a collaborative system should support conversation among the participants. To achieve this goal for a visual and a visually impaired partner, both partners have to have a relatively similar amount of information. This means the visual and the non-visual interface have to be synchronized with as little delay as possible.

Synchronization between two interfaces with the same capabilities is relatively straightforward, but synchronizing two user interfaces with such a heterogeneous possibility of conveying information density is quite difficult and can only be realized in a multimodal manner. The amount of information available for a sighted user of a GUI is higher than that of a blind user retrieving his information in a more serialized way. The lack between the two highly differing speeds of information absorption cannot be completely resolved—especially in a graphical content—by the interface design only. Therefore, supporting communication between the two users becomes even more important.

Two things are necessary to facilitate this communication. Firstly, both partners need to be aware of each other's manipulations, so that nobody overrides the changes of the other. Secondly, both partners need to be able to determine the elements of the image that need to be discussed. Both tasks can be handled by a suitable focus awareness method.

Focus awareness means that both partners are aware of the element which is selected. The TUI is fully independent of the GUI, and therefore, has its own focus, which marks an element for editing—called Braille focus. The sighted user commands the GUI focus. With this method, two independent foci exist and have to be synchronized.

To the blind user, the editing focus (Braille focus) appears as a blinking solid frame of pins covering the bounding box of the selected object (see the circle shape in the middle of Fig. 19.3b). The frame is designed as a solid frame since solid lines are easiest to follow and to identify in tactile graphics. Blinking means that the frame is changing between raised and lowered pins with a frequency of about 10 Hz. This makes the focused element easier to find. This is also supported by the gentle mechanical sound of the piezo-electric actuators, indicating that there is something blinking in the display area.

The focus of the GUI itself, which is the element given focus by the sighted partner, is extracted through the application's accessibility interface. The GUI focus is presented as a dashed blinking frame on the tactile display, covering the bounding box of the selected element (see the rounded square shape on the right side of Fig. 19.3b). If the blind user wants to know which element his sighted partner has selected, he can request the current GUI focus to be displayed. To avoid searching the whole drawing, the view port of the TUI is automatically set to the position of the focused element.

Focus awareness is not only important for the blind user, but it is also crucial for the sighted partner. As a result, the sighted partner is notified about manipulations on elements through the non-visual interface. The Braille focused object is overlaid for the sighted partner on the screen by a red to gray blinking clone of the object, surrounded by a frame whenever the Braille focus changes (see the circle shape in the middle of Fig. 19.3a). In contrast to the tactile marker of the GUI focus, this overlay only appears for about two seconds. There is no need for a longer marking as the overlay is distracting. A continuous marking would annoy the sighted user too much. For making an index operation to his sighted partner, the blind user can force the system to display this overlay again at any time. This way, the sighted and the blind user can make pointing and indexing operations, not only for preventing an overwriting of changes, but also for guiding discussions.

19.3.5 Assessment of the System and Its Collaborative Approach

With the Tangram workstation, a collaborative creation of and quality management for tactile graphics become possible in a digital and direct way through a user-centered approach. To get feedback about the system, we have conducted a user study with eight pairs of sighted and blind participants. The sighted users assumed the role of graphic transcribers. Four of them are professionals in transcribing tactile graphics from universities and a library for blind readers. The others are laymen as they have no or only minor experience with tactile graphics. The eight legally blind participants all have experience with tactile graphics and some experience with the pin-matrix device. Before the study, the sighted partner had to transform a schematic image into a tactile version of that same image (pre version). Every sighted participant then discussed his transcription with a blind reviewer. Afterwards, both participants should adapt or improve the graphic collaboratively until both partners were satisfied with the result (post version). The pre and post versions of the graphic were produced as tactile hardcopies. A detailed description of the study and its results is reported in [20]. We now explain the most important findings of the study for giving recommendations. Furthermore, we present the assessment of the produced graphics by independent blind persons.

19.3.5.1 General User Feedback

Blind participants loved the possibility of getting easy and direct access to graphics, as well as controlling graphical elements on their own. They also liked the low latency between making changes and feeling them. However, the blind users also wished for more independent image creation possibilities without the need of a sighted user as partner.

Sighted participants were less optimistic when evaluating the system. Most of the professional graphic transcribers complained about the limited or differing functionalities of the graphics editor in contrast to their normally used drawing applications. However, non-professionals liked the system and explored the system more. Overall, all users in the evaluation could see benefits in involving a blind user in the creation process, but also worried about the time it took to prepare a graphic.

19.3.5.2 Limitations of the System

Currently it is neither possible to create nor to delete objects though the non-visual interface. We classified creation and deletion—especially deletion—as critical functions with a special need for protection against an accidentally execution. Our protection mechanism does not to allow the blind user to delete elements. To delete elements, the blind partner needs to confer with the sighted partner. Another reason why such functionalities have to be installed with care is that the non-visual interface manipulates the DOM directly. This means no application error correction, such as undo-redo-logging or validity checking for object properties, is present when editing. In fact, the blind user is more powerful than the GUI user since the original DOM is manipulated, and therefore can theoretically bring the document or system into an invalid state, destroying the whole collaborative work. The GUI user, in contrast, is restricted by the GUI and can fall back on error correction methods. Therefore, it is more reasonable to let critical operations be handled via the sighted user. However, this issue is not a critical one for the system because it is designed as a reviewing tool. The visually impaired user should act as an adviser to the sighted graphic creator. The available active manipulation methods should allow the visually impaired reviewers to give a better judgment of their own recommendations. With the collaborative approach, the results of the study show that the sighted partner performs the editing tasks, as the sighted user is typically faster than the blind user.

During evaluation of the system, it became clear that the heterogeneous interface types for the sighted and the blind partner are the main challenge for collaboration. For the sighted users, it was hard to understand the way their blind partner thought and interact. It became apparent that the provided visual tool for monitoring the tactile interface is essential for supporting collaboration.

Overall, both sighted and blind users do not trust the tactile visualization of the system. All had some concerns that the final image would look different when distributed as a hardcopy. This is due to the fact that the low resolution of the dynamic pin-matrix display is only as good as tactile embossers for Braille text. Normally tactile graphics are distributed as high resolution embossed print-outs with a resolution of about 20 dpi or as microcapsule based hardcopies with a resolution of about 300+ dpi.

Another reason for some of the uncertainties of the users is the way the dynamic tactile representation is created. By scaling down the screenshot image to a low resolution, interpolation between pixel regions occurs. As a result, thin lines will sometimes result in light areas and be rendered into lowered pins, even if the lines are solid black. Therefore, thin lines can disappear. This also happens for small or sharp solid objects, such as arrowheads, which can be deformed and unrecognizable in the tactile image.

However, creating a digital abstract model of a product through a computer aided design approach is a common task in today's professional work environment. This process also includes knowledge on how it would look and feel after producing the designed model as a real object. Certainly, useful knowledge on the appearance of virtual objects after final production is cultivated through practice and experience. The participants in the evaluation lack this experience. Furthermore, upcoming new technologies have the possibility to provide dynamic tactile displays with a higher resolution, which would make it easier to transfer the feeling of the digital image to a hardcopy version.

19.3.5.3 Observations on Collaborative Awareness

The focus awareness method works well, although there are also some problems. While blind users want to know which element is selected at any given time, they do not want to see the blinking frame during editing, as this frame can mask the shape and its surroundings. Based on this observation, the focus markings turn off when switched into editing mode. This also leads to the problem that the users sometimes lose the focused element after exploring the surroundings. Therefore, some functionality is required to remark the focus and to bring the focused element back to the center of the display.

The marking of the GUI focus for the blind user also presents problems. The blind users liked to be aware of getting tactually and auditory information about the activities of the sighted partner. To start informing the blind user about the sighted user's activities, the non-visual user interface has to be switched into a special mode, in which every GUI focus change is reported. In addition to the auditory feedback about the selected element and a tactually blinking marking, the focused element is brought to the center of the tactile display. By this, the user does not have to search for it. When discussing with their sighted counterparts, blind users forgot that they were still in this mode and started to explore the surroundings. During their exploration, the sighted user started to do some further editing and his new focused element was brought to the center of the tactile interface. This happened often and after such a 'jump', most blind participants lost their context and did not longer know where they were. A description of the current mode of the system was given every time in the top right corner of the display, although blind users did not utilize it a majority of the time.

Furthermore, focus-change reports from the GUI appear in a high frequency when the sighted user does some editing on the graphic. This is caused when a user performs a high amount of selection and deselection operations with a mouse. In such a situation, the blind partner immediately lost the context, which they found annoying. The sighted user felt also annoyed by the incessant auditory output while editing.

When collaborating, the sighted users often guided the blind users to a region of interest by using the tactile visualization monitor. The sighted users then asked the blind users about changes after giving a general explanation of about the content of the image. Sighted partners often not vetoed against change requests from the blind user, even if the sighted partners thought that these changes would not increase the quality of the resulting image.

19.3.5.4 Rating of the Produced Tactile Graphic Versions

As mentioned above, the conventional production of tactile graphics is only done by a sighted person. To measure the benefits of collaboration over the singular production of the graphics by one person, we gave eleven additional and independent blind users the graphics as tactile print-outs produced by the eight teams to rate their quality. The sighted partners alone created half of these graphics (pre versions). The other graphics were revised through collaboration with blind partners (post versions).

Based on the ratings of the independent blind assessors, we conclude that the approach of our collaborative workstation can improve the quality of tactile graphics (see Figs. 19.4 and 19.5). Nevertheless, the images created collaboratively were not always rated better than the images created by a single person. Especially in groups with experienced sighted tactile-graphic creators, the quality was sometimes worse if the blind partner made unqualified suggestions. Therefore, it would seem one requirement for successful collaboration is that the blind user should have some experience in working with the pin-matrix device. The blind partner needs some imagination about the rendering of the image on a tactile print-out to make appropriate improvements.

As the charts 19.4 and 19.5 demonstrate, sighted experts could benefit from an experienced blind user, but do not benefit from collaboration with a blind layman. However, sighted non-professionals do benefit from collaboration with blind users.

Regardless of the better rating for the pre versions of expert transcribers, the evaluation scenario did help sighted transcribers improve their work. For example, the inexperienced blind partners wanted to make Braille-text labels more recognizable by adding a solid frame around them. The sighted professionals were surprised by this request and added, against their better knowledge, outlines around the Braille-labels that were too thick (see Fig. 19.6). These massive outlines were a

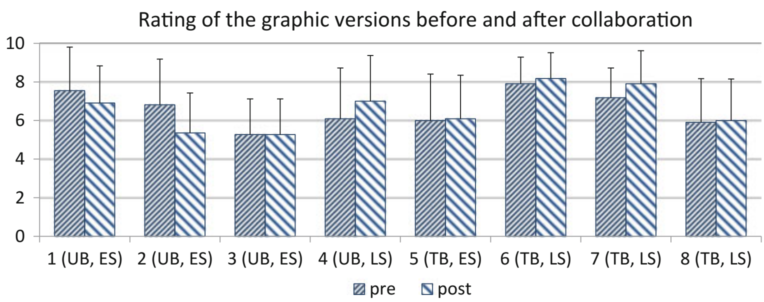


Fig. 19.4 Rating for the pre and post graphic versions given by eleven blind assessors on a scale from 0 (very bad) to 10 (very good). Means and standard deviations are also given. The eight collaborative teams consisted of either an untrained (UB) or a trained (TB) blind user, as well as a sighted laymen (LS) or graphical expert (ES)

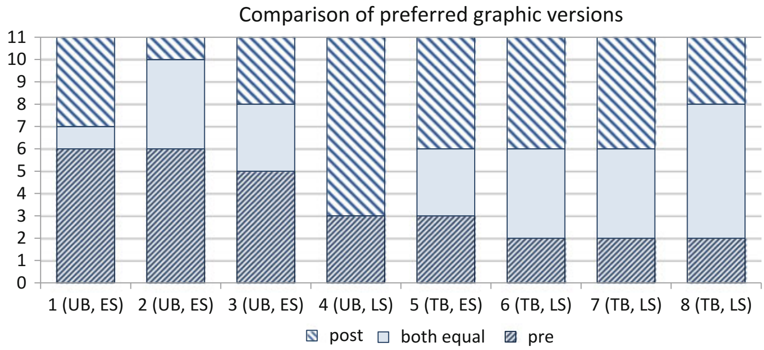


Fig. 19.5 Count of how often certain graphic versions are preferred based on the ratings of eleven blind assessors (see Fig. 19.4). *Legend*: untrained blind (*UB*), trained blind (*TB*), sighted layman (*LS*), graphic expert (*ES*)

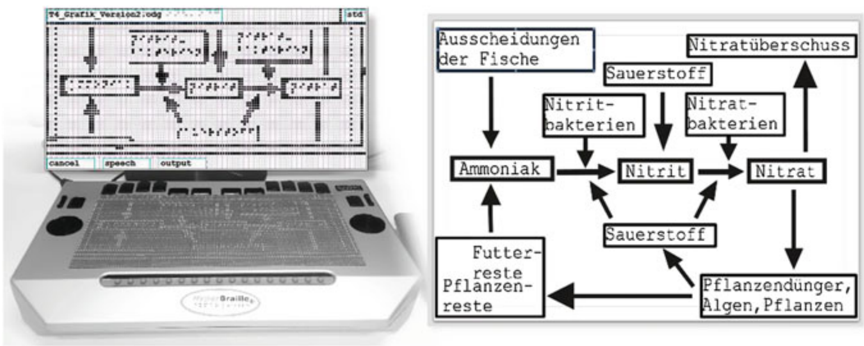


Fig. 19.6 Example of a transcribed image that was extended with *thick outlines* surrounding the Braille-labels. Document view in LibreOffice Draw (*right*), Tangram debug monitor (*top-left*) and Tangram tactile display output (*left-bottom*)

main cause of the overall devaluation of the resulting tactile hardcopy in the later assessment.

To ensure effective production of tactile graphics within a collaborative system, the blind partner should be adequately qualified to be able to fulfill his role as quality manager. This means that the blind partner should not only adapt an image to their own needs and preferences, but should also include general requirements for the whole group of blind readers. Also, the sighted partners should be more critical against change requests by contributing their own knowledge.

19.4 Recommendations

Mutual respect towards the user with the most limited abilities helps to let no participant feel left behind in a collaborative scenario. This recommendation is not focused solely on people with handicaps; it also addresses users with limited technical functionalities, such as restrictions for in- and output devises, small screen sizes, and low network bandwidths. Respect can only be cultivated if other users participating in the collaboration are aware of these limitations. Finally, of course, this can bring down the pace of innovation but let all members should equally participate in the collaborative process—which, in the end, is the main reason for applying collaborative approaches. We propose some general recommendations for visual to non-visual collaborative systems based on current literature as well as on our experiences with the Tangram system:

- To support collaboration, all interfaces have to be coherent [21] and synchronized with as low latency as possible.
- Awareness of the other person in the collaborative environment is important to prevent destructive situations [5]. For this purpose, the focused elements, current position in a document or environment, or the status of other members should be available.
- Guiding and indexing functionalities should be included to support discussions, especially in a complex context. If one partner gets lost or is not able to find the objects of interest, there have to be tools to help other participants assist him. This can be done, for example, by some 'gravity' function, which would allow the lost user to be actively captured. The other users should at least become aware of the position of the lost user to be able to provide verbal guidance.
- Every participant should have the possibility to interact independently with the system on his own device [12]. This enables all users to participate actively.
- In a collaborative context, conversation and discussion are desirable. For this reason, the usage of auditory output exclusively is not useful. However, within a multimodal context, auditory output can help users become aware of changes. Therefore, it is not always necessary that the auditory feedback is fully understandable [10]. Nevertheless, the possibility to review important or critical system outputs, for example by providing an activity log, helps to keep in track with the evolving product.
- Critical functions have to be protected in order to prevent the destruction of the collaborative result. Therefore, the system itself should be as stable as possible. Undo and redo functionalities for all participants' edits help to prevent losing important changes in the document.

Special recommendations for non-visual interfaces in collaborative environments are as follows:

- Talking about objects and explanations can be helpful for understanding a graphic as a whole. However, tactile feedback seems to be necessary to allow for an independent exploration of (especially graphical) content. Both modalities should be combined in a collaborative scenario to overcome limited tactual information reception capabilities, for example by verbal explanations from other users.

- Verbal guidance, especially in giving directions, distances, or interaction instructions, often leads to misunderstanding between the partners. On the one hand, active guidance, such as changing the focused object or the position inside a document by another user, can overcome such problems; on the other hand, the active and external change of the non-visual output tends to lead to a loss of context for the visually impaired user.
- A reliable reference system for maintaining context is necessary [13]. Additional functionalities, such as position markers or a grid, can help maintain this context for a blind user.

19.5 Conclusion and Outlook

In this article, a collaborative workstation for the creation of vector based tactile graphics is presented. The Tangram workstation turns a common single-user drawing application into a collaborative environment by extending it with an independent non-visual interface. The new interface enables a blind user to access not only the graphical data of the drawing, but also the object structure of the document itself. The main idea behind the workstation is to enable a blind user to take part in early stages of tactile image creation. Beyond pure observation, several possibilities and functions for editing and supporting collaboration were added. An important tool in context of collaborative environments is the awareness of the other partner's focus, which is realized by special visual or tactile presentations on the two interfaces. The concept of sharing information about the individual focused elements, supports also the possibility to index to elements and, thereto, to support conversation among the sighted and visually impaired partner.

The highly heterogeneous modalities of information presentation and interaction made it necessary to help the sighted user achieve an understanding of the tactile image presented to his blind partner. As a result, a monitor for displaying the tactile view to the sighted partner was installed.

Based on literature and practical experiences with the system, some general recommendations for building a visual to non-visual collaborative system were set up. These cannot only be applied to non-visual interfaces, but can also be transferred to other domains in which different users with different information modalities or input capabilities are combined. The recommendations are not limited to handicapped people. In addition, users using different devices for collaborative applications can benefit from these recommendations, for example, in a mobile to desktop context.

The Tangram system became a valuable educational tool here at the University of Dresden. It is used for the production of tactile graphics for textbooks and lecture material, as well as for explaining graphical content to blind students, quickly and easily sketching issues, and even allowing blind students to fulfill graphical education tasks.

Ultimately, a sighted user is necessary to create the drawings. In the future, we hope to allow blind users to use the system independently of a sighted user. For transforming the workstation into a more independent, non-visual drawing application, functionalities for creating and deleting elements have to be provided and protected by some kind of confirmation strategies and backup mechanisms.

The Tangram workstation furthermore demonstrates that blind people can be included in complex collaborative tasks and become valuable members. Hopefully this project provides the motivation to further include blind people in equivalent challenging fields.

Acknowledgements The Tangram project was sponsored by the Federal Ministry of Labour and Social Affairs (BMAS) under the grant number R/FO125423. We would also like to thank all blind and sighted participants, as well as Dillon Montag for his editorial suggestions.

References

1. Edman PK (1992) Tactile graphics. AFB Press, New York
2. Braille Authority of North America and Canadian Braille Authority (2012) Guidelines and standards for tactile graphics, 2010—Web Version, Feb 2012. http://www.brailleauthority.org/tg/web-manual/index.html. Accessed 24 June 2016
3. Prescher D, Bornschein J, Weber G (2014) Production of accessible tactile graphics. In: Computers helping people with special needs, pp 26–33
4. Winberg F (2006) Supporting cross-modal collaboration: adding a social dimension to accessibility. In: Haptic and audio interaction design. Springer, Berlin, Heidelberg, pp 102–110
5. Dourish P, Bellotti V (1992) Awareness and coordination in shared workspaces. In: Proceedings of the 1992 ACM conference on computer-supported cooperative work, pp 107–114
6. JAWS Tandem quick start guide, Freedom Scientific, Inc. http://www.freedomscientific.com/JAWSHq/JAWSTandemQuickStart. Accessed 9 Mar 2016
7. Köhlmann W, Lucke U (2015) Alternative concepts for accessible virtual classrooms for blind users. In: IEEE 15th international conference on advanced learning technologies, pp 413–417
8. Edwards WK, Mynatt ED, Stockton K (1995) Access to graphical interfaces for blind users. Interactions 2(1):54–67
9. Plimmer B, Crossan A, Brewster SA, Blagojevic R (2008) Multimodal collaborative handwriting training for visually-impaired people. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, New York, pp 393–402
10. Moll J, Sallnäs Pysander E-L (2013) A haptic tool for group work on geometrical concepts engaging blind and sighted pupils. ACM Trans Access Comput 4(4):14
11. Sallnäs E-L, Moll J, Severinson-Eklundh K (2007) Group work about geometrical concepts among blind and sighted pupils using haptic interfaces. In: Second joint EuroHaptics conference and symposium on haptic interfaces for virtual environment and teleoperator systems, IEEE, pp 330–335

12. Tanhua-Piiroinen E, Pasto V, Raisamo R, Sallnäs E-L (2008) Supporting collaboration between visually impaired and sighted children in a multimodal learning environment. In: Haptic and audio interaction design. Springer, Berlin, Heidelberg, pp 11–20
13. Sallnäs E-L, Bjerstedt-Blom K, Winberg F, Severinson-Eklundh K (2006) Navigation and control in haptic applications shared by blind and sighted users. In: Haptic and audio interaction design. Springer, Berlin, Heidelberg, pp 68–80
14. McGookin D, Brewster S (2007) An initial investigation into non-visual computer supported collaboration. In: Extended abstracts on human factors in computing systems. ACM, New York, pp 2573–2578
15. Rassmus-Gröhn K, Magnusson C, Eftring H (2007) AHEAD—audio-haptic drawing editor and explorer for education. In: IEEE International Workshop on Haptic Audio Visual Environments and their Applications, pp 62–66
16. Fujiyoshi M, Fujiyoshi A, Osawa A, Kuroda Y, Sasaki Y (2014) Development of synchronized CUI and GUI for universal design tactile graphics production system BPLOT3. In: Computers helping people with special needs, vol 8548. Springer International Publishing, pp 18–25
17. Bornschein J, Prescher D, Weber G (2015) Collaborative creation of digital tactile graphics. In: Proceedings of the 17th international ACM SIGACCESS conference on computers and accessibility. ACM, New York, pp 117–126
18. Prescher D (2014) Redesigning input controls of a touch-sensitive pin-matrix device. In: Proceedings of the international workshop on tactile/haptic user interfaces for tabletops and tablets
19. Jacob RJK (1991) The use of eye movements in human-computer interaction techniques: what you look at is what you get. ACM Trans Inf Syst (TOIS) 9(2):152–169
20. Bornschein J, Prescher D, Weber G (2015) Inclusive production of tactile graphics. In: Human-computer interaction–INTERACT, Springer International Publishing, pp 80–88
21. Mynatt ED, Weber G (1994) Nonvisual presentation of graphical user interfaces: contrasting two approaches. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, New York, pp 166–172

Chapter 20
Rich Digital Collaborations in a Small Rural Community

Alan Dix, Alessio Malizia, Tommaso Turchi, Steve Gill, Gareth Loudon, Richard Morris, Alan Chamberlain and Andrea Bellucci

Abstract In this chapter we describe experience in the design and installation of a low-cost multi-touch table in a rural island community. We discuss the creation of the table including: pragmatic challenges of installation, and then re-installation as the physical fabric of the multi-purpose building (café, cinema, meeting area and cattle market) altered; technical challenges of using off-the-shelf components to create state-of-the art multi-touch interactions and tactile BYOD (bring your own device) end-user programming; design challenges of creating high-production value bespoke mountings and furniture using digital fabrication in an environment that could include sewing needles, ketchup laden sandwiches and cow manure. The resulting installation has been used in semi-in-the-wild studies of bespoke applications, leading to understandings of the way small communities could use advanced interactions. More broadly this sits within a context of related studies of information technology in rural developments and a desire to understand how communities can become users of the rich streams of open data now available, and, perhaps more important, offer ways in which small communities can become empowered through the creation and control of their own data.

A. Dix (✉)
Talis and University of Birmingham, Birmingham, UK
e-mail: alan@hcibook.com

A. Malizia · T. Turchi
Brunel University, Uxbridge, UK

S. Gill · G. Loudon · R. Morris
Cardiff Metropolitan University, Cardiff, UK

A. Chamberlain
University of Nottingham, Nottingham, UK

A. Bellucci
Universidad Carlos III de Madrid, Madrid, Spain

C. Anslow et al. (eds.), *Collaboration Meets Interactive Spaces*,
DOI 10.1007/978-3-319-45853-3_20

20.1 Introduction

The widespread availability of touch and gesture sensitive displays has begun to transform many areas of life. In train stations large interactive timetables can be interrogated and in the museum and heritage sector touch-tables and other tangible technologies are emerging from research labs. Sometimes these displays are used in isolation, but they also may be used in conjunction with users' smartphones [12], or in assemblies of displays [38]. To some extent the use of touch and multi-touch technology in smartphones has both commoditised the underlying technology and changed public expectations about the nature of a display. Furthermore, many researchers have demonstrated the value of collaborations around large-scale touch-tables and similar surfaces [31], part of a broader research agenda looking at more 'natural' approaches to interaction incorporating, touch, tangible and other interaction modalities [42]. Unfortunately, large touch surfaces are still expensive, which restricts their use.

In this chapter we describe the design and installation of a low-cost multi-touch table in a rural island community. This demonstrates how existing technology can be used in a creative way to spread the benefits of interactive surfaces. In addition, it allows us to see potential uses and issues once the falling cost of dedicated multi-touch tables become more widely available.

We discuss the creation of the table including: (i) pragmatic challenges of installation, and then re-installation due to alterations in the physical fabric of the multi-purpose building (café, cinema, meeting area and cattle market); (ii) technical challenges of using off-the-shelf components to create state-of-the art multi-touch interactions and tactile BYOD (bring your own device) end-user programming; (iii) design challenges of creating high-production value, bespoke mountings and furniture using digital fabrication in an environment that could include sewing needles, ketchup laden sandwiches and cow manure.

The resulting installation has been used in semi-in-the-wild studies of bespoke applications, leading to understandings of the way small communities could use advanced interactions. More broadly this sits within a context of related studies of information technology in rural developments and a desire to understand how communities can become users of the rich streams of open data now available, and, perhaps more importantly, offer ways in which small communities can become empowered through the creation and control of their own data.

In the rest of this chapter, we first introduce the physical context of the installation: the Island of Tiree; its community: the Rural Centre within which the display is installed; and the potential for open data in small communities. We then describe the two phases of design and deployment of two versions of the interactive display, which differed in terms of the kinds of technology offered (multi-touch, tangible), the physical constraints (5 m vs. 2.4 m mounting), and production values ('DIY' installation vs. digital design and fabrication). Three semi-wild studies were conducted with these installations; we present some of the results and explore wider issues thrown up by the experiences during deployment and use.

20.2 Context—Tiree Island and Community

20.2.1 Demographics and Economics

Tiree is a small island off the west coast of Scotland. It has a land area approximately the same as Manhattan and a population of about 650 (c.f. Manhattan 1.6 million). By SIMD (Scottish Index of Multiple-Deprivation) metrics it is one of the most deprived areas in Scotland, alongside poorer parts of major urban areas [27], and is in the most deprived area in terms of access to services [1, 2].

Despite these gloomy statistics Tiree is a strong, resilient community with a school that caters for children to the end of secondary age (on many islands secondary pupils have to go the mainland weekly only returning home for weekends). Alongside tourism, rural industry is central, with one of the most well preserved crofting systems (small scale farming) using methods that help protect a rich natural environment.

Economic and social development are important issues for the island, particularly as the population shrank by about 15 % between 2001 and 2011 censuses. Population decline puts various services at risk. Of particular concern is the continued viability of the school, and with it, the attractiveness of the island to families. In 2010 the island community built a 950 kW wind-turbine ‘Tilley’, one of the most efficient in the world due to Tiree’s near constant wind. The income from Tilley helps fund other community projects, such as a feasibility study into the potential for a community purchase of land, large parts of which are owned by an historic estate.

The island has workable, albeit problematic, broadband infrastructure, about half of which is delivered by commercial ‘copper’ phone lines, and the remainder by a community company ‘Tiree Broadband’, which uses wireless links to reach outlying areas. Digital access has been identified as a major issue in Scotland, for both economic and social inclusion reasons, since there is a strong correlation between digital access and other deprivation factors [24, 32]. Without specific government action, digital technology tends to increase existing inequality. The Scottish Government have therefore instituted a programme to ensure optical fibre connectivity across the country, and, as this chapter is being written, the island is being connected through fibre to the mainland broadband networks.

20.2.2 Tiree Tech Wave and Tiree Rural Centre

Tiree Tech Wave is a twice-yearly maker/meeting event on the island. It attracts technologists, artists, product designers, and others interested in the way technology can be used in interesting and innovative ways, with a particular slant on rural issues. The Tech Wave is partly aimed towards participants: offering them a space to think innovatively, inspired by a wild and open environment; and partly towards the community as the long-term sustainability of remote communities will almost certainly involve increasing digital technology. Bridging the two is an education mission, helping participants to understand the information technology challenges for those at the physical margins.

The Tiree Tech Wave has led to numerous collaborations and other research benefits for participants, but also a number of more specific projects. One of these was *Frasan*, a Nesta funded mobile app [14] for the heritage centre, *An Iodhlann*, which houses 15,000 archive items. Another was OnSupply, a project led by Lancaster University looking at awareness of renewable energy availability [35]. In addition, there have been several projects connected with communications and data.

The Tech Wave is held in the Tiree Rural Centre, a building constructed as the cattle market. It is typical in rural areas to see buildings that are multi-functional. As well as the cattle sale ring, the Rural Centre includes a café, meeting space, public WiFi, and a tourist information point. The cattle sale ring itself converts into the island's cinema and lecture hall.

20.2.3 Open Data Islands and Communities

Many governments across the world have embraced open data [29] and in the UK the government-funded Open Data Institute promotes open data practices across civic society [28]. As well as national and governmental data, many large cities have adopted open data policies, and this has even extended to smaller local authorities [27]. However there are barriers, not least the expertise to use open data effectively. Ian Bartram, global manager for analytics at Gartner:

> I don't know if any public sector has necessarily cracked the nut on attracting the right skills and capabilities," … "The commercial sector has, because they've got the dollars to spend. [22].

The 'Open Data Islands and Communities' report [15] asks how open data could be made to work for smaller communities. There are many potential benefits:

i. easing communication within and between communities (see Fig. 20.1, flows 3 and 4);
ii. using public data for local action, external funding bids, and negotiation with external commercial or public bodies (flow 2); and
iii. perhaps most important of all, creating data locally that may be combined and used by others, shifting the community citizens from being simply data subjects to active data providers (flow 1).

However, the barriers are higher still than for local government since it would be rare to have suitable expertise in a community of a few hundreds or thousands of people. Various projects have addressed this on Tiree, in several cases leveraging the expertise brought by Tiree Tech Wave. These include a unified SMS and social

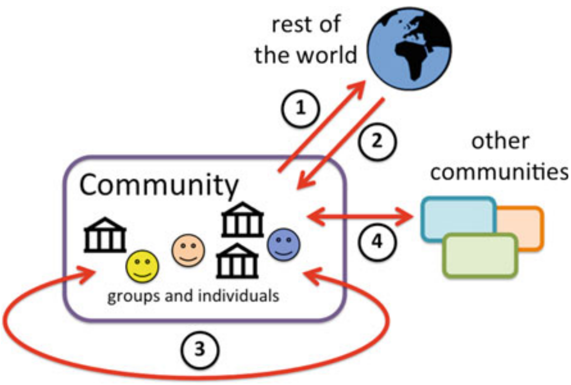


Fig. 20.1 Open data islands and communities—data flows

media portal for local youth work, a public 'ticker tape' display in the Rural Centre café, an internet enabled shop 'open sign', and a web dashboard.

The Tiree touch-table project is set within this context. Two studies were focused particularly on flow 2, the 'obvious' open data flow, using multi-touch interactions on a large projection to visualise and interact with existing data. However, even here we shall see that participants opened up discussion to look at wider flows. The final study, using tangible interactions, focused much more on the means for participants to create their own data flows.

20.3 Design and Installation—Phase 1

20.3.1 Touch-Table Software and Hardware

Various technologies have been used to enable touch and multi-touch surfaces. Electrical solutions, such as the capacitive displays in most smartphones, that embed electronics into the display surface do not scale well [43]. Optical solutions found in many large commercial and DIY surfaces make use of interference of infrared light on a semi-transparent surface detected by a rear-view camera [19]. While the latter can scale to very large displays, they are bulky and need surface instrumentation (e.g. special semi-transparent material with a fixed projector and infrared camera).

In 2010, Andrew Wilson from Microsoft suggested that new publicly available depth-sensing cameras, such as the Microsoft Kinect, could be used as inexpensive touch sensors [43] to overcome the limitations of capacitive and other optical approaches. An initial calibration phase provides a 3D map of the fixed surfaces, so when a finger, hand or pointer is detected in contact with the surface using depth estimation, it is interpreted as a 'touch'. Because it is a full vision-based method, any number of touch points can be tracked simultaneously (see Fig. 20.2).

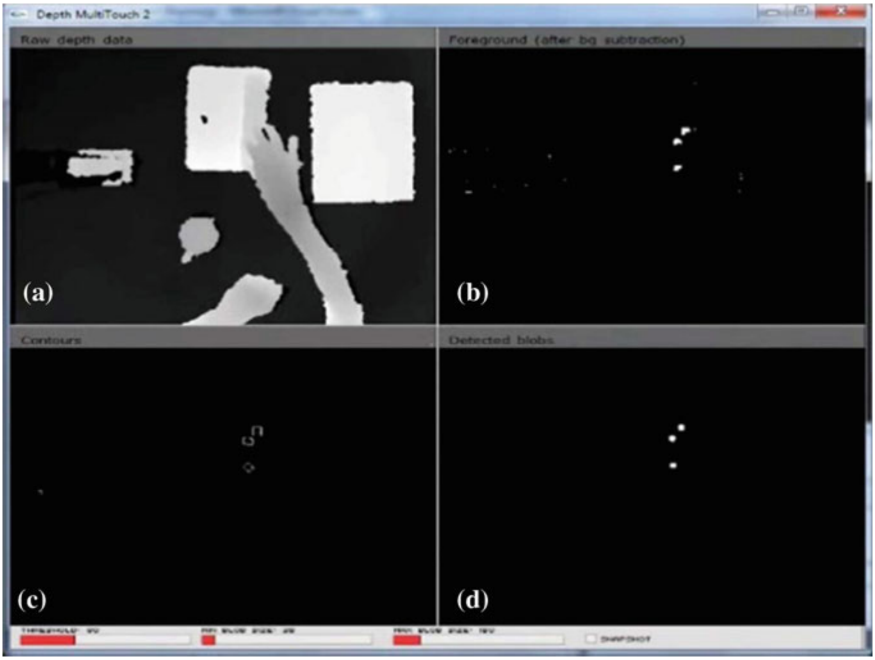


Fig. 20.2 TESIS using depth information to enable touch input **a** raw depth map; **b** subtract background and threshold to points within small distance of surface; **c** transform to connected regions; **d** recognized touch points

TESIS (Turn Every Surface into an Interactive Surface) was developed using this principle at the DEI Lab at University Carlos the III of Madrid [3, 4]. TESIS had three advantages over touch-tables available at the time: cost, portability and the flexibility to make ad hoc use of existing surfaces.

As well as using an off-the-shelf sensor (the Kinect), TESIS made use of open source software components: (i) openNI [30], which interprets the Kinect depth data; (ii) openCV for touch recognition; and (iii) openFrameworks for tracking fiducial markers to allow forms of tangible interaction such as the ReacTIVision amoeba codes [33]. In addition, openCV provided support for the TUIO (Tangible User Interface Object) protocol [23], an open protocol (and the de facto standard) allowing device-independent access to tangible and multi-touch tabletops. The openNI framework was chosen because it was well documented and offered support to different depth sensors. It also integrates better with other open-source software than the official MS Kinect SDK and benefits from a large community of developers.

The initial physical deployment used a micro-projector and Kinect co-mounted on an adjustable desk lampstand. The stand allowed the projector and Kinect to be positioned above any surface, transforming it into an active desktop, not unlike Wellner's [41] early DigitalDesk envisionment. One of the authors, AM, brought this to the Spring 2012 Tiree Tech Wave, giving rise to the idea of a permanent installation in the Rural Centre.

20.3.2 Physical Installation

Many meetings at the Rural Centre take place in the 'foyer', an area that also serves as a tourist information and WiFi access area during summer months, and seemed the ideal spot to deploy a large version of TESIS. A large table is usually positioned towards the centre of the area, directly below the apex of the roof, which is about 5 m from floor level. This meant that a large projector could be situated well out of the way, and connected to the girders that formed the ridge.

The deployment was carried out over a week by two of the authors, AB and AD. Part of the time was dedicated to software installation, running and testing at ground level, but the majority of the time was spent creating a platform to be installed at the 5 m ridge. To ensure a strong light contrast, a 3500 lm projector was chosen which was correspondingly heavy. This had to be mounted together with a Mac mini to run the software and the Kinect. The projector was mounted horizontally so a mirror was arranged off one end of the platform.

A critical design consideration was safety. Both adults and children use the area and the fear of a heavy projector or sharp-edged Mac mini falling on a child's head led to deliberate over-design. As the projector platform was quite sizable (about 70 cm^2, see Fig. 20.3 left), it needed to be designed to be bolted in position.

Another practical design consideration was the height of the Kinect. While the projected image could be adjusted to be table size, from a distance of 5 m, the Kinect's effective range was only about 1.5 m above a standard table height. Because Kinect precision deteriorates exponentially with distance and empirical tests demonstrated that a bigger distance would not provide the required precision, the Kinect had to be suspended half way down from the roof apex where the rest of

Fig. 20.3 Phase 1, (*left*) projector platform being constructed, note mirror cantilevered from the platform, and (*right*) installation at 5 m apex of Rural Centre roof

the equipment was mounted. A long, adjustable T-piece was constructed from timber with the Kinect mounted at the lower end. Adjustments in increments of 10 cm were possible, partly to allow us to experiment with different heights, and partly because we wanted to make it possible to store it out of the way to avoid accidental damage (recall that this building is a work place including its designed purpose for cattle sales).

The eventual design was somewhat 'Heath Robinson',[1] but the lengthy preparations proved successful and the entire assembly was installed, deployed and tested in one day at the end of the week.

20.4 Design and Installation—Phase 2

20.4.1 Physical Re-design

After completion of phase 1, the Tiree Rural centre gained funding for the installation of a false ceiling in the foyer area to make it warmer and more suitable for meetings. This was good news for the Rural Centre, but meant that the projector installation had to be completely removed and re-designed for a ceiling height of 2.4 m. Staff and students from the Cardiff School of Art & Design (CSAD) took on the re-design and installation of TESIS' next iteration at a subsequent Tiree Tech Wave event. The re-furbished Rural Centre had a more sophisticated feel than before, meaning that the previous utilitarian approach would no longer fit in either sense of the word. What was now required was something more akin to a fully developed product. This was a challenge. Tiree is located 550 miles (885 km) from CSAD's well-equipped base, and the available manufacturing facilities limited. Lateral thinking was required in five major domains: cost, understanding, time, design and manufacture.

Cost: A large part of a design project's costs lies in the designer time required. This project was no exception. However, CSAD used the touch-table project as a teaching tool. This made both economical and pedagogical sense, so it was written into the Product Design M.Sc. for 2014–2015. Material costs were met by Tiree Tech Wave and manufacturing costs could be kept to a minimum provided we could work out how to construct and install the designs in a remote area with very limited resources.

Understanding: The next issue was how to give students an insight into how people live on Tiree, including the community use of the Rural Centre and the space itself. The only effective solution was to bring them to the island. CSAD Product Design students are taught ethnographic research principles along the lines

[1]W. Heath Robinson (1872–1944) drew images of complex machines with a superfluity of levers, cogs, wheels, and pulleys, to perform mundane and often not very useful purposes. These are called Rube Goldberg devices in the United States.

advocated by Hammersley and Atkinson [18] and Squires and Byrne [36]. They put theory into practice by conducting interviews with members of the community. Their key findings were:

1. The centre is used in multifarious ways and configurations;
2. The table in particular must be robust and should seat 12;
3. The projection module must not get in the way when offline;
4. There needed to be a way to link a laptop to the projection module;
5. The table should accommodate storage;
6. The table and projection module needed to allow vertical and horizontal projections.

Time: Tiree Tech Waves last for 4 days; not long to research, design, prototype, test, complete and install two designs. By bringing the efforts of six students and three members of staff to bear we used a lot of people hours in a short space of time. This was further enhanced by the student cohort who put in many hours of 'overtime'.

Design: The cohort was divided into two teams. A user-centred design approach was followed [26, 40]. Our approach also emphasised the role of relaxation and play [25] and the importance of the physical prototype [17]. Using research insights to form an appropriate brief, both teams began by producing concepts individually.

They then brought their ideas together, consulted with the community to select the strongest proposals and refined them by combining the best features. Tasks were then divided amongst the team to maximise efficiency. Sketches and iterative prototypes were produced throughout (Fig. 20.4).

Manufacture: Prototyping facilities on Tiree are limited (see above). Our solution was to bring FabLab Cardiff [16] and its manager, to Tiree. There was an obvious limit to the number of prototyping tools that could be economically and practically imported, so equipment was limited to two 3D printers, a laser cutter, a small CNC machine, a CNC vinyl cutter, hand tools, battery drills, some electronics

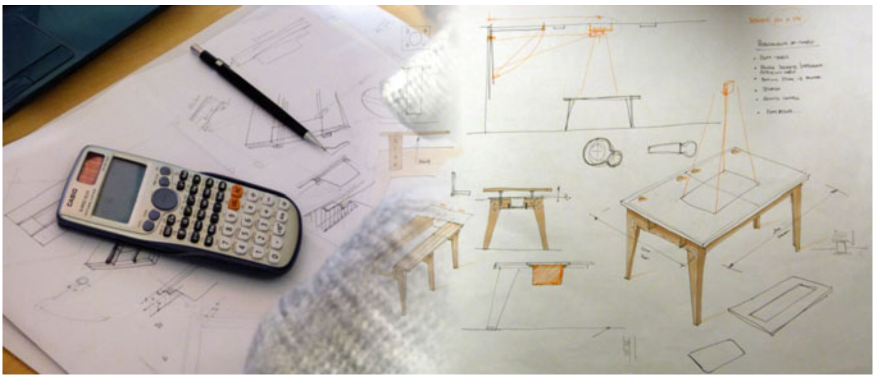


Fig. 20.4 Sketch development work

Fig. 20.5 FabLab Cardiff in the Tiree Rural Centre Cattle Market/Cinema

prototyping kit and a Perspex bender. Material included card, dowel, laser ply, Perspex and other plastics, MDF and modelling foam (Fig. 20.5). A small budget allowed for the local purchase of additional supplies.

The manufacturing limitations had one immediate effect on the group designing the table: the CNC to manufacture it was too large to be transported. We reasoned however that digital files are scalable, so as long as the design was proven at scale, full size manufacture at a later date should be straightforward.

The finalised table design is a sturdy product designed for rough handling. It is height adjustable to allow seated or standing use, and the projection surface can be removed and wall-hung to form a projection screen. Removing the projection surface also exposes storage trays so that the community can keep frequently accessed items safely and neatly stored. The projection module is ceiling mounted with wiring fed into the loft. A pico projector sits to the side of the main chassis on a swivel joint that allows it to project downwards onto the table or horizontally onto a screen. The Kinect is co-mounted on the swivel to orient wherever the projector is pointed (Fig. 20.6).

Fig. 20.6 Finished table model and projection module

Fig. 20.7 The full-sized table in situ

Following modifications, the CAD files were used to produce a full sized variant of the table, which was assembled and installed during a subsequent Tiree Tech Wave (Fig. 20.7).

20.4.2 Tangible Software

With the second phase of physical installation completed and with the passage of time since the 2012 installation having given rise to a host of new software platforms, it was decided that the time was right to develop a second generation tangible software system. It was clear from the outset that the system had to be flexible.

Unfortunately, due to their public and moderated nature, Pervasive Display ecosystems do not usually provide a wide set of general and unfixed features, even though their user base is heterogeneous and evolving over time. Enabling users to adapt a system themselves could promote more serendipitous and prolonged usage [21], fostering their appropriation in contexts where frequent supervision over mundane maintenance and upgrade activities is not feasible. We theorised that this would be the case with the Tiree Rural Centre, which is why an End-User Programming-enabled approach was chosen.

End-User Programming provides us with design guidelines to enable users to adapt software systems to their needs, allowing them to exploit the computational capabilities enjoyed by professional programmers. By employing it together with an easy to use interaction modality, we designed a Pervasive Display system that can be deployed in public spaces and can be used by non-experts to repurpose the system to their own needs.

We also chose to exploit Tangible User Interfaces (TUIs). TUIs consist of a set of physical objects that users manipulate to interact with a computing system. They are a popular choice for interaction with Pervasive Displays and have proven very effective in making highly abstract activities such as programming, more direct and widely accessible; thus our decision to deploy one here.

The end result was a successor system to TESIS called TAPAS (TAngible Programmable Augmented Surface). TAPAS is a TUI-based BYOD End-User Programming system [39] comprising a horizontal tabletop display and a RGB camera capturing the movements of the users' smartphones on the main display's surface using fiducials [5]. We decided to exploit smartphones since they hold users' preferences and can display a wide range of widgets depending on the required input (e.g. a virtual keyboard to input text).

TAPAS allows users to develop simple workflows by composing different available services, where the output of one becomes the input of the next. We used a puzzle metaphor to communicate the services' control-flow since it is familiar to end users [10]: each puzzle piece represents a service which could require inputs and produce outputs, the type constraints of which are displayed using shapes. The smartphone itself is associated with a circle halo with a hollow to accommodate the next piece, which moves alongside the smartphone on the tabletop projected surface. Joining a suitable piece to it will add the latter's represented function to the user's workflow. If a single piece requires additional inputs from the user, a dynamic widget will appear on the lower half of the smartphone screen (varying accordingly to the type of input required, e.g. list menu or keyboard).

20.5 Studies

The software and hardware setups were tested through three studies, the first two using the phase 1 installation (TESIS) and the third after the phase 2 installation (TAPAS). Studies 1 and 2 are described in greater detail elsewhere [7, 8].

20.5.1 Study 1

The first study was with a small group of islanders who had responsibilities or interests related to future policy and investment in the island. They used the phase 1 installation as a large computer desktop display, showing a variety of documents, but principally a map of the island (Fig. 20.8).

The researchers began the session by giving a brief demonstration of the system, and introducing the topic of big data. Participants were also offered the ability to draw or write on the map (as it was projected onto a large sheet of paper, see Fig. 20.9).

Initially the group used the map to show where they lived, and to recognise key features and locations. Although perhaps a frivolous use of the system, it was important for participants to relax with the technology and get used to its features.

Fig. 20.8 Participants gathered around the tabletop display. Sometimes they split into small groups to discuss topics

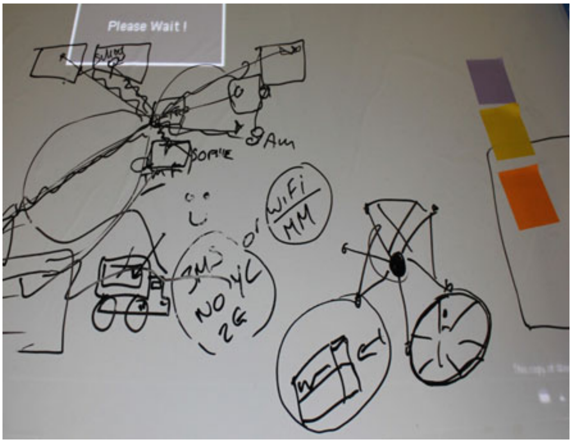


Fig. 20.9 Illustrations drawn directly on the projected surface (projection turned off)

They soon switched to an in-depth discussion of plans for a future island skate park using the system to plan the most appropriate location. Discussions included the availability of data concerning weather and how this might affect the materials used if it were to stand up to the harsh island climate. The participants also considered the way data could be aggregated to give a truer picture of island life.

The conversation shifted outwards, initially around comparisons with the neighbouring islands of Coll (similar physical size, but much smaller population) and Mull (substantially bigger size and population). The participants believed that a system such as this could potentially be used to communicate between related islands, where physical flights and ferry usually have to be via a hub port. They also considered its use with the diaspora, including the descendants of those who emigrated to Canada, Australia and elsewhere in the 19th century.

The participants drew on the map adding connectivity of roads, times and locations of shops, fast food, WiFi and 2G coverage (Fig. 20.9). This free form use, although not captured digitally, enabled us to understand the kinds of features that might be added.

The researchers introduced the issue of health planning and this led to further discussions and the need for effective data and figures on health on the island in order to prepare funding bids or similar projects. Currently this requires accessing information directly from the island surgery records with the aid of a local expert.

Fuel was also discussed, as it is particularly expensive on the islands. The researchers introduced the idea of community fuel purchasing, but the participants explained that the island garage is an important local business and so would be reluctant to introduce community-based competition. However, they were more interested in the use of data as part of an exploration of the additional costs of island living.

These discussions also led on to the way the island was sometimes asked for locally sourced information and the advantages if that information could be more readily available for external bodies, not only to ease this fact finding, but so that potential funders could more appropriately target the island. If such externally facing information were available the participants also wanted to be able to track where and how it had been used, reflecting concerns in the authors' own earliest writings on privacy and HCI [11].

20.5.2 Study 2

During Study 1, the participants suggested that the system would be useful during the launch of the Tiree Heritage app [14]. This involved using a full-screen version of a web browser to display the web-based application, which while designed primarily for small-screen devices adjusted to the larger screen and enabled shared discussions of what would otherwise have been individual interactions. Touch sensing was turned off for this and replaced by a wireless keyboard and trackpad. This was partly because the calibration could still be quite fragile if the table was moved, and it would not be possible to maintain this in a fully 'wild' situation.

The evening launch event involved around 30 people who gathered round the table in small groups, often using the map-based information as a catalyst for conversation and collaborative reflection (Fig. 20.10). It was this reflection upon places, people and history that displayed the power of the TESIS system at a

Fig. 20.10 Discussion around the map at the *Frasan* launch

community level when used with local maps. It is also important to allow this technological appropriation, because as Bucciarelli [6] writes, "…different participants in the design process have different perceptions of the design, the intended artifact, in process. […] The task of design is then as much a matter of getting different people to share a common perspective, to agree on the most significant issues, and to shape consensus".

In the next section, we further discuss some of the issues raised, including the importance of local mapping.

20.5.3 Study 3

The last study utilised the new hardware/TAPAS software installation. To get some feedback on TAPAS we interviewed three interaction design experts in a controlled environment during one of the Tiree Tech Waves. The study lasted 45 min and involved two HCI experts and a product designer. We introduced the prototype and explained the rationale behind its design, including the scenarios we are targeting; we then gave a brief demonstration, going through examples of its usage in a real world scenario on Tiree. We then proceeded with semi-structured interviews focusing on TAPAS's strengths and weaknesses in relation to its interaction modality and applicability in a context like the Tiree Rural Centre, where tourists and locals often meet up to get information about what is going on on Tiree.

The designers liked the idea and the use of a smartphone for personalization and tangible interaction, and recognized the potential of a cheap, available and easily deployed system in a public space. They also liked that it can be left for long periods of time without the need to perform maintenance operations or bring in experts to add new features, since users can repurpose it themselves, a particularly valuable feature in a remote setting such as Tiree.

Some of the participants' suggestions focused on the coupling between data visualization and the dynamic widget: Due to the type of data currently handled (directory and library books listings) it makes sense to restrict user prompts to lists or keyboard input. Nevertheless, dealing with more structured data types—such as points of interest on a map requires more flexible and personalisable widgets based on the two-folded level of interaction between the user and data perspectives.

Finally, interviewees pointed out how the continuous back and forth interaction between the smartphone and large display might confuse users since switching between tangible and multi-touch interaction styles requires extra cognitive effort. Instead they suggested making the tabletop the main interaction focus by providing a mixed interaction modality with the smartphone used to assemble the workflow, but using a multi-touch-enabled widget on the tabletop surface once an input is required. So while it was agreed that the system has clear strengths, such as low cost, ease of prototyping in the wild and the flexibility of the architecture, there are also some major challenges to be addressed in term of interaction design requirements, like the flexibility and programmability of the widgets.

20.6 Discussion

20.6.1 Practical Lessons

Many very practical issues drove aspects of the development of the Tiree touch-table installation. Some of these concerned the physical aspects of the space and equipment (5 m roof, entailing heavy projector), some more to do with the social setting (crayons and cow dung), and some about the relationship between the two (young children below large equipment). This has led to issues of safety including the installation process itself as well as the protection of equipment (retractable arm for the Kinect).

This kind of issue will be familiar to anyone who has created long-term installations. Similar issues occurred in the Lancaster eCampus project where projectors were installed in an underpass at the University, but suffered continual problems related to access, safety and shear dirt [37].

The changes in the building are also quite a normal part of a real setting; social, organisational and physical settings all change over time. In this case the physical change meant changing equipment (lightweight LCD projector instead of large high-power one) and also production values. While the foyer area roof was effectively that of an agricultural building, a rough-and-ready install was sufficient; but once the new ceiling had been installed, a higher standard of design was required.

As this is a real setting there were diverse stakeholders. Although we did not produce a classic 'rich picture' [9], it is clear that there are a wide variety of uses of the space (cattle sales, information point, meeting area, WiFi access) and each has a range of users. The needs and expectations of the more 'official' members of the community in study 1 are different from the 'general public' in study 2 and the 'experts' in study 3. One example of this is the design of the table. The initial model in Fig. 20.6 has a single 8 foot × 4 foot tabletop (2.4 m × 1.2 m), but the final design in Fig. 20.7 consists of two square sections. The students' client for the design was one of the Rural Centre directors. He felt that a single large table was sufficient; however subsequent conversations with actual users of the area (some elderly) suggested that moving around a single large table would be very difficult. The final compromise was a two-part table on lockable castors where the two halves clip together. Again conflicting requirements from different stakeholders is far from a new lesson for any practical design project, but can often be ignored for small-scale or lab-based experiments. Even though the deployment is partly for research purposes, it must still meet professional physical and digital design standards.

20.6.2 From Global Big Data to Local Small Data

The touch-table and projected map in study 1 were initially presented as a way for small communities to be able to access big data (Fig. 20.1, flow 2), for example, government open data. However, the participant discussions soon changed to looking at internal island data (flow 3), inter-community communications (flow 4) and the creation of data for external use (flow 1). This counter narrative of the importance of data, knowledge and wisdom of the community, was also evident in the physical marks they left behind.

The whiteboard markings in Fig. 20.9 are embodiments of local understanding linked to the external data and satellite view maps of the island. However, they are transient, ephemeral; the canonical external data persists, but the local knowledge is wiped with the cleaning of the surface. Although this may be all that is needed during a meeting, there often is also a need for a more persistent connection.

With a paper map, one might draw on the map, highlight areas, or add pins and thread to link points on the map with each other. With digital maps and data the 'external' view is privileged, being digital in many ways makes it more 'immutable' as well as more authoritative. For both maps and data we need ways to enable communities to easily annotate and augment 'official' big data with their own contextual small data. Semantic web technologies are a move in this direction in that they allow multiple statements to not just link to, but add data to existing resources in a way that is, in principle, on an equal footing [34]. To some extent, Linked Data, the more practical side of semantic web technology, has re-established a privileged source with more asymmetric relationships, but does still allow easy augmentation and linking [20]. None of this is yet in a form that is easy for ordinary users, however.

The map projected during the more formal sessions was a Google map, but on the wall is a large map of the island divided into ‘townships’. This division of the land is crucial both for local understanding of identity, and also for the crofting: The crofts in a township share common grazing rights, but these are not part of the external mapping of the island. In contrast, the Tiree Heritage app (*Frasan*), projected onto the surface in the evening session, uses ‘standard’ mapping in detailed views, but for the overall island view, adopts hand-created maps used in tourist information. These local maps, like tourist maps elsewhere, emphasise certain features and may ‘distort’ geometry for cartographic or aesthetic reasons [14]. Local maps embody a sense of local identity, challenging the uniform view of ‘standard’ maps.

Finally, recall that the participants wished that they were able to provide island data to outside bodies, so that they might be more visible to potential funders. This reminds us that the power of data cuts two ways.

On the one hand the consumption, visualisation and analysis of data is often easiest for those with large budgets and available expertise; that is data consumption may reinforce existing power relationships.

On the other hand, the production of data is typically asymmetric, with the powerful, whether government or multinational corporations, in the best place to provide information. If that data is easily accessible, then it is that which will frame discourse. Even if the data is factually correct, the choice of what to provide, the methods of collection, filtering and presentation, all reinforce an external normative viewpoint.

Making local data available globally, especially if connected with others as part of the ‘long tail of small data’ [13] means that the voice of local communities is more likely to be heard. Of course, this small data from large numbers of communities becomes big data, allowing local knowledge to contribute to large-scale understanding. This poses technical problems, including the need to deal with heterogeneous datasets. Crucially these technical challenges need to be seen in the light of the social and political implications they entail, for example, the need for the tracking of provenance, as highlighted by the participants in Study 1.

20.7 Conclusions

A significant part of this chapter has focused on the practical issues of deployment and installation. Any in situ long-term prototype has to deal with these kinds of issues, although they are perhaps particularly severe in a relatively remote location. Those deploying in the developing world often face harder issues, not least, lack of power. For Tiree the power supply is somewhat less stable than the mainland, but only with minor fluctuations and short outages. This is an issue we need to face however for future work. The aim is to have the table running permanently, especially through the summer months when the island has over 20,000 visitors.

During the studies we saw examples of all the data flows described in open data islands and communities; some existing, some potential. The tangible end-user framework, deployed as part of the phase 2 installation, has the potential to offer ways of manipulating external data and creating local data, but so far, has only been subject to experts' evaluation. More work is needed to make this useful for local needs.

There is always a tension when creating public installations between research goals and making it useful for those in the setting. When installing in a large municipal building, or university, the 'client' site often has a level of technical oversight. While there is frequently significant local expertise, this cannot be assumed. So when installing in local communities it is particularly important to be sensitive to local needs and not simply impose a solution because it is your latest, favourite technology. Of course this creates equal challenges in interpreting the research data as each setting is unique with specific stakeholders and issues.

We hope in the work reported here, and in our on-going research, that we can both be sensitive to the particular rich setting of Tiree, but also to learn more widely, socially and technically. In particular, we are aware that the regular presence of expertise in the Tiree Tech Wave is unusual, and so we wish to create re-usable technology that can be easily re-purposed to other settings and communities allowing each to express, in their own unique way, what it means to be a small community in an age of global data.

References

1. Argyll and Bute Council (2016) Understanding Argyll and Bute. https://www.argyll-bute.gov.uk/understanding-argyll-and-bute#deprivation. Accessed 27 Feb 2016
2. Bailey N, Spratt J, Pickering J, Goodlad R, Shucksmith M (2004) Deprivation and social exclusion in Argyll and Bute. Report to the Community Planning Partnership. Scottish Centre for Research on Social Justice, Universities of Glasgow and Aberdeen, Feb 2004
3. Bellucci A, Malizia A, Aedo I (2011) TESIS: turn every surface into an interactive surface. In: Proceedings of the ACM international conference on interactive tabletops and surfaces
4. Bellucci A, Malizia A, Aedo I (2014) Light on horizontal interactive surfaces: input space for tabletop computing. ACM Comput Surv 46(3):32

5. Bonnard Q, Lemaignan S, Zufferey G, Mazzei A (2013) Chilitags 2: Robust fiducial markers for augmented reality and robotics. http://chili.epfl.ch/software. Accessed 27 June 2016
6. Bucciarelli LL (1984) Reflective practice in engineering design. Des Stud 5(3)
7. Chamberlain A, Malizia A, Dix A (2014) Visual and tactile engagement: designing projected touch-surfaces for community use in a rural context. In: AVI 2014, the 12th international conference on advanced visual interfaces, Como (Italy), 27–30 May. ACM Press. doi:10.1145/2598153.2598202
8. Chamberlain A, Malizia A, Dix A (2013) Engaging in Island Life: big data, micro data, domestic analytics and smart islands. In: HiCUE 2013: workshop on human interfaces for civic and urban engagement, Ubicomp 2013, Zurich, Switzerland. ACM Press. doi:10.1145/2494091.2495994
9. Checkland P, Poulter J (2006) Learning for action: a short definitive account of soft systems methodology and its use for practitioners. Wiley
10. Danado J, Paternò F (2012) Puzzle: a visual-based environment for end user development in touch-based mobile phones. In: Proceedings of the 4th international conference on human-centered software engineering
11. Dix A (1990) Information processing, context and privacy. In: Cockton DG, Shakel B (eds) Human-computer interaction—INTERACT'90. North-Holland, pp 15–20. http://alandix.com/academic/papers/int90/
12. Dix A, Sas C (2010) Mobile personal devices meet situated public displays: synergies and opportunities. Int J Ubiquit Comput (IJUC) 1(1):11–28. http://alandix.com/academic/papers/MPD-SPD-2010/
13. Dix A (2011) In praise of inconsistency—the long tail of small data. Distinguished alumnus seminar, University of York, UK, 26 Oct 2011. http://www.hcibook.com/alan/talks/York-Alumnus-2011-inconsistency/
14. Dix A (2013) Mental geography, wonky maps and a long way ahead. GeoHCI, Workshop on geography and HCI, CHI 2013. http://alandix.com/academic/papers/GeoHCI2013/
15. Dix A (2014) Open data islands and communities. http://tireetechwave.org/projects/open-data-islands-and-communities/. Accessed 27 June 2016
16. FabLab (2016) FabLab Cardiff. http://www.fablabcardiff.com/. Accessed 27 June 2016
17. Gill S, Dix A (2012) The role of physicality in the design process. In: Adenauer J, Petruschat J (eds) Prototype! Physical, virtual, hybrid, smart—tackling new challenges in design and engineering. Form + Zweck
18. Hammersley, Atkinson (1994) Ethnography: principles and practice. Raithledge
19. Han JY (2005) Low-cost multi-touch sensing through frustrated total internal reflection. In: Proceedings of the 18th annual ACM symposium on user interface software and technology, pp 115–118
20. Heath T, Bizer C (2011) Linked data: evolving the web into a global data space (1st edn). In: Synthesis lectures on the semantic web: theory and technology, vol 1, no 1. Morgan & Claypool, pp 1–136
21. Hosio S, Goncalves J, Kukka H, Chamberlain A, Malizia A (2014) What's in it for me: exploring the real-world value proposition of pervasive displays. In: Proceedings of the international symposium on pervasive displays
22. Howarth B (2014) Big data: how predictive analytics is taking over the public sector. Guardian, 13th June 2014. http://www.theguardian.com/technology/2014/jun/13/big-data-how-predictive-analytics-is-taking-over-the-public-sector. Accessed 27 June 2016
23. Kaltenbrunner M, Bovermann T, Bencina R, Costanza E (2005) TUIO: a protocol for table-top tangible user interfaces. In: Proceedings of the 6th international workshop on gesture in human-computer interaction and simulation
24. Longley P, Singleton A (2009) Linking social deprivation and digital exclusion in England. Urban Stud. 46(7):1275–1298. doi:10.1177/0042098009104566
25. Loudon G, Deininger G, Wilgeroth P (2012) The importance of play and creativity in the design curriculum. In: International conference on engineering and product design education, Antwerp, 6–7 Sept 2012

26. Luchs M, Swan S, Griffin A (2015) Design thinking: new product development essentials from the PDMA. Wiley-Blackwell
27. Nesta (2014) Open data Scotland. http://www.nesta.org.uk/project/open-data-scotland. Accessed 27 June 2016
28. Open Data Institute (2015) Open data roadmap for the UK, 2015. http://theodi.org/roadmap-uk-2015. Accessed 27 June 2016
29. Open Knowledge Foundation (2016) Open data handbook. http://opendatahandbook.org. Accessed 27 Feb 2016
30. OpenNI (2016) OpenNI resource page. http://structure.io/openni. Accessed 27 June 2016
31. Rogers Y, Lim Y, Hazlewood W (2006) Extending tabletops to support flexible collaborative interactions. Proceedings of Tabletop 2006, 5–7 Jan 2006. IEEE, Adelaide, Australia, pp 71–79
32. Royal Society of Edinburgh (2013) Spreading the benefits of digital participation: an interim report for consultation. Royal Society of Edinburgh, Edinburgh, pp 22–24
33. ReacTIVivsion (2016) ReacTIVision—a toolkit for tangible multi-touch surfaces http://reactivision.sourceforge.net. Accessed 27 June 2016
34. Shadbolt N, Hall W, Berners-Lee T (2006) The semantic web revisited. IEEE Intell Syst
35. Simm W, Ferrario M, Friday A, Newman P, Forshaw S, Hazas M, Dix A (2015) Tiree energy pulse: exploring renewable energy forecasts on the edge of the grid. In: Proceedings of the 33rd annual ACM conference on human factors in computing systems. ACM, New York, NY, USA, 1965–1974. doi:10.1145/2702123.2702285
36. Squires S, Byrne B (eds) (2002) Creating breakthrough ideas: the collaboration of anthropologists and designers in the product development industry. Greenwood Press
37. Storz O, Friday A, Davies N, Finney J, Sas C, Sheridan J (2006) Public ubiquitous computing systems: lessons from the e-Campus display deployments. IEEE Pervasive Comput 5(3):40–47. doi:10.1109/MPRV.2006.56
38. Terrenghi L, Quigley A, Dix A (2009) A taxonomy for and analysis of multi-person-display ecosystems. J Personal Ubiquit Comput 13(8):583–598. doi:10.1007/s00779-009-0244-5
39. Turchi T, Malizia A, Dix A (2015) Fostering the adoption of pervasive displays in public spaces using tangible end-user programming. In: IEEE symposium on visual languages and human-centric computing, At Atlanta, Georgia, US, 18–22 Oct 2015
40. Ulrich K, Eppinger S (2011) Product design and development. McGraw-Hill Higher Education
41. Wellner P (1993) Interacting with paper on the DigitalDesk. Commun ACM 36(7):87–96. doi:10.1145/159544.159630
42. Wigdor D, Wixon D (2011) Brave NUI world: designing natural user interfaces for touch and gesture, 1st edn. Morgan Kaufmann Publishers Inc.
43. Wilson A (2010) Using a depth camera as a touch sensor. In: ACM international conference on interactive tabletops and surfaces, pp 69–72