Cross-Surface

Interacting with Multi-Device Ecologies in the Wild

2015

Proceedings of the first Cross-Surface workshop

Editors: Steven Houben, Jo Vermeulen, Nicolai Marquardt, Johannes Schöning, Clemens Klokmose and Harald Reiterer

Publication: Published online in 2016.

Contact: **Steven Houben**
Department of Computer Science
University College London
Gower Street, London WC1E 6BT
United Kingdom

About

In this workshop, we reviewed and discussed opportunities, technical challenges and problems with cross-device interactions in real world interactive multi-surface and multi-device ecologies. We aim to bring together researchers and practitioners currently working on novel techniques for cross-surface interactions, focusing both on technical as well as interaction challenges for introducing these technologies into the wild, and highlighting opportunities for further research. The workshop will help to facilitate knowledge exchange on the inherent challenges of building robust and intuitive cross-surface interactions, identify application domains and enabling technologies for cross-surface interactions in the wild, and establish a research community to develop effective strategies for successful design of cross-device interactions. Please find more details about the workshop, in the submitted proposal [1]. The workshop was held in conjunction with the 2015 ACM International Conference on Interactive Tabletops and Surfaces, that took place from November 15 to 18 in Funchal in Madeira, Portugal.

[1] Steven Houben, Jo Vermeulen, Clemens Klokmose, Nicolai Marquardt, Johannes Schöning, and Harald Reiterer. 2015. Cross-Surface: Workshop on Interacting with Multi-Device Ecologies in the Wild. In Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS '15). ACM, New York, NY, USA, 485-489. *DOI*: http://dx.doi.org/10.1145/2817721.2835067

Editors

Dr. Steven Houben – *University College London*
Dr. Jo Vermeulen – *University of Calgary*
Dr. Nicolai Marquardt – *University College London*
Prof. Johannes Schöning – *Hasselt University – tUL –iMinds*
Prof. Clemens Klokmose – *Aarhus University*
Prof. Harald Reiterer – *University of Konstanz*

Program

09:00 Introduction to workshop by the organizers
09:15 Keynote by Professor Yvonne Rogers
10:00 Paper Presentations
10:30 Coffee break
11:00 Brainstorm based on prepared insights and patterns
13:30 Session 1: Use cases in the real world
14:15 Reflections on session 1
14:30 Session 2: From lab technologies to real-world solutions
15:00 Coffee Break
15:30 Reflections on session 2
15:45 Session 3: Beyond interaction techniques and social Issues
16:30 Reflections on session 3
16:45 Closing

Keynote

Title: "Research in the Wild"

There have been several turns in Interaction Design. Most notable has been 'a turn to the social', 'a turn to design' and a 'turn to experience'. Each has called for a new way of framing research and conceptualizing its discourse. During the last 10 years, there has been a move towards doing more 'in the wild'; be it deploying and evaluating new technologies in situ or observing whatever is happening out there. As part of this 'turn to the wild' I will discuss the challenges of doing research in the wild – illustrating the new discoveries that can be achieved together with the tensions and problems that can arise when giving up control.

Biography:

Yvonne Rogers is a Professor of Interaction Design, the director of UCL Interaction Centre (UCLIC) and a deputy head of the Computer Science department at UCL. Her research interests are in the areas of ubiquitous computing, interaction design and human-computer interaction. A central theme is how to design interactive technologies that can enhance life by augmenting and extending every day, learning and work activities. This involves informing, building and evaluating novel user experiences through creating and assembling a diversity of pervasive technologies. Central to her work is a critical stance towards how visions, theories and frameworks shape the fields of HCI, cognitive science and Ubicomp. She has been instrumental in promulgating new theories (e.g., external cognition), alternative methodologies (e.g., in the wild studies) and far-reaching research agendas (e.g., "Being Human: HCI in 2020" manifesto). She has also published a monograph (2012) called "HCI Theory: Classical, Modern and Contemporary."

List of accepted papers

1. ***XD-MVC: Support for Cross-Device Development***
   Maria Husmann and Moira Norrie
   ETH Zurich, Switserland.
2. ***Reporting Experiences on Group Activities in Cross-Device Settings***
   Johannes Zagermann [a], Ulrike Pfeil [a], Mario Schreiner [a], Roman Rädle [b], Hans-Christian Jetter [c] and Harald Reiterer [a]
   [a] University of Konstanz, Germany
   [b] Microsoft Research, UK
   [c] University of Applied Sciences, Austria
3. ***Deployable Cross-Device Experiences: Proposing Additional Web Standards***
   Mario Schreiner [a] , Roman Rädle [b] , Kenton O'Hara [b] and Harald Reiterer [a]
   [a] University of Konstanz, Germany
   [b] Microsoft Research, UK
4. ***A Framework Towards Challenges and Issues of Multi-Surface Environments***
   Chi Tai Dang and Elisabeth Andre
   University of Augsburg, Germany
5. ***Towards Enhancing Data Exploration with Multiple Mobile Devices***
   Sven Mayer [a], Przemysław Kucharski [b] , Lars Lischke [a] and Paweł Woźniak [c]
   [a] University of Stuttgart, Germany
   [b] Lodz University of Technology, Poland
   [c] Chalmers University of Technology, Sweden
6. ***Using Semantic and Responsive Web Design Technologies for Cross-Device Interactions in Industrial Applications***
   Jens Ziegler, Sebastian Heinze, Markus Graube, Stephan Hensel and Leon Urbas
   Technische Universität Dresden, Germany
7. ***Device Boundaries: Posture in Interaction Ecologies***
   Daniele Savasta
   Yasar University, Turkey
8. ***Mixed Reality Environments as Ecologies for Cross Device Interaction***
   Jens Müller and Harald Reiterer
   University of Konstanz, Germany
9. ***Toward Utopian Multi-Surface Interactive Systems***
   Cornelius Toole
   Tougaloo College, USA

10. ***A Reality Checklist for Multi-Device Systems in the Wild?***

    Henrik Korsgaard [a] , Steven Houben [b] and Clemens Klokmose [a]

    [a] Aarhus University, Denmark.

    [b] University College London, UK

11. ***Ad hoc adaptability in video-calling***

    Sean Rintel, Kenton O'Hara, Behnaz Rostami Yeganeh and Roman Rädle

    Microsoft Research, UK

12. ***Challenges for Managing Multi-Device Notifications***

    Abhinav Mehrotra [ab] , Jo Vermeulen [a] , Robert Hendley [a] and Mirco Musolesi [b]

    [a] University of Birmingham, UK

    [b] University College London, UK

XD-MVC: Support for Cross-Device Development

Maria Husmann
Department of Computer Science
ETH Zurich
husmann@inf.ethz.ch

Moira C. Norrie
Department of Computer Science
ETH Zurich
norrie@inf.ethz.ch

Presented at the Cross-Surface '15 workshop, in conjunction with ACM ITS'15.
November 15, Funchal, Madeira, Portugal.

Abstract
Cross-device applications are still rarely encountered in the wild. In order to facilitate application development, we introduce XD-MVC, an open-source cross-device framework based on web technologies. XD-MVC is lightweight and integrates with MVC frameworks. Most modern browsers are supported and client devices require no installation. XD-MVC provides connection management, data-synchronisation, and support for UI distribution based on device types and roles. Developer needs are addressed with defaults for common tasks and powerful customisation mechanisms. If available, peer-to-peer communication is used to achieve fast data-synchronisation between devices, but a hybrid architecture offers a client-server fallback to ensure that a wide range of devices can be supported. As a first evaluation, several applications have been built on top of XD-MVC, using both its JavaScript API and the provided integration with the Polymer MVC framework.

Author Keywords
cross-device; framework; web; peer-to-peer.

ACM Classification Keywords
H.5.2 [Information interfaces and presentation (e.g., HCI)]: User Interfaces - Input devices and strategies

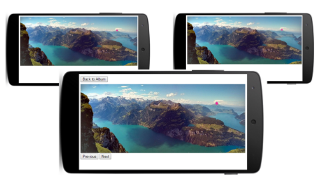

Figure 1: Example application. A gallery application distributed over three smartphones.

Figure 2: Example application. The same gallery application distributed over a smartphone and a TV.

Motivation

Despite a lot of interest in academia, there are still relatively few cross-device applications available to the public. In part, this could be caused by the fact that developing them is challenging due to the lack of available tools as many of the research projects that do offer support are not accessible to developers (e.g. Panelrama [8], Conductor[3]). Two notable exceptions are Polychrome [1] and Connichiwa [7]. Our goal is to support developers in the development process from prototyping through implementation to testing and debugging. We have addressed the first part in MultiMasher [4], a tool for cross-device mashups which could be used for prototyping, and XDStudio [6], a GUI builder for cross-device applications. As a next step, we have created XD-MVC, a framework for implementing web-based cross-device applications. XD-MVC is an open-source project and has been published on Github[1].

System Design

Unlike Connichiwa [7], XD-MVC requires *no installation* on client devices and runs in most modern browsers, however it does depend on network infrastructure. While similar in architecture to Polychrome [1], XD-MVC is not limited to visualisation applications. Besides this *versatility* in terms of application domain, we wanted to give the developers the choice of which specific technologies and UI frameworks to use. Working with familiar technologies could lower the threshold to developing cross-device applications. Thus, we paid attention to making XD-MVC *lightweight*. XD-MVC can be used in combination with MVC frameworks[2] and we provide an integration with the Polymer[3]. We chose Polymer for our proof-of-concept implementation because it is a future-facing framework that supports interesting concepts such as two-way databinding, encapsulation and specification of custom tags. XD-MVC consists of a lower level pure JavaScript API and a higher level API based on Polymer. As argued by Myers et al. [5], UI frameworks should aim for a low threshold and a high ceiling. In order to achieve a low threshold, we provide *defaults* for common use cases in the Polymer API. On the other hand, to keep the ceiling high, control options and means for *customisation* are offered to the developers in the JavaScript API. A developer opting for a different MVC framework, or none at all, can build directly on the JavaScript API (Fig. 3). In order to achieve *fast communication* between devices, we have experimented with a peer-to-peer architecture. Performance tests revealed large performance gains when devices were co-located and the server remote when compared to a client-server architecture. However, not all modern browsers support the necessary WebRTC[4] protocol yet. We thus built XD-MVC with a hybrid architecture where devices preferably communicate peer-to-peer, but use client-server communication as a fallback.

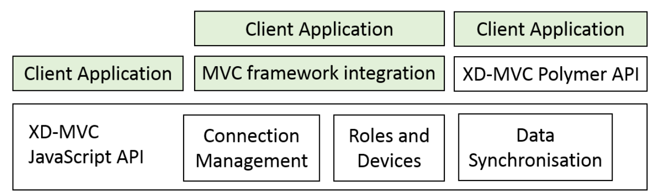

Figure 3: XD-MVC framework structure

XD-MVC Concepts

The framework conceptually consists of three main parts, namely *connection management*, *data synchronisation* and *UI distribution*.

[1] https://github.com/mhusm/XD-MVC
[2] See http://todomvc.com/ for a comprehensive list of frameworks.
[3] https://www.polymer-project.org
[4] http://www.webrtc.org/

```
<xdmvc-connection
  server="xdmvc.
    ethz.ch"
  peerport="9000"
  socketport="3000"
  ajaxport="9001"
  reconnect
  architecture="
      hybrid">
</xdmvc-connection>
```

Listing 1: Declaratively adding connection management using the Polymer API. All attributes have default values and need not be specified.

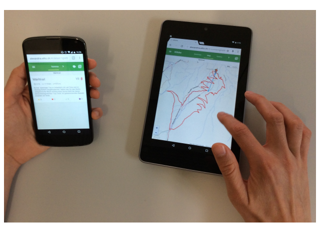

Figure 4: Example application. A bike trip planning application that displays information on a selected route across multiple devices. The map is displayed on the largest device.

Connection Management
Each device is identified by a device ID either generated by the system or set by the developer. To group two devices together, one of them specifies the ID of the other in a connection request. If either device is already connected to a group of other devices, the groups will be merged.

The framework allows the device ID and the connections to be made persistent. Upon loading the web application, the device will attempt to reconnect to all previously connected devices, thus reducing the often tedious pairing process. This behaviour may not always be desired and can be changed or disabled, but has been invaluable during the development process of the showcase applications.

Data Synchronisation
Conceptually, there is a single model that is shared among all connected devices. Depending on the application scenario, it may be desirable to not only synchronise the domain model, but also the state of the view, for example, the content of an input field. In XD-MVC, this requires that a model of the view is defined, referred to as viewmodel in a variation of the MVC pattern. The data synchronisation module of XD-MVC is agnostic to the kind of data that is shared. The developer can specify any object to be shared among the devices—it makes no difference whether the object represents a model or a viewmodel.

XD-MVC's API provides both defaults and options for customisability. While, per default, the framework observes the objects for changes, the developer can also explicitly push updates. This can be useful when the objects are not fully under the control of the developers. In XD-MVC, synchronisation is based on a publish-subscribe mechanism. Each device publishes changes of its objects to devices that subscribed. By default, the changes will be integrated into the local version of the object, however, in the changed event-handler, the developer can specify other actions. The event-handler not only reports the changes to the object, but also which devices sent the changes. In addition to sending changes to connected devices, they can also be sent to the server where they can be made persistent.

UI Distribution
UI distribution mostly concerns the views, which in turn are highly dependent on the specific MVC frameworks used. XD-MVC provides a query mechanism for devices as a basis for the distribution, so that distribution can be based on the type of the local device as well as the devices which are connected to it. In addition, XD-MVC supports roles. Both [2] and XDStudio [6] introduce the notion of user roles which relates to the tasks carried out by users. In XD-MVC, we interpret the terms more loosely and simple consider it additional information associated with a device. While it can relate to users, a device can assume multiple roles simultaneously and roles can be created and changed dynamically at run-time. When a role is added to a device, this information is propagated to all connected devices so they can adapt accordingly. We implemented a publish-subscribe API for role changes. In addition to this event-based mechanism, the current state of roles in the group can also be queried at any time. The query API can be used to check the roles of the local device (List. 2) as well as the number of roles (or devices) of a given type to which it is connected.

```
<template is="dom-if" if="{{roles.
    isselected.owner}}">
  <gallery-element>
  </gallery-element>
</template>
```

Listing 2: Specifying the UI distribution by querying roles in a Polymer application. The Gallery elment will only be displayed if the local device has the owner role.

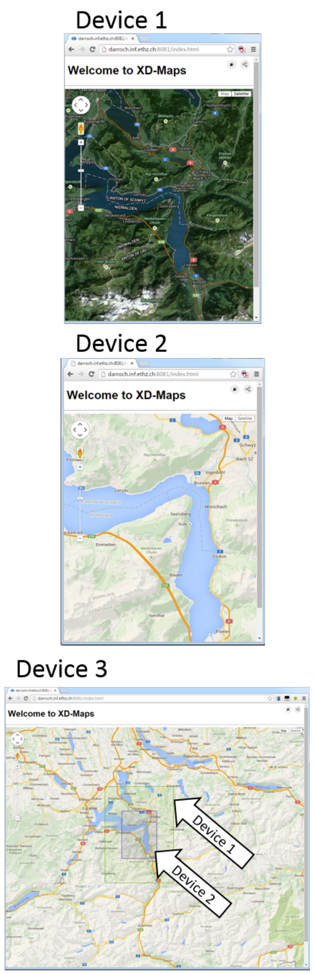

Figure 5: Example application. A maps application. Device 1 and 2 synchronise the map centre while device 3 shows their viewports.

Applications

Several applications have been built based on XD-MVC. XDMaps is a maps application that allows users to explore a location in multiple views (e.g. street map and satellite). It also includes an overview mode that visualises the viewports of the collaborating devices on a map. XDMaps builds on the JavaScript API. XDGallery is an application for easily showing pictures to friends using whatever devices are at hand, such as smartphones but also large screens such as smart TVs. A student used AngularJS[5] on top of XD-MVC to build a slideshow application. A group of four students used XD-MVC to build an application for planning bike trips. The application distributes itself over the available devices taking into account the screen size of the devices. For example, map information will be shown on a larger screen while a short description of a bike route will be displayed on smaller screens. XDBike and XDGallery haven been built using the Polymer API.

Conclusion

With XD-MVC, we lay the foundation of our goal to make cross-device application development easier. However, during the development of the applications, we noticed that further support is needed. We are currently building a set of tools for testing and debugging cross-device applications. Cross-device applications are typically built to support a large number of device configurations which is challenging to test as not all devices may even be available to the developer and switching between configurations could be cumbersome. In parallel, we are working on a tool for analysing usage data of deployed cross-device applications. The goal is to gain insights into how users actually use cross-device applications in order to optimise and improve them.

[5] https://angularjs.org/

REFERENCES

1. Sriram Karthik Badam and Niklas Elmqvist. 2014. PolyChrome: A Cross-Device Framework for Collaborative Web Visualization. In *Proc. ITS*. ACM, 109–118. DOI: http://dx.doi.org/10.1145/2669485.2669518
2. Luca Frosini and Fabio Paternò. 2014. User Interface Distribution in Multi-Device and Multi-User Environments with Dynamically Migrating Engines. In *Proc. EICS*. ACM, 55–64.
3. Peter Hamilton and Daniel J. Wigdor. 2014. Conductor: Enabling and Understanding Cross-Device Interaction. In *Proc. CHI*. ACM, 2773–2782.
4. Maria Husmann, Michael Nebeling, Stefano Pongelli, and Moira C. Norrie. 2014. MultiMasher: Providing Architectural Support and Visual Tools for Multi-Device Mashups. In *Proc. WISE*. Springer, 199–214.
5. Brad A. Myers, Scott E. Hudson, and Randy F. Pausch. 2000. Past, present, and future of user interface software tools. *ACM TOCHI* 7, 1 (2000), 3–28. DOI: http://dx.doi.org/10.1145/344949.344959
6. Michael Nebeling, Theano Mintsi, Maria Husmann, and Moira C. Norrie. 2014. Interactive Development of Cross-Device User Interfaces. In *Proc. CHI*. ACM, 2793–2802.
7. Mario Schreiner, Roman Rädle, Hans-Christian Jetter, and Harald Reiterer. 2015. Connichiwa: A Framework for Cross-Device Web Applications. In *Proc. CHI EA*. ACM, 2163–2168. DOI: http://dx.doi.org/10.1145/2702613.2732909
8. Jishuo Yang and Daniel Wigdor. 2014. Panelrama: Enabling Easy Specification of Cross-Device Web Applications. In *Proc. CHI*. ACM, 2783–2792.

Reporting Experiences on Group Activities in Cross-Device Settings

Johannes Zagermann
HCI Group
University of Konstanz
johannes.zagermann@uni.kn

Ulrike Pfeil
HCI Group
University of Konstanz
ulrike.pfeil@uni.kn

Mario Schreiner
HCI Group
University of Konstanz
mario.schreiner@uni.kn

Roman Rädle
HCI Group
University of Konstanz
roman.raedle@uni.kn

Hans-Christian Jetter
University of Applied Sciences
Upper Austria
hans-christian.jetter@fh-hagenberg.at

Harald Reiterer
HCI Group
University of Konstanz
harald.reiterer@uni.kn

Presented at the Cross-Surface '15 workshop, in conjunction with ACM ITS'15. November 15, Funchal, Madeira, Portugal.

Abstract
Even though mobile devices are ubiquitous and users often own several of them, using them in concert to achieve a common goal is not well supported and remains a challenge for HCI. In this paper, we report on our observations of cross-device usage within groups when they engaged in a dyadic collaborative sensemaking task. Based on our findings, we discuss limitations of a state-of-the-art cross-device setting and present a set of design recommendations. We then propose an alternative design that aims for greater flexibility when using mobile devices to enable a free configuration of workspaces depending on users' current activity.

Author Keywords
Evaluation; cross-device interaction; group work

ACM Classification Keywords
H.5.2. User Interfaces: Evaluation/methodology

Introduction
Collaborative group tasks such as searching, organizing, or problem solving in general are usually facilitated by shared group spaces. In these spaces, tables are often used because of their physical affordances that are especially appropriate for group work activities [10]. Therefore conferences like CHI, CSCW, or ITS have closely investigated the impact of

Fig. 1: Tabletop size 10.6''

Fig. 2: Tabletop size 27''

Fig. 3: Tabletop size 55''

(interactive) tabletops on group work activities and related phenomena [4,8,11]. In addition, the applicability of activity spaces for multiple devices usage [3] and multi-screen environments for active reading [1] showed that cross-device settings feasibly support knowledge work. Based on such prior work, this paper aims at informing the design of future cross-device systems by observing how users make use of mobile technologies around interactive tabletops. To achieve this, we used a refined version of our TwisterSearch [8] system in a user study to observe group work activities in such a cross-device setting. TwisterSearch consists of two tablets for personal work and one shared tabletop for group work. The personal devices enable individual search and reading activities, while the shared group workspace enables structuring and sharing information shown as snippets during around-the-table collaboration. The system allows for a digital transfer of documents between all three devices.

Based on our results, we propose an alternative design for future cross-device settings that allows for a flexible configuration of multiple devices for collaborative sensemaking activities. This conceptual solution aims to move beyond traditional tabletop settings to support more complex device ecologies [3,5].

User Study

We conducted an exploratory study in which three different tabletop sizes (see Fig. 1-3) were used as shared group spaces in a between-subjects design. In total, we investigated five dyads in each condition resulting in 15 dyads and 30 participants respectively. Each dyad was asked to work on the VAST 2006 Stegosaurus tasks [2]. This involved searching for relevant information in a data set of 238 documents, images and data sheets. The aim was to find relations between these documents and come up with a solution to a hidden plot. Participants had 90 minutes in order to work on the task and find relevant information.

The focus of the evaluation laid on the investigation of the groups' activities as well as their collaboration behaviors. Based on video observations, our results report participants' experiences with the system. Limitations are identified and feed into the development of our alternative design that aims for a flexible configuration of personal and shared work spaces depending on the activity on hand.

Results

In the following, we describe the main activities as observed in our study and discuss them in relation to existing work. Based on our findings, we formulate five recommendations for the design of future systems. They are shown in the side bar on the next page.

Configuration of personal and shared spaces

As intended, participants used the tablets in order to search and read the found documents. However, participants only shared a document on the table when they were sure that it was relevant. Often, they displayed the content of a document to their partner in order to ask for agreement to transfer it to the shared space. Even when adding documents to the table, participants tended to first keep the documents on their side of the table without immediately combining the results with their partner. It seemed that this was done to first get an overview over the found documents in private. Thus, we conclude that future systems should allow for a free configuration of private and shared work spaces (**R2**), which resonates with [11]'s findings.

Recommendations

R1: *Offer three views (search, read, collect) on all devices in order to break the dependency of functionality and device.*

R2: *Allow for a fluid transition of personal and shared space to enable users to decide when and how they want to publish their work.*

R3: *Allow for easy and seamless transfer of documents and search results between all incorporated devices.*

R4: *Allow for handover and sharing of devices.*

R5: *Allow for a flexible configuration of the shared space size to support a fluid transition of activities that need a large display size as well as division of labor.*

Document transfer
When working together, participants often utilized the possibility to transfer documents between devices. Users acknowledged this functionality. It was found to be very helpful and considered essential, and should thus be included in future systems (**R3**).

Furthermore, participants often showed documents on their tablet to their partner to share interesting findings, treating the tablet like a paper-document. However, participants were reluctant to give away their personal tablet as it was introduced as their personal device and they only had one of this kind. Thus, future settings should provide users with multiple tablets to support active reading [1] and to encourage them to handover mobile devices (**R4**).

Device functionalities
The shared tabletop in TwisterSearch allows to visualize documents as a snippet to cluster them. This activity was found to act as a common ground for discussions about possible relations between the documents. However, participants first wanted to collect and sort their findings on their own before sharing their results with their partner (see **R2**). They did not have a possibility to collect or keep documents and search results on their personal device for further usage [3] as the collection of documents was bound to be done on the shared table. Similarly, when working on the table, participants often lost track of the content of the snippets and wished to view the content of the document. Thus, they sent the document from the table to their personal tablet to read it again. This was found to be a quirky work-around as it was not possible to seamlessly switch between the snippet-view and the detail-view on the shared work space. Thus, we conclude that functionalities should be available on all personal and shared devices (**R1**). This would also allow for a better comparison of single documents when working on them as a group.

Configuration of the workspace
Finally, towards the end of the session, participants made much more use of the table for clustering documents to externalize relevant relations between them. In addition, the shared table was used to gain an overview of the task progress. For these activities, a large display size was considered to be important. In addition, participants utilized the table to divide further tasks between them. This is similar to [6]'s activity of "divide & conquer" and [3]'s description of fragmenting resources across different devices. We thus propose to allow for an easy configuration of the workspace depending on the task at hand (**R5**).

In the following, we address our identified recommendations in a concept based on a flexible setup of interactive tablets that can be (de)coupled ad hoc depending on the current activities. We believe that especially in collaboration tasks where users frequently switch between different activities, such a flexible setup allows for a better adjustment of the system to the task at hand and current user needs.

Concept
One of the main shortcomings of our system was found to be the forced distinction between activities that could be performed only on the personal tablet or only on the table. In order to break the dependency of functionality and devices (**R1**) and allow for a fluid transition between personal and shared work spaces (**R2**), we propose an interactive setting consisting on a higher

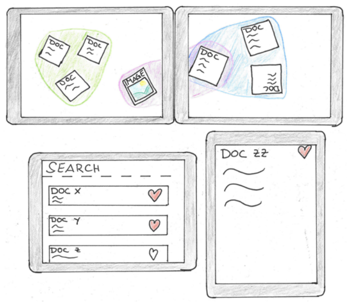

Fig. 4: Mixed Shared Workspace

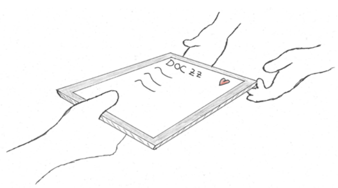

Fig. 5: Physical Handover

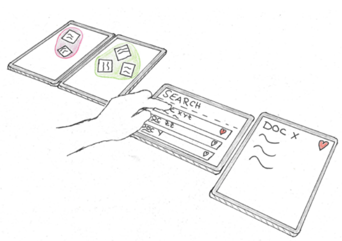

Fig. 6: Mixed Personal Workspace

number of tablets that can be freely configured spatially as well as content-related. This accounts for users' extensive device ecologies of up to ten interactive devices [5] allowing for their utilization as a single device as well as offering the possibility to connect several tablets and treat them as a larger interactive surface to perform shared activities (**R5**) (see Fig. 4). This also loosens the borders between single-user and group-based activities (**R2**). Users are able to perform actions in the same fashion in several phases and on multiple devices of their group work session.

The concept still utilizes one view to search and one to read documents. In addition to this, we add a third view to allow organizing, sorting and clustering of documents on single tablets in the same manner as in the previous tabletop system (**R1**). Thus, the snippet-view does not only act as a container for *shared* documents but as an alternative interactive visualization of search results. Users can find connections or dismiss irrelevant information more easily and with a higher visibility. As the basis for each of the three views are the found documents, changes to them in one view will instantly take effect in other views (**R3**). Seamless transitions between all views are provided, e.g. documents can be read by tapping on a search result or snippet.

In addition, users are able to use multiple tablets. The higher number of tablets on-hand might lead to a change of perspective: The former one-to-one distribution of tablets lead to a perception of personal devices – providing multiple tablets per persons might change this to temporarily personal or activity-based device perceptions [3], encouraging users to use multiple tablets at the same time, each both, separately and combined (**R5**) (see Fig. 4 & 6). This allows users to read various documents whilst engaging in visual structuring activities at the same time [1]. In addition, users can combine multiple tablets to a larger interactive surface (**R5**) using technologies like [7] or [9]. The additional space can be used to organize or sort documents in a private session, without having to interrupt group partners or overcome personal barriers (**R2**).

The flexible work space can be rearranged as shared space and tablets can be added or removed at any time allowing multiple users to work together by adding their tablets to build a larger interactive shared work space (**R2** & **R5**). This flexibility might help to support group activities like the division of labor as each user can take a device to further investigate information. Interesting documents can be transferred to different devices (**R3**) or handed over physically to each group member (**R4**) (see Fig. 5). Thereby, the perception of devices might change, as they no longer appear as tools to search and read, but as physical manifestations of digital documents.

Conclusion

Overall, our concepts allows for a flexible and tailored use of space, smoothly adapting to changing requirements, the experience of which goes beyond the usage of a single large screen [1]. As [3] suggest, we aim for a light-weight setting that does only afford minimal configuration work to allow users to dynamically relate interconnected devices to best support the activity at hand.

References

[1] Chen, N., Guimbretière, F., and Sellen, A.Graduate student use of a multi-slate reading system. *In Proc. of CHI ’13*, ACM Press (2013), 1799-1808.

[2] Grinstein, G., O’Connell, T., Laskowski, S., Plaisant, C., Scholtz, J., and Whiting, M.Vast 2006 Contest - A Tale of Alderwood. *In Proc. of the IEEE Symposium on Visual Analytics Science and Technology*, February (2006), 215–216.

[3] Houben, S., Tell, P., and Bardram, J.E.ActivitySpace: Managing Device Ecologies in an Activity-Centric Configuration Space. *In Proc.of IST'14*, ACM (2014), 119–128.

[4] Isenberg, P., Fisher, D., Morris, M.R., Inkpen, K., and Czerwinski, M.An Exploratory Study of Co-located Collaborative Visual Analytics Around a Tabletop Display. *2010 IEEE Symposium on Visual Analytics Science and Technology*, (2010), 179–186.

[5] Jung, H., Stolterman, E., Ryan, W., Thompson, T., and Siegel, M.Toward a Framework for Ecologies of Artifacts: How Are Digital Artifacts Interconnected Within a Personal Life? *In Proc. of NordiCHI'08: Building Bridges*, ACM (2008), 201–210.

[6] Morris, M.R.A Survey of Collaborative Web Search Practices. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2008), 1657–1660.

[7] Rädle, R., Jetter, H.-C., Marquardt, N., Reiterer, H., and Rogers, Y.HuddleLamp: Spatially-Aware Mobile Displays for Ad-hoc Around-the-Table Collaboration. *In Proc. of ITS ’14*, ACM (2014), 45-54.

[8] Rädle, R., Jetter, H.-C., and Reiterer, H.TwisterSearch: A Distributed User Interfaces for Collaborative Web Search. *Distributed User Interfaces: Collaboration and Usability*, (2013), 1–17.

[9] Schreiner, M., Rädle, R., Jetter, H.-C., and Reiterer, H.Connichiwa - A Framework for Local Cross-Device Web Applications. *In Proc. of CHI EA’15*, ACM (2015), 2163-2168.

[10] Scott, S.D., Sheelagh, M., Carpendale, T., and Inkpen, K.M.Territoriality in collaborative tabletop workspaces. *Proceedings of the 2004 ACM conference on Computer supported cooperative work - CSCW ’04*, ACM Press (2004), 294-303.

[11] Tang, A., Tory, M., Po, B., Neumann, P., and Carpendale, S.Collaborative coupling over tabletop displays. *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI ’06*, ACM Press (2006), 1181-1190.

Deployable Cross-Device Experiences: Proposing Additional Web Standards

Mario Schreiner
University of Konstanz
78457, Konstanz, Germany
mario.schreiner@uni.kn

Harald Reiterer
University of Konstanz
78457, Konstanz, Germany
harald.reiterer@uni.kn

Roman Rädle
Microsoft Research
Cambridge, UK
t-rorad@microsoft.com

Kenton O'Hara
Microsoft Research
Cambridge, UK
keohar@microsoft.com

Presented at the Cross-Surface '15 workshop, in conjunction with ACM ITS'15.
November 15, Funchal, Madeira, Portugal.

Abstract

Cross-device interaction is rarely observed in everyday life and outside of research facilities. In this position paper we explore potential reasons for this shortcoming and discuss why the web is a promising enabling technology for cross-device interactions. We propose a concept for new, cross-device centric web standards that would allow to develop, deploy, and use cross-device applications in everyday life.

Author Keywords

cross-device; api; ad hoc; web; standards; position paper

ACM Classification Keywords

D.3.0 [Programming Languages]: General — Standards

Problem Statement

Interaction across multiple devices is an everyday activity and support for moving information and working between devices is wanted by their owners [6]. Yet, not many real-world applications exist that satisfy this need. Features such as Apple's Continuity allow users to transition a task from one device to another, but such technologies focus on sequential use of devices rather than using them in parallel on the same task and interacting across devices [6]. Even more, a technology standard to find, connect, and interact across nearby devices does not exist, which can be discouraging for developers when designing and implementing cross-device applications. Based on our experiences in developing cross-device technologies [9, 10], we defined four key requirements for future cross-device technologies, which we believe will be essential to enable and deploy cross-device interactions in everyday life.

Low threshold to join: While device augmentation is often utilised to make devices aware of each other [4, 5, 8], the need for special hardware hinders ad hoc cross-device interactions. Software-side setups [3, 7] face similar problems and increase the threshold for joining a cross-device application. We propose the use of off-the-shelf consumer devices to enable everyone to participate in cross-device interaction. Also we believe cross-device technology with minimal to no configuration effort can leap cross-device interactions beyond the scope of research projects.

Independent of location: Augmenting the environment [1, 9, 11] enables accurate tracking of device locations, but confines interaction to a small space (e.g., a table or a room). Such augmentation seems unfeasible for real-world deployment and an approach that enables interaction anywhere and at any time should be sought. Understanding appropriate groupings of devices in cross device interactions is also not something that is bound up entirely in physical characterisations of proximity. How these physical properties map onto social characterisations of proximity is something that also needs consideration in the definition of cross-device interaction. One might be physically close to another person (e.g. on a bus) but not socially proximate. Augmenting physical proximity indicators with social indicators (e.g. presence in contact lists may better support the establishment of appropriate device-to-device connections).

Fluidity in device configurations: Users must rely on whatever devices are currently available in their surrounding. They must be able to join an activity and add or remove devices at any time during a task. This requires cross-device technologies to cope with and support fluid changes of device configurations. There also needs to be good intelligibility of device availability to facilitate socially appropriate configuration of device assemblies.

Small development effort: Increased development effort in porting an application to all major operating platforms can lead to developers not adapting a new technology, which in turn will make it difficult to establish cross-device applications for everyday use. Developers must be able to build cross-device experiences for users and adapt to the multitude of platforms without heavy migration efforts. APIs like Microsoft UWP, Mono, PhoneGap, or Cordova are already targeting this issue and translate a shared code base into devices' respective native language. However, this still requires deployment of native apps on the devices.

Based on our experience [9, 10], we believe that web technologies can fulfil these requirements in the future. In this position paper, we want to discuss the benefits and shortcomings of state-of-the-art web technologies in regard to cross-device interaction. We further present a concept for how current web standards could be extended to establish them for real-world cross-device application development.

Cross-Device Web APIs

The modern web stands out with its massive availability and large standardisation across consumer devices [2]. At first glance, this seems to make the web an ideal choice for cross-device development with web applications running on mobile and desktop systems, TVs, gaming consoles, digital cameras, watches, across different hardware, screen sizes, and input modalities. Most modern device have a built-in web browser, making software installation obsolete. Development effort is minimised due to standardised markup, styling, and scripting languages (HTML, CSS, JavaScript). User interface presentation and interaction consistent across different devices, making web development an appealing underlying technology for many companies and researchers. Notably, many applications formerly developed for particular platforms are transitioning to the web (e.g. Microsoft Office, Spotify, Mendeley, Apple Mail).

```
window.ondevicenearby =
function(device)
{
  if
  (
    device.clientWidth >= 1920
    &&
    device.clientHeight >= 1080
    &&
    device.memory.total >= 2048
    &&
    device.supports("camera")
  )
  {
    device.connect();
  }
};

window.ondeviceleave =
function(device) {
   //do cleanup here
};
```

Listing 1: Example code of enter and exit events of nearby devices. `.connect()` will request a peer-to-peer connection.

```
window.ondeviceconnect =
function(device)
{
  device
   .getElementById("content")
   .innerHTML = "Hello!";
};
```

Listing 2: Device objects allow to perform actions on a device such as remotely updating content.

Despite these advantages of web technologies, combining devices remains a technical challenge. Web communication is based on central, remote servers, and information about devices in close physical proximity is not available, which would be necessary for nearby devices to join ad hoc in the formation of co-located communities of devices. We therefore propose a JavaScript API to access local communication technologies such as Bluetooth, NFC, and Wi-Fi Direct. This would allow web applications to detect nearby devices and establish a local peer-to-peer communication channel (see Listing 1). By using already available technologies, augmentation of devices is not necessary and the approach is independent of the devices' current location.

With devices connected in such a manner, the available device information should be extended to allow applications to adapt to different device configurations. As of now, JavaScript can access only limited information about devices, such as resolution or operating system. Other information such as physical screen size, input capabilities, available memory, or attached hardware are not discoverable through an API. This, however, will be necessary to determine the role of devices in a task [12]. For example, a presentation software needs to find nearby large screens for displaying the presentation, and handheld devices for displaying controls. Attendees could get annotation abilities on stylus-enabled personal devices. Therefore, browsers should make device information accessible to web applications. Of course, owners of devices should be able to opt out from broadcasting such information.

In addition, web languages should be extended for multi-device support. We propose enter and exit events for devices in close, physical and social, proximity (see Listing 1). Complexity should be reduced by encapsulating devices in JavaScript *device objects*. A device object represents a `window` object of a close device and is passed as a parameter to the enter and exit events. This, for instance, would allow developers to gather information about remote devices (see Listing 1) or to execute commands on remote objects (see Listing 2). Internally, the device object would handle serialising, transmitting, parsing and executing commands, completely transparent to the developer. Due to the asynchronous nature of network communication, such function calls require callbacks or JavaScript *Promises*. For instance, `device.getElementById()` would not instantly return a DOM element but rather a Promise that is resolved when the element was received on the caller's end. Multi-device concepts can also be applied to CSS through new media queries (see Listing 3), allowing developers to adjust styles based on the number or features of remote devices.

In a prototype framework [10], we showed the feasibility of these concepts. During a small scale user study, we handed the framework to developers, including data synchronisation, reactive templating, and position detection via cross-device gestures. Developers adopted quickly to the framework and responded positively to the new cross-device interaction possibilities.

Limitations

In order for the given concept to be feasible for real-world deployment, this standard must be proposed to the W3C, accepted, and implemented by all major browser vendors.

```
@media
  (device-role: main)
  and
  (min-number-of-devices: 2)
{
  #main-content
  {
    display: block;
  }
}

@media
  (device-role: sidebar)
  and
  (min-number-of-devices: 2)
{
  #sidebar {
    display: block;
  }
}

@media
  (any-device-role: video)
{
  #button-airplay {
    visibility: visible;
  }
}
```

Listing 3: Example of multi-device CSS support. Features such as the total number of devices or the device role - set via JavaScript - are available to adjust styles.

Still, there are some limitations to the proposed concept:

Communication over the Internet: A stable Internet connection to retrieve assets from and store data on a remote server is still required. Technologies such as the HTML5 ServiceWorker API or the HTML5 IndexedDB API could be powerful and decentralised alternatives to remote servers.

Performance of local communication: Local communication, e.g. over Bluetooth tends to be slower than Wi-Fi or GSM. Eventually, such communication might not satisfy the performance needs of applications. A solution could be communication over a shared Wi-Fi network or offering developers the ability to take control over the communication using an on-device or remote WebSocket server.

Security: Security is of large concern on the web, and some features proposed must be implemented with care to ensure the protection of private data and user data.

Conclusion

We believe web technologies could provide a new standard for cross-device development. Their availability, standardisation, and support by the majority of devices and operating systems is appealing to application developers and users. In order to enable interaction with nearby devices in everyday life, methods for detection of and communication with devices in their vicinity are needed. The web languages must be extended for APIs that allow developers to easily interact with other devices, e.g. exchanging data and remotely updating content. If the web standard is extended in such a way, we believe cross-device experiences will find their way out of the labs into our everyday life.

References

[1] T. Ballendat, N. Marquardt, and S. Greenberg. 2010. Proxemic Interaction: Designing for a Proximity and Orientation-aware Environment *(ITS '10)*. ACM.

[2] caniuse.com. 2015. Can I use - Web Feature Availability Overview. http://caniuse.com//#feature_sort=usr_score. (2015). Online - Accessed: 19th August 2015.

[3] D. Dearman, R. Guy, and K. Truong. 2012. Determining the Orientation of Proximate Mobile Devices Using Their Back Facing Camera *(CHI '12)*. ACM.

[4] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen. 2001. Smart-Its Friends: A Technique for Users to Easily Establish Connections Between Smart Artefacts *(UbiComp '01)*. Springer.

[5] D.-Y. Huang, C.-P. Lin, Y.-P. Hung, T.-W. Chang, N.-H. Yu, M.-L. Tsai, and M. Y. Chen. 2012. MagMobile: Enhancing Social Interactions with Rapid View-stitching Games of Mobile Devices *(MUM '12)*. ACM.

[6] T. Jokela, J. Ojala, and T. Olsson. 2015. A Diary Study on Combining Multiple Information Devices in Everyday Activities and Tasks *(CHI '15)*. ACM.

[7] A. Lucero, J. Holopainen, and T. Jokela. 2011. Pass-them-around: Collaborative Use of Mobile Phones for Photo Sharing *(CHI '11)*. ACM.

[8] D. Merrill, J. Kalanithi, and P. Maes. 2007. Siftables: Towards Sensor Network User Interfaces *(TEI '07)*.

[9] R. Rädle, H.-C. Jetter, N. Marquardt, H. Reiterer, and Y. Rogers. 2014. HuddleLamp: Spatially-Aware Mobile Displays for Ad-hoc Around-the-Table Collaboration *(ITS '14)*. ACM.

[10] M. Schreiner, R. Rädle, H.-C. Jetter, and H. Reiterer. 2015. Connichiwa: A Framework for Cross-Device Web Applications *(CHI EA '15)*. ACM.

[11] J. Schwarz, D. Klionsky, C. Harrison, P. Dietz, and A. Wilson. 2012. Phone As a Pixel: Enabling Ad-hoc, Large-scale Displays Using Mobile Devices *(CHI '12)*.

[12] J. Yang and D. Wigdor. 2014. Panelrama: Enabling Easy Specification of Cross-device Web Applications *(CHI '14)*. ACM.

A Framework Towards Challenges and Issues of Multi-Surface Environments.

Chi Tai Dang
University of Augsburg
Human Centered Multimedia
Universitaetsstr. 6a
Augsburg 86159, Germany
dang@hcm-lab.de

Elisabeth André
University of Augsburg
Human Centered Multimedia
Universitaetsstr. 6a
Augsburg 86159, Germany
andre@hcm-lab.de

Figure 1: Interactive portals with mobile devices and tabletop.

Abstract
In this paper, we discuss four of the most challenging technical issues for interactions and applications in environments comprised of nowadays widespread ecologies of touch-enabled mobile and immobile devices. Such issues are important particularly for applications in the wild. We address these issues by means of an appropriate software architecture that is implemented as the reference framework *Environs* in order to foster interactions and applications in multi-surface environments and help bring those into the wild. The framework is available as open-source software thereby we contribute to basic enabling technologies for multi-surface environments.

Author Keywords
Multi-Surface Environment; Portal; Lense; Framework; Cross-device interaction; Tabletop; Tablet; Smartphone; Mobile Device.

ACM Classification Keywords
D.2.6 [Programming Environments]: Interactive environments; H.5.2 [Information Interfaces and Presentation (e.g., HCI)]: Miscellaneous

Introduction
During the last decade, people's interaction habit with daily devices has changed remarkably. While ten years ago peo-

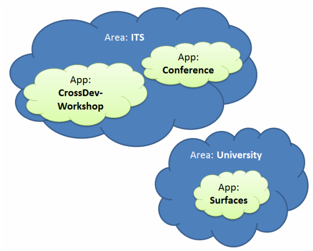

Figure 2: Partitioning concept for MSEs: areas as containers for application environments. Multiple logically separated application areas can exist within the same physical network.

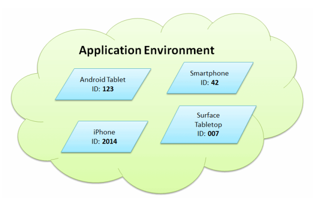

Figure 3: An example application environment comprised of multiple devices which are identified through a numerical ID.

ple were used to interact with mice, touchpads and keyboards only, today, interaction with touch-enabled surfaces has become an inherent part of nowadays interaction repertoire of everyday people. More and more touch enabled mobile (smartwatches, smartphones, phablets, tablets) and immobile (tabletops, wall-mounted display) devices entered the consumer market since around 2007/2008 and the amount of device types as well as proliferation of devices is still increasing rapidly.

Hence, multi-surface environments (MSE) comprised of mobile and immobile interactive surfaces are quite likely to become commonplace in the foreseeable future. The increasing number of recent research articles targeting applications and interactions within MSEs and across multiple surfaces endorse this trend.

With the increase of MSE occurrences, the desire for cross-device interactions and applications will inevitable rise. However, issues and questions originating from differences between lab environments and outside the lab environments need to be addressed by research to enable successful transition of cross-device interactions and applications from labs into the wild.

Outside the lab, Issues and Challenges

Even though HCI research investigated MSEs for many years with great results, studies were conducted in controlled *sterile* lab environments and situations. When going into the wild, things may be different and what worked in the lab may not necessarily work in the wild. Within this paper, we briefly discuss the most challenging issues from a technical point of view and present our research aiming at those issues.

Heterogeneity of platforms is the most challenging issue for interaction designers as well as for application developers. While the device ecology in a lab is manageable, device ecologies in the wild are literally wild. There are different form factors (smartwatch, smartphone, phablet, tablet, etc.), different set and kind of embedded sensors (accelerometer, gyroscope, GPS, heartbeat, etc.), different operating systems (Android, iOS, Windows Phone, etc.), or different programming APIs and platform languages (Java, Objective-C, C#, etc.). Even within the same device platform, the fragmentation of the operating system may result in a multitude of differences.

Network and device management are interrelated and not necessarily optimal in the wild. In terms of network, there may be environments with mobile data only, with wireless network but no internet access, or multiple logically separated/connected subnets with/without internet access. In terms of device management, devices usually take part in an ad-hoc manner and may vanish suddenly which is usually the case for decentralized loosely coupled devices. Both aspects together renders centralized server-based approaches quite difficult for robustness and stability of a system in the wild.

Performance, efficiency, stability, and low latency have direct influence on users' experience. Approaches that work perfectly well for one's lab devices may require further research for other device platforms or to be scalable across device platforms.

Security and safety of data on devices and on transport channels are usually neglected in lab studies. However, those aspects are nowadays mandatory requirements for applications in the wild. Users would behave different or prefer different strategies in studies if they know about the safety of their data.

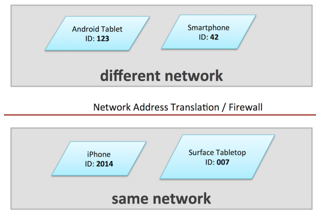

Figure 4: Supported network configurations of Environs: Devices within the same network (broadcast); Devices in different networks (STUN/STUNT mechanism).

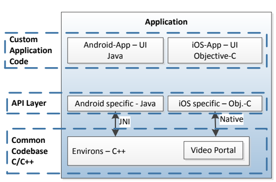

Figure 5: 3-layer architecture of Environs exemplified for the Google Android and Apple iOS platforms.

Development of MSE Applications

In software engineering and development practice, frameworks or toolkits are regularly used for recurrent and/or abstract tasks, e.g. node.js, jQuery, or jQT. Hence, it's conceivable that this will also be the case for development of MSE applications. For example, a multi-platform framework that automatically handle network and connectivity, or device management and communication would greatly unburden developers from implementing the required logic for each supported platform and new application which is prone to errors. Considering the high complexity induced by nowadays heterogeneous device ecologies, such an MSE framework would also be beneficial for research studies and reproduction of research results. Previous research efforts [1, 5, 7, 8] that address MSE framework concepts further confirm this assumption. However, they did not target nowadays device ecologies and the challenges in the wild. Furthermore, lack of availability of the presented frameworks and portability of the concepts (e.g. to a smartwatch) often inhibit its reuse.

We briefly describe some highlights of the multi-platform MSE framework called *Environs* [2] which explicitly addresses the aforementioned issues and challenges. The reference implementation together with several introduction tutorials are publicly available[1] as open-source software in order to foster MSE research as well as development. Moreover, the framework easily enables real-time video-based interactive portals that open up a rich direction for demanding cross-device interactions and applications, e.g. aboard ships [4] or for collaborative tasks [3, 6].

Heterogeneity of platforms is handled by Environs through a 3-layer architecture for applications, see Figure 5. The *application layer* represents the actual application logic and UI that may be designed and auto-generated for multiple platforms by appropriate development tools. The framework itself is implemented in the remaining two layers, whereof the *API layer* provides a thin object oriented API to access the native layer. Under the hood, API objects merely keep object states and function as a proxy to native calls. The *native layer* is realized as a common code base for all platforms and contains the majority (~90%) of the framework logic which is implemented in portable C/C++. Hence, the whole native layer can be compiled for all platforms thereby greatly reduces development time, increases manageability and maintainability, and benefits from less programming errors. Currently, Environs supports the platforms Google Android / Android Wear, Apple iOS/WatchOS/OSX, Microsoft Windows (.NET/Surface 1/PixelSense 2/MultiTaction Cells), and Linux.

Network and device management is completely handled by the native layer which supports devices within the same network as well as devices across different networks. Environs manages so called application environments which can further exist in logically separated areas (e.g. meeting room, office, airport), see Figure 2, thereby enable multiple separated application environments within the same physical network. Each device assigns itself into an application area with a numerical ID, see Figure 3. Devices across different networks (e.g. both devices behind firewalls) require an additional mediator service which helps connect each other by means of STUN/STUNT mechanisms known from peer-to-peer networks. Overall, devices operate in a loosely coupled decentralized network (peer-to-peer), but server-like services are still possible through specialized device nodes.

Performance, efficiency, stability, and low latency is addressed through the native C/C++ implementation and

[1] http://hcm-lab.de/environs

native optimizations. Low latency network communication is based on priority handling of data types that employs low latency transport channels as well as a channel for large chunks of bulk data. For example, touch events are passed to other devices using the shortest code paths and UDP channels. Furthermore, interactive portals make use of hardware encode/decode for low latency.

Security and safety is handled transparently in the native layer by state of the art AES encryption of transport channels. Each device automatically generates its own private/public key and certificate in order to encrypt AES session keys. This is particularly important when connecting multiple locally different MSEs to one application environment over unsecured networks (internet).

Author's Interests and Further Research

Our interest lies in research of natural and intuitive interaction techniques and enabling technologies for novel interactive portal applications in MSEs, which nowadays powerful touch-enabled device ecologies easily enable. Currently, Environs detects the location of devices within an MSE only by means of markers under mobile devices and only if they are placed on supported tabletops. Therefore, we intend to add additional position and spatial tracking of devices and users by means of location nodes within the MSE as proposed in [1] in order to investigate spatial interaction techniques for interactive portals.

REFERENCES

1. Apoorve Chokshi, Teddy Seyed, Francisco Marinho Rodrigues, and Frank Maurer. 2014. ePlan Multi-Surface: A Multi-Surface Environment for Emergency Response Planning Exercises *(ITS '14)*. ACM, New York, NY, USA, 219–228.
2. Chi Tai Dang and Elisabeth André. 2014. A Framework for the Development of Multi-display Environment Applications Supporting Interactive Real-time Portals. In *Proceedings of the 2014 ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '14)*. ACM, New York, NY, USA, 45–54.
3. Chi Tai Dang, Masood Masoodian, and Elisabeth André. 2015. Private Focus Portals to Shared Energy Visualizations, Fostering Smart Energy Applications. In *Adjunct Conference Proceedings of INTERACT 2015*. Human-Computer Interaction - Lecture Notes in Computer Science, Vol. 9299. Springer International Publishing, 657–658.
4. Veronika Domova, Elina Vartiainen, Saad Azhar, and Maria Ralph. 2013. An Interactive Surface Solution to Support Collaborative Work Onboard Ships *(ITS '13)*. ACM, New York, NY, USA, 265–272.
5. Tony Gjerlufsen, Clemens Nylandsted Klokmose, James Eagan, Clément Pillias, and Michel Beaudouin-Lafon. 2011. Shared Substance: Developing Flexible Multi-surface Applications *(CHI '11)*. ACM, New York, NY, USA, 3383–3392.
6. Yucheng Jin, Chi Tai Dang, Christian Prehofer, and Elisabeth André. 2014. A Multi-Display System for Deploying and Controlling Home Automation. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces (ITS '14)*. ACM, New York, NY, USA, 399–402.
7. Bonifaz Kaufmann, Martin Gratzer, and Martin Hitz. 2012. 3MF - A Service-Oriented Mobile Multimodal Interaction Framework. *PPD'12* (2012), 18–21.
8. Florian van de Camp and Rainer Stiefelhagen. 2013. GlueTK: A Framework for Multi-modal, Multi-display Human-machine-interaction *(IUI '13)*. ACM, New York, NY, USA, 329–338.

Towards Enhancing Data Exploration with Multiple Mobile Devices

Sven Mayer
Institute for Visualization and Interactive Systems
University of Stuttgart
Stuttgart, Germany
sven.mayer@vis.uni-stuttgart.de

Przemysław Kucharski
Institute of Applied Computer Science
Lodz University of Technology
Łódź, Poland
pkuchars@iis.p.lodz.pl

Lars Lischke
Institute for Visualization and Interactive Systems
University of Stuttgart
Stuttgart, Germany
lars.lischke@vis.uni-stuttgart.de

Paweł W. Woźniak
t2i Interaction Lab
Chalmers University of Technology
Gothenburg, Sweden
pawelw@chalmers.se

Presented at the Cross-Surface '15 workshop, in conjunction with ACM *ITS'15*.
November 15, Funchal, Madeira, Portugal.

Abstract
In a world of an increasing number of mobile devices in everyday life, people are dealing with large amounts of data every minute. It is an emerging need to create interfaces for multiple devices to support the process of data exploration and understanding. New sensors, enabling mobile devices to be spatially aware, inspire the design of context-aware adaptive interfaces. We indicate a possible direction of further research, where we treat the spatiotemporal relationships between different subsets of a given data set as part of the information communicated by the system. That give us the opportunity to create more effective visualizations to enhance perception. This approach builds on a natural human tendency to organize information spatially, as shown in previous research in cognitive science.

Author Keywords
multidevice; dual device; cross-surface; interactive visualization; multisurface environment; mobile interaction

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

Introduction
Nowadays, people are using mobile devices to explore data on the run, both in work and private life [1]. Data sets

are constantly becoming larger, so supporting effective data exploration emerges as a big challenge for Human-Computer Interaction (HCI). Finding the methods to present large amounts of information in the most effective way is required. The technology available in everyday situations should augment human's perception and sensemaking as effectively as possible. While a number of mobile devices in our homes and offices is still increasing, and those devices are getting more and more powerful, users do not benefit to the extent they could from the fact that those devices can be interconnected.

Certain research was done on possible solutions and applications for multidevice systems. MochaTop [7] and Thaddeus [6] focused on creating new methods of navigation through data sets via cross-device interaction. Rädle et al. [5] created a system where images were displayed on multiple, spatially aware displays. In Conductor [3], Hamilton and Wigdor investigated how multiple spatially-aware devices could enhance user experience and performance. Our work is also inspired by the research on collaborative scenarios for tabletops, focusing, for example, on the flow of information between users, or possible applications of tabletops in real world. This paper focuses on leveraging multidevice interaction for collaborative data exploration though visualisation. Thus, it appears to be an interesting approach to make use of space around and in-between devices. We decided to investigate it in the field of collaborative data exploration and understanding. It seems that in several years the technology enabling the device to sense its close surroundings will be embedded into commercially available devices. Past research focused mostly on creating distributed displays, like in Conductor [3] or HuddleLamp [5], or on expanding the model of control and navigation by using the relative position of devices [7].

Figure 1: Users exploring data in the preliminary paper-based study. The different pieces of paper contain clues in a crime mystery game.

Preliminary inquiry

People in the course of data exploration tend to organize the information spatially, even though the information itself does not necessarily contain any spatial aspects [4]. They find relations (using temporal, person-wise or other abstract criteria) and them to a plane or space. In order to investigate how this approach may be translated into interaction design domain, a study with three participants was conducted. They were given a set of clues and asked to solve a crime mystery, as shown in Figure 1. The entire study was recorded and the video material were analyzed to find how users manipulate pieces of information on the table. This preliminary study showed that the proxemics, i.e. spatial distance, orientation and other parameters between the pieces of information is meaningful in this process. The analysis shows that objectification, i.e. connecting abstract concepts with physical object is important.

Methodology

We believe that using multiple, spatially aware mobile devices acn support collaborative sensemaking by offering more effective methods of data visualization. Sensemaking is still often performed using sheets of paper, post-it notes or other physical objects. The performance in understanding may be limited by the simplicity of the tools. Translating this process to digital world opens a wide range of possibilities to make it more effective, for example by giving the user possibility to create multiple arrangements of data simultaneously. Furthermore, managing user attention by highlighting a certain relation between pieces of information in spatial conditions can benefit sensemaking. Although we consider a set of problems where human experts are required, e.g. the data given may be partly analyzed automatically, using well-known methods and algorithms.

This scenario engages the entire space around the user and is not limited to the device displays. The space between devices becomes part of the interactive space and the spatial relations between devices become parts of the user interface. This enables examining the spatial organization of pieces of data as clues for content-related relations between information pieces. As the process of understanding consists of creating an associative network, which constitutes the multidimensional, fuzzy relations between pieces of information, it would seem that organizing the information in space and the process of understanding are closely connected. We are planning a study, in which we will observe users during the process of processing a given data set, e.g. creating a complex associative network of mutual relations between pieces of information they explore. Then, by analyzing the obtained study data, we will try to build a model of how users manage the information in space and time. Based on that we will create a system, which will be able to predict the possible relations between pieces of information based on observation of spatiotemporal parameters of data in the interactive interface.

We believe that we can create a fuzzy mathematical model for translating the spatio-temporal information to an associative network of interactions. A example of operation of such model can be a situation when two pieces of information are placed close to each other for certain period of time. By parameterizing the numerical values of distance and time, we can create a relation between two pieces of information. Based on the relations between information and data, a fuzzy inference model can be applied to transfer the spatial relations to relations between pieces of information.

An important feature of the aforementioned mathematical model is considering the history of the movement of a piece of information in the process of inference. This means that the strength of the relation does not only depend on the current position and time spent in this position, but also on the previous movements of the piece of information. One way the system could help in sensemaking would be by visualizing how strongly two objects are related if the value of the (model-inferred) relation exceeds a specific value, as shown in Figure 2. Creating such an intelligent system could open new possibilities for supporting sensemaking such as enhanced attention management. Such systems could also lower the mental demand of communicating insights between users, as they could be shared implicitly, similarly to scenario shown by Goyal et al. [2].

The advantage of implementing such solutions on mobile devices is that tablets and smartphones are physical, tangible objects. Tangibility offers additional advantages to interaction e.g. the socioconstuctivist flavor of tangible tabletops. Also, the system provides the possibility of tracking the proxemics not only on a plane, but in three (or "two and

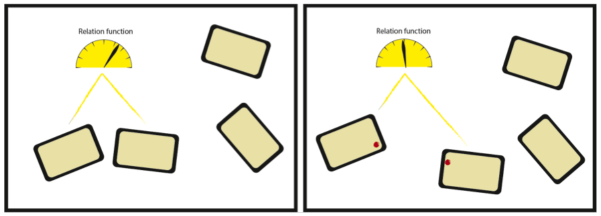

Figure 2: Illustration of a possible application of a relation function. If two devices are close, the value of relation increases. When the distance between device increases, the value of the relation slightly decreases, but the relation is still stored in the system and visualised on the screens.

a half") dimensions. Overlapping may be meaningful in this context. We believe that such enhanced systems would find applications in many fields, such as thematic or visual analysis, decision making and other areas, where human experts and their performance is crucial.

Conclusions and further work

This paper proposes a new way of designing interactive systems for mobile devices, leveraging upcoming technology. We suggest incorporating proxemics in context-aware adaptive interactive systems for multiple mobile devices, which can contribute to more effective access to information visualization, better collaboration and easier sensemaking. With this position paper, we aim to inspire further work in this area. We recommend further efforts for creating specific design methodologies and methods for examining and creating user behavior models that incorporate device proxemics in the process of sensemaking.

Acknowledgments: This work is partially funded by DFG within SimTech Cluster of Excellence (EXC 310/2) and by the EC's H2020 Program: "FETPROACT-1-2014: GSS", # 641191 CIMPLEX. P. W. Woźniak thanks the Adlerbertska Research Foundation for its support for this research.

REFERENCES

1. 2014. U.S. Smartphone Use in 2015 | Pew Research Center. (2014). http://www.pewinternet.org/2015/04/01/us-smartphone-use-in-2015/ Accessed: 2015-08-30.
2. Nitesh Goyal, Gilly Leshed, Dan Cosley, and Susan R. Fussell. 2014. Effects of implicit sharing in collaborative analysis. In *Proc. CHI '14*. ACM Press, New York, USA. DOI: http://dx.doi.org/10.1145/2556288.2557229
3. Peter Hamilton and Daniel J. Wigdor. 2014. Conductor: enabling and understanding cross-device interaction. In *Proc. CHI '14*. ACM Press, New York, USA. DOI: http://dx.doi.org/10.1145/2556288.2557170
4. Thomas W. Malone. 1983. How Do People Organize Their Desks?: Implications for the Design of Office Information Systems. *ACM Trans. Inf. Syst.* (1983). DOI: http://dx.doi.org/10.1145/357423.357430
5. Roman Rädle, Hans-Christian Jetter, Nicolai Marquardt, Harald Reiterer, and Yvonne Rogers. 2014. HuddleLamp: Spatially-Aware Mobile Displays for Ad-hoc Around-the-Table Collaboration. In *Proc. ITS '14*. ACM Press, New York, USA. DOI: http://dx.doi.org/10.1145/2669485.2669500
6. Paweł Woźniak, Lars Lischke, Benjamin Schmidt, Shengdong Zhao, and Morten Fjeld. 2014a. Thaddeus: a dual device interaction space for exploring information visualisation. In *Proc. NordiCHI '14*. ACM Press, New York, USA. DOI: http://dx.doi.org/10.1145/2639189.2639237
7. Paweł W. Woźniak, Benjamin Schmidt, Lars Lischke, Zlatko Franjcic, Asim Evren Yantaç, and Morten Fjeld. 2014b. MochaTop: building ad-hoc data spaces with multiple devices. In *Proc. CHI EA '14*. ACM Press, New York, USA. DOI: http://dx.doi.org/10.1145/2559206.2581232

Using Semantic and Responsive Web Design Technologies for Cross-Device Interactions in Industrial Applications

Jens Ziegler
Markus Graube
Sebastian Heinze
Stephan Hensel
Leon Urbas

Technische Universität Dresden
01062 Dresden, Germany

jens.ziegler@tu-dresden.de
markus.graube@tu-dresden.de
sebastian.heinze1@tu-dresden.de
stephan.hensel@tu-dresden.de
leon.urbas@tu-dresden.de

Presented at the Cross-Surface '15 workshop, in conjunction with ACM ITS'15.
November 15, Funchal, Madeira, Portugal.

Abstract

Many industries today urgently call for applications that provide data and software services to mobile users on-site. Complex workflows spanning across different contexts of use, using different interaction devices and involving different people need to be supported seamlessly. Such collaborative cross-device interactions call for Semantic and Responsive Web Design technologies. In this contribution, we will explain why we need these technologies, how we may employ them and what we expect from their utilization. We will present a functional prototype that demonstrates the current state of our research and highlights the remaining challenges.

Author Keywords

Cross-Device Interaction; Responsive Web Design; Semantic Web; Industrial Application; Computer Supported Cooperative Work

ACM Classification Keywords

H.5.2. Information interfaces and presentation: User Interfaces: Graphical user interfaces (GUI)

H.5.3. Information interfaces and presentation: Group and Organization Interfaces: Computer-supported cooperative work

Introduction

Industrial applications pave their way from the office out into the factories, warehouses and chemical plants. Mobile applications offer services to support commissioning, operation and maintenance. Such applications are increasingly engineered with two main design objectives in mind. First, users shall be able to seamlessly switch between different target devices to accomplish their tasks. Depending on the task and context at hand, target devices may be office PCs, smartphones or even wearables. Data and software services may be provided anywhere and anytime (*on-site support*). Second, spatially and temporally separated people shall be able to form teams and jointly collaborate and support each other regardless of their current location (*remote support*).

These design objectives require highly flexible, dynamic, integrated and adaptive user interfaces. The integration needs to start already at the information level and must find its continuation on the interaction level. The large heterogeneity and complexity of industrial software tools and frameworks, the immense variety of the contexts of use as well as the specific requirements of professional use place high demands on performance, availability, safety, security and usability of industrial applications. Thus, the use of established, proven and well understood open software technologies instead of today's proprietary, platform and vendor-specific solutions is more than necessary. The speed of development of production processes on the one hand and of information and communication technology (ICT) on the other hand further calls for the use of long-term available and continuously developed, worldwide deployed standard technologies in combination with an explicit Design for Evolution.

Mission Statement

The use of global World Wide Web standards and technologies is the logical answer to these demands. In particular, the Semantic and Responsive Web Design technologies seem to provide solutions to the challenges in implementing appropriate future industrial applications. Breslin et al. have shown that Semantic Web technologies can be deployed to supply, production and order fulfillment processes [1]. Jetter et al. could show that cross-device interactions can be used to collaboratively perform a task in a central controlled use case [2]. Further, web-based platforms can be used to create cross-device solutions for collaborative, multi-role tasks in industrial settings as well [3, 4]. However, none of these works have really strived for creating integrated information *and* interaction spaces by combining the both technologies.

Our mission thus is 1) to create an information and interaction space for migratory, collaborative industrial applications; 2) to combine Semantic and Responsive Web Design technologies to realize such applications; 3) to use context information beyond screen properties to adapt the user interfaces to the users' needs; and 4) to integrate resulting applications into the existing industrial ICT landscape to prove the applicability of the approach.

The research presented in this contribution is grounded on a series of focus group workshops and participatory observations with industrial partners carried out over a period of several years. A Rapid Software Prototyping approach has been employed to create horizontal, functional prototypes on the target platforms (see e.g. [5]). Emphasis is placed on the use of established design standards and best practices.

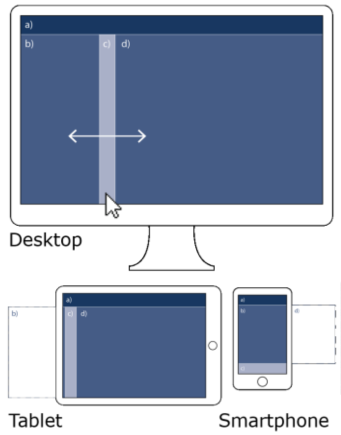

Figure 1: Structural layout for different device classes. Elements are re-arranging automatically.

Figure 2: Description of the UI elements for desktop screens. Other screens adapt according to Figure 1.

Creative Vision

The creative vision of our research is an integrated information and interaction space that can be accessed by multiple persons with multiple roles using a variety of interaction devices in order to collaboratively cope with a complex task. Therefore, we aim at creating a responsive Computer Supported Cooperative Work (CSCW) system that automatically adapts to device properties, context information, user role and the task at hand.

The integrated information space shall provide shared access, flexible modification of both structure and content including the integration of external information spaces as well as a sophisticated revision management. All information models should be self-describing and analyzable by computer algorithms. The integrated interaction space shall support a holistic responsive cross-device interaction providing appropriate views for various interaction devices. A role and task management system may adapt the content that is provided to the user. This includes access control but also automatic content selection and preprocessing as well as an adaptation of the visualization to the task and role at hand. The current task may be predefined by means of a workflow description or derived from the available context information. A job management system may further group data sets, states and conversations that belong to a particular job, thus creating a common context for collaborative tasks. Each user can be assigned to multiple jobs, and multiple users, possibly with different roles, can be assigned to one job as well. The application hence allows the user to switch between assigned jobs, so that a single, yet extensible, application running on multiple devices serves as common user interface for all users.

System Prototype

As a demonstrator, we are currently developing a prototypical CSCW system for operation and maintenance (O&M) in the domain of the process industries. The prototype provides cross-device interaction and activity-based presentation of information. The prototype supports target screens of three different device classes: desktop monitor, tablet and smartphone (Figure 1). Currently, the screen orientation is fixed to portrait view for smartphones and landscape view for the other two. The employed ZURB Foundation responsive front-end framework [6] automatically adapts the main elements of the structural layout (Figure 2) to the screen properties and contextual information such as geolocation. Three different roles have been considered: a shift leader that is responsible for organizational tasks and thus is mainly working in the office, an operator that is responsible for the supervisory control of the plant operation and thus is mainly working in a control room, and a maintainer performing the maintenance and service tasks on-site using mobile devices.

The prototype currently provides six different functions: a Process Operator Screen (Figure 3), a Batch Process and Workflow Modeler, a Task Organizer, an Alarms and Events List, a Chart Viewer and a Communicator including chatroom and dialer (Figure 4). The application might provide more than one view per function if the task varies according to the actual user role. For example, the shift leader may create and categorize tasks in the Task Organizer, whereas maintainers can assign themselves to existing tasks that fit to their particular skills and knowledge. Hence, the two roles require different data and dialogs to collaboratively accomplish their task being the organization of their maintenance tasks.

Figure 3: Process Operator Screen views for desktop and smartphone. The process data of the plant is shown in the Content Area (d) and can directly be modified in the Context Area (b).

Figure 4: Communicator views for tablet and smartphone. The Content Area (d) shows an ongoing conversation, the Context Area (b) (shown on the tablet only) provides the contact list with the conversation partners.

The information space is realized as a Linked Data (LD) cloud using standard Semantic Web technologies [7]. The LD cloud aggregates data from several proprietary software tools and exposes all distributed data in a uniform and self-describing manner. The main advantage of this approach is the simple confederation of heterogeneous data sources, both structured and unstructured. Having meta-information as a first citizen data opens new perspectives for flexibility, extendibility and maintainability [8] of applications.

Future Research

Our preliminary findings indicate that great care must be taken in selecting appropriate Web standards and technologies. Hence, a first major research objective is to identify high-quality Web frameworks that meet the demands of the industry in terms of reliability, maintainability and long-term support.

Necessary foundation for holistic responsiveness is an integrated information space. We will continue our efforts to integrate the diverse data sources of an industrial IT landscape into LD clouds and to define common information models (*ontologies*) representing these data. In addition, we are working on data interfaces to LD clouds that meet the particular requirements of the industry with respect to performance, security, reliability and traceability on the one hand, and that perform well in combination with Responsive Web Design technologies on the other hand (see e.g. [9]).

The current prototype is based on a preliminary conceptual design framework. As a next step we will refine and generalize this framework, characterize and categorize the different possibilities for UI adaptation to the various factors, and draw up recommendations for their use and possible coaction. Further, we will investigate in the utilization of information on situational and environmental context to improve responsiveness. We will also integrate interactions which enable users, to easily push information and application context back and forth between devices in order to leverage effective working in multi-device settings, both in single-user and multi-user settings.

References

1. Breslin, J. G. et al. 2010. Semantic Web computing in industry. *Computers in Industry, 61-8*: 729-741.
2. Jetter, H.C. et al. 2013. Collaborative interactions in future crisis rooms. *Proc. JRC ECML WS*: 35-39.
3. Yang, J. et al. 2014. Panelrama: Enabling Easy Specification of Cross-Device Web Applications. *Proc. ACM CHI 2014*: 2783-2792. ACM.
4. Badam, S. K. et al. 2014. PolyChrome: A Cross-Device Framework for Collaborative Web Visualization. *Proc. ACM ITS 2014*: 109-118. ACM.
5. Ziegler, J. et al. 2012. Beyond app-chaining: Mobile app orchestration for efficient model driven software generation. *Proc. IEEE ETFA 2012*: 1-8. IEEE.
6. ZURB Foundation responsive front-end framework. Online: http://foundation.zurb.com/
7. Graube, M. et al. 2011. Linked Data as integrating technology for industrial data. *Proc. NBiS 2011*: 162-167. IEEE.
8. Ziegler, J. et al. 2014. Implementation and Operation of Collaborative Manufacturing Networks. *Collaborative Systems for Smart Networked Environments*: 197-208. Springer Berlin Heidelberg.
9. Graube, M. et al. 2015. R43ples: Revisions for Triples. *Proc. LDQ 2015*. CEUR-WS.

Device Boundaries: Posture in Interaction Ecologies

Daniele Savasta
Yasar University,
Izmir, Turkey
daniele.savasta@gmail.com

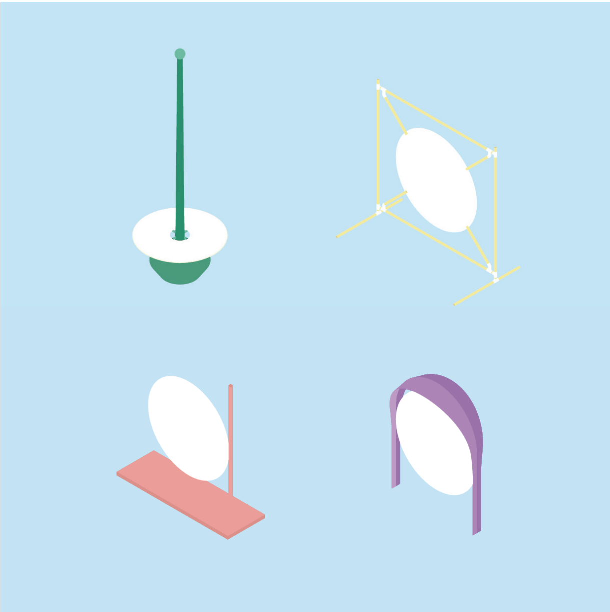

Figure 1 Device Boundaries project, illustration of possible configurations.

Presented at the Cross-Surface '15 workshop,
in conjunction with ACM ITS'15.
November 15, Funchal, Madeira, Portugal.

Abstract
The number of interactive artifacts that surround us is constantly increasing. Though these device-species constitute new ecologies with humans, the communication with each other is very limited. People and devices involved in an interactive environment can, and I claim should, assume different roles that are yet to be defined. The traditional model of single-user single-device is becoming obsolete in the contemporary distributed communicational space. To explore these complexities, I developed a project that tries to bring these issues together: a multi-user, multi-device ecology constituted by some devices available on the market and some devices designed on purpose. The roles of people and devices changes according to the application in use. This experimentation helps to explore the issues raised here, and in previous studies. In particular, it shows the weaknesses and strengths of this analyzed ecology, it outlines possible paths for future applications and for helping in the definition of a relational quality and the use of roles as a tool of thought.

Author Keywords
interaction qualities; distributed user interfaces; cross-surface interaction; collocated interaction.

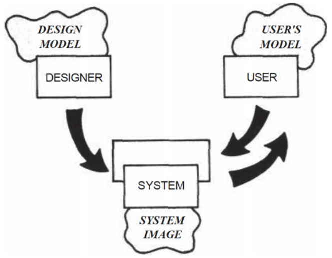

Figure 2: "Conceptual Models". This model from Norman [6] appears in the influential best seller "The Design of Everyday Things" [7].

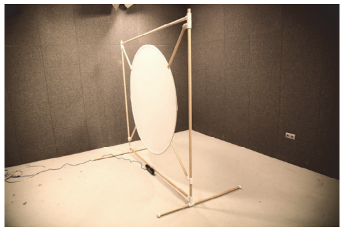

Figure 3 Membrane prototype. The prototype is made by a structure in wood and a circular display in elastic textile.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction

In the last ten years, with the introduction in the market of smart objects, or better, mobile internet devices (MIDs), we populated our lives with an enormous quantity of *asocial* technical species. Supposedly "smart" phones, tablets and more recently other wearables are designed in a user-centered fashion. The commonly adopted model refers to the one proposed by Norman (Figure **2**2) in which a single user is facing a single device. It was a model worthy of use for the time being. The *interaction qualities*, or *use qualities*, as introduced by Löwgren [3] were also focused on the relation single user-interface. Today, the communicative capacities of the devices, and their simultaneous use, goes way beyond this one-to-one equivalence and so, the need for an improved model and relative qualities arises. In a recent research proposal Lundgren [4] introduces a framework of qualities for collocated interaction (**Figure 4**3) and highlights the need for a reflection that goes in the direction of a more inclusive paradigm. The four perspectives assumed (social, technological, spatial and temporal) gather the different thirteen qualities. This first attempt defines an overview of possible qualities opening opportunities for further studies. In this general perspective, I introduce the concept of *posture* in device ecologies supported by a project called Device Boundaries. Device Boundaries is an experimental ecology that involves multiple users and multiple devices. The roles of both species are fluctuating according to the context and the application. The posture quality of artifacts analyses the relations between different species with different roles.

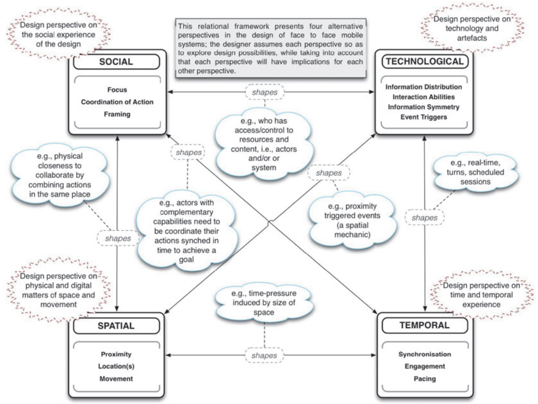

Figure 4 Framework for designing mobile experiences for collocated interaction (diagrammatic representation) [3].

Device Boundaries Implementation

Device Boundaries is a project developed to probe the different positions of artifacts and humans in an interaction ecology. The aim of the study was to define a general purpose tool, but for clarity and testing some applications were realized. The project involves different devices some of them available on the market and some designed on purpose. The artifacts involved in the project are: Antenna (central control system), Membrane (textile touch-display), Smartphones.

Antenna

Antenna is the "brain", the central control system made by a Raspberry Pi with a Wi-Fi antenna, a memory and a battery. It works as a bridge for the other devices providing a web-server for the access to the applications.

Figure 5 Arcade Pong Application on Membrane.

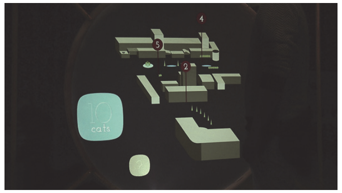

Figure 6 Feline colony Application on Membrane.

Membrane
Membrane is a device constituted by a projector, a large textile display and a Microsoft Kinect (Figure 5). One of the possible configurations was developed for the tests (Figure 1). The pressure of fingers on the screen creates a deformation that is read by the sensor bar making the textile interactive.

Smartphones
Two Android 4.2 smartphones (HTC and Samsung) were used with a custom software developed in Processing.

Device Boundaries is designed for public spaces (such as a street or a square) in order to encourage contact between people by superimposing a digital layer over the physical community. The *Antenna* acts as a server connecting any personal device to the digital environment. As for any activity carried on in a public space the role assumed by the person can vary widely. A person can be deeply involved in the use of the interface, can just observe from far, be a bystander, or actively develop new applications for the system or do maintenance. The posture towards the artifacts and the other people changes radically. In the same way, the devices, as technical individuals, can play different roles: the phone can be used exclusively as a controller thanks to its touch-screen and sensors, the *membrane* can be used as a passive display or accord to the state of the system; the *antenna* can be active, passive, provide data, or even disappear from the ecology, in the case of a headless system. During the design process attention was given to cost and feasibility of the project, considering that the system could be implemented spontaneously by the community in a bottom up approach.

Applications
Some applications were developed for trying the system and its limits. The contents are chosen by the typology of activities in the public sphere defined by Jan Gehl [1]. The two applications developed were focused on *social* and *optional activities*, specifically the first activity focus is play and the second one is the care of environment (in particular the cat community).

Arcade Pong
Arcade Pong is a game clearly inspired by the old videogames as *Pong* (1972) and *Arcade Volleyball* (1988). The game is played as a table-tennis match in which the two players synchronize their smartphones with the system and use them as rackets. The *membrane* display positioned in between the two players shows the position of the “ball”. The single screen plays with size giving the illusion of perspective (Figure 5).

Feline Colony
Feline Colony is an application to keep track of the cat communities in a specific neighborhood (Figure 6). The problem of large communities of feral cat is diffused in Turkey, were the application was developed, but emerged recently in the Australian news [5]. The preservation of the community and its sustainability is a complex activity that includes: sheltering, feeding, health care and population control. In the interest of cats, humans and environment, I developed an application that provides support for these necessities. A database of the existing community and their information is created for the purpose and accessibility to the system is given by the sole use of *Membrane* interface or remotely on smartphone (Figure 6).

Posture

The *posture* quality of interaction ecologies is a property that evaluate the roles of the species involved in an action. It considers the proximity of actors [2] focusing on their role in the ecology created with other humans or devices. The relationships among humans and technical individuals are explored and mapped verbally and visually. The map includes the communication links with other species and their roles in the interaction. We can explain diagrammatically the ecology analysis of the two examples presented (Figure 7 and 8).

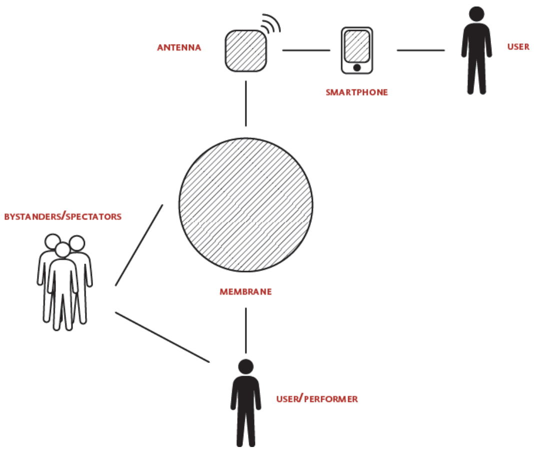

Figure 7 Feline Colony Posture Diagram. The diagram shows the relation assumed by the species in the ecology. A user operates her/his smartphone to access the applications while independently another user operates *membrane* becoming a performer for the bystanders.

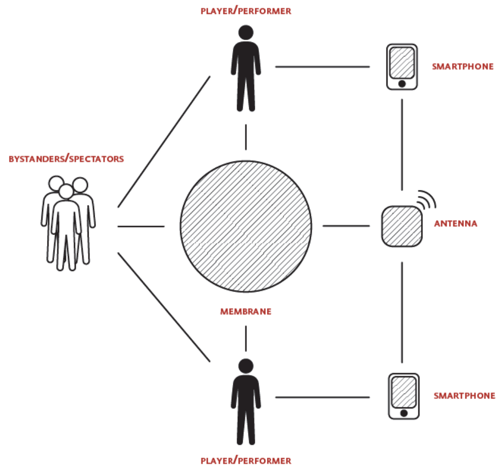

Figure 8 Arcade Pong Postures Diagram. The players user their smartphones as controllers, the bystanders observe the players becoming spectators and making them performers, the Membrane display is both functional to the players and performative for the bystanders.

Conclusions and Future Work

I believe that the *posture* quality, and the relative diagrammatical representation, can help designers in their work for a better development of device ecologies. This concept enriches the framework introduced by Lundgren [4] trying to provide a theoretical answer to the actual device *asociality*.

The project *Device Boundaries* on its own can be developed further to reduce the number of components required. The ecology can be fragmented in more independent elements and constitute a headless system. Finally, a more horizontal hierarchy can improve the flexibility and increase the possible applications of the project.

References

1. Jan Gehl. 1987. “Three Types of Outdoor Activities,” “Life Between Buildings,” and “Outdoor Activities and the Quality of Outdoor Space”. In *The City Reader Fifth Edition,* Richard T. LeGates, Frederic Stout (Eds.). Routledge, London.
2. Saul Greenberg, Nicolai Marquardt, Till Ballendat, Rob Diaz-Marino, and Miaosen Wang. 2011. Proxemic interactions: the new ubicomp?. In *interactions*, 18(1), pp.42-50.
3. Jonas Löwgren. 2002*. The use qualities of digital designs*. From http://www.ixdfactory.net/ixdfactory/Handouts_files/Lo%CC%88wgren_TheUseQualitiesofDigitalDesigns.pdf
4. Sus Lundgren, Joel E. Fischer, Stuart Reeves, and Olof Torgersson. 2015. Designing Mobile Experiences for Collocated Interaction. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing* (CSCW '15). ACM, New York, NY, USA, 496-507. DOI=10.1145/2675133.2675171 http://doi.acm.org/10.1145/2675133.2675171
5. Oliver Milman. Australian government declares war on feral cats in bid to save native animals. *The guardian*, 2015. http://www.theguardian.com/environment/2015/jul/16/australian-government-declares-war-on-feral-cats-in-bid-to-save-endangered-species.
6. Donald A. Norman. 1987. Cognitive engineering—cognitive science. In *Interfacing thought: cognitive aspects of human-computer interaction*, John M. Carroll (Ed.). MIT Press, Cambridge, MA, USA 325-336.
7. Donald A. Norman. 2002. *The Design of Everyday Things*. Basic Books, Inc., New York, NY, USA.

Mixed Reality Environments as Ecologies for Cross-Device Interaction

Jens Müller
University of Konstanz
78464 Konstanz, Germany
jens.mueller@uni-konstanz.de

Harald Reiterer
University of Konstanz
78464 Konstanz, Germany
harald.reiterer@uni-konstanz.de

Abstract
In Mixed Reality (MR) environments, virtual objects can be represented as if they were situated in the viewer's physical environment. While the potentials of MR have been recognized and extensively researched for single user scenarios (e.g., in perceptual studies), MR for collaborative scenarios has not been widely investigated. In this paper we propose MR environments as ecologies for collaborative, cross-device interaction. We provide a scenario that illustrates its potentials and discuss possible research directions. We then present intermediate results of our research.

Presented at the Cross-Surface '15 workshop, in conjunction with ACM ITS'15.
November 15, Funchal, Madeira, Portugal.

Author Keywords
Mixed Reality; collaboration; cross-device interaction.

ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

Introduction

Mixed Reality (MR) describes the combination of the representation of a physical environment (e.g., a room) and virtual objects (e.g., a virtual plant in a corner of that room) on a single display [6]. Because the virtual objects have a distinct position in the real-world coordinate system, they are perceived as if they were situated in the real world. This allows the user to interact with the real world and the digital world at the same time [1]. When virtual objects are experienced as part of the physical environment, MR environments can leverage our natural abilities that refer to the interaction and navigation in the real world. This includes the perception of spatial relationships of the objects in our environment, but also the social skills we've developed in the physical world (e.g., social protocols). In addition, in MR our physical environment can be considered as an information space in which we can lay out, navigate, and share digital data. This is particularly relevant for collaborative scenarios, as it allows for seamless, computer-supported, collaborative work [2].

Collaborative MR in the early (research lab) days:

Figure 1. "Transvision" [7], a design tool where collaborators look on a real table-top through a palmtop-sized see-through display. 3D data from the display is processed by an external graphics workstation.

Figure 2. "Studierstube" [8], a system capable to visualize three-dimensional, scientific data with see-through glasses. Tracking data is processed on an external tracking server.

To render virtual content correctly, MR devices need to be able to determine their location and orientation within their physical environment. There are several approaches to achieve this spatial awareness. (For an overview see [1]). Precise tracking within larger 3D volumes, however, typically required additional hardware, such as infrared cameras (Figures 1 and 2). Due to technological advances, displays for large-scale MR environments no longer require additional hardware and are now becoming affordable to the public. One of the recent devices is the Project Tango tablet (Figure 3). Spatial awareness is achieved by the principle of area learning: *"Using area learning, a Project Tango device can remember the visual features of the area it is moving through and recognize when it sees those features again. These features can be saved in an Area Description File (ADF) to use again later."*[1]

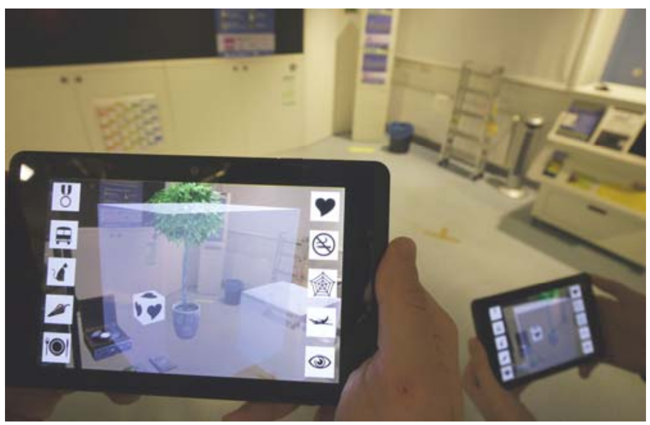

Figure 3. Collaborative MR currently: markerless tracking with Google's Project Tango[1] tablets. In this application, each person can place items on the shared information space and has their own perspective.

[1] https://www.google.com/atap/project-tango/

When an ADF is loaded, the device can localize, i.e., it becomes spatially aware. We consider this a core feature to enable MR-based, cross-device interaction: When collaborators within the same physical environment share the same ADF, their devices become spatially aware in terms of both their physical environment and of the other devices. The physical environment can then be used as a shared information space.

Chances and Challenges

To illustrate the potentials of MR environments as ecologies for collaborative cross device interaction in everyday situations, consider the following scenario:

Bob (an architect) and Alice (a civil engineer) are involved in an architectural project that aims at constructing a new central station. They meet at Alice's office to discuss Bob's latest drafts of the barrier-free main entrance. As Bob enters Alice's office, his tablet notifies him that an ADF is available for the current environment. After the ADF has loaded, he looks on his tablet and sees a virtual construction site laid out in Alice's office. Alice is standing in front of the window where the main entrance is supposed to be placed on the site. Bob opens the local folder on his tablet where he stores his drafts. He walks towards Alice and positions his 3D draft, which is now a part of the MR environment and thus visible to Alice. After a short discussion, Alice picks Bob's newly added model to make some modifications on it later on her computer. Bob in turn asks Alice for some project related text documents. Alice browses her local file system on the tablet and tells Bob that she was going to position them over her little houseplant (she points at the houseplant on her desk). Bob instantly sees the documents on his *display, which appear as a virtual stack of papers, and stores them on his file system.*

In the following we share our first ideas that may help to make such scenarios possible and propose associated research directions.

Technological Challenges

First, in order to enable participation in cross device activities, devices need to be able to communicate with each other. Thus, cross-platform technologies (e.g., HTML 5) should be used. Secondly, besides the spatial capabilities of devices described in this paper (e.g., through ADFs), there are other features and sensors that enable other forms of cross-device interaction. Therefore, profiles or a classification of device capabilities are needed. These profiles should summarize the features that are relevant for cross-device interaction. Following the example scenario, imagine a third person – Carol – is joining the meeting. Unlike Alice's and Bob's devices, Carol's device cannot handle area description files but has some other sensors (e.g., a proximity sensor) that can be used to establish cross device interaction in a different way. If such profiles are provided, a server could suggest to Carol an alternative way to establish a connection to the other devices. A third aspect refers to the type of task. Certain cross-device interaction techniques may be more appropriate for specific tasks than others. Once a server has registered the profiles of the present devices and the anticipated task, it could make suggestions considering the way interaction between the devices should be established.

Research Directions

On a conceptual level, spatial relations between the entities in an MR environment (persons and other physical and virtual objects) can be taken into account to facilitate interaction and collaboration. In particular for MR-based, cross-device interaction, proxemics dimension [4] and F-formations [5] could be used to trigger situation-dependent actions, e.g., render specific content at specific positions depending on the spatial and personal constellation of present collaborators. Furthermore, for tasks that require negotiation, visual cues play an important role [3] and help coordinating users' actions. Users who interact within the same MR environment can thereby make use of the same spatial cues. This raises the question what features and representation need to be available in MR environment in order to leverage "seamless, computer supported collaborative work" [2].

Ongoing Research

In our ongoing research, we are interested in the representation of MR environments, in particular in how virtual cues (Figure 4) shape communication and coordination in search and reconstruction tasks. We conducted a controlled lab experiment with 16 dyads. The experiment was designed as a counterbalanced, within-subjects design with the presence of virtual cues (cues provided vs. no cues provided) being the independent variable. For MR devices we provided each participant a Project Tango tablet which allowed a shared view on the MR environment.

Figure 4. Top: Dyads during the search task. All boxes are covered in this state. Lower left: no virtual cues provided, lower right: virtual cues provided (e.g., virtual chair, plant, and snack machine). Collaborators made extensive use of virtual cues to communicate spatial information (e.g., “I remember that symbol ‘x’ is located in front of the snack machine”).

Search task: Dyads had to collaboratively solve a three-dimensional memory game. 10 symbol pairs from the Wingdings font were randomly distributed in a 3D volume with a dimension of 4m x 4m x 2m. Each symbol was represented as a texture on a box. In their initial state, boxes where white (Figure 3) and could be “uncovered” by touching them so that the symbol became visible. Collaborators had to find matches by uncovering two boxes per move. In order to induce the communication of spatial information, each collaborator had to uncover one box during each turn. Once a match was found, the boxes were removed. If the two uncovered boxes were not matches, they had be covered again in order to continue with the next move.

Reconstruction Task: In the reconstruction task, collaborators had to reconstruct the virtual scene by placing the symbol boxes at the correct position (Figure 3). The task was performed in the same condition (virtual cues provided or virtual cues not provided) as in the prior search task.

Intermediate results and implications: Our intermediate results show that all groups made extensive use of virtual cues to communicate object locations and to coordinate their actions. In the concluding interview, participants reported to have fully accepted the virtual objects as part of the environment. In addition, all groups gave the condition with the virtual cues a better rating. We therefore propose to provide virtual cues to support collaboration in MR environments.

Conclusion

In this paper we proposed MR environments as one possible approach to establish collaborative, cross-device interaction. We presented recent technological advances and discussed potentials and challenges for ad hoc, MR environments in everyday situations. Finally, we reported intermediate results from our ongoing research in which we suggest the use of virtual cues in MR environments to facilitate collaboration.

References

1. Mark Billinghurst, Adrian Clark, and Gun Lee. 2015. A Survey of Augmented Reality. *Foundations and Trends® in Human–Computer Interaction* 8, 2: 73–272.
2. Mark Billinghurst and Hirokazu Kato. 1999. Collaborative Mixed Reality. *Proc. of ISMR '99*, Springer, 261–284.
3. Alphonse Chapanis. 1975. Interactive Human Communication. *Scientific American* 232 (3): 36–42.
4. Saul Greenberg, Nicolai Marquardt, Rob Diaz-marino, and Miaosen Wang. 2011. Proxemic Interactions : The New Ubicomp ? *interactions* XVIII, february: 42–50.
5. Nicolai Marquardt, Hinckley, and S Greenberg. 2012. Cross-Device Interaction via Micro-mobility and F-formations. *Proc. of UIST 2012*, 13–22.
6. Paul Milgram and Fumio Kishino. 1994. Taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, 1321–1329.
7. Jun Rekimoto. 1996. Transvision: A hand-held augmented reality system for collaborative design. *Proc. of VSM '96*, 85–90.
8. Dieter Schmalstieg, Anton Fuhrmann, Zsolt Szalavari, and Michael Gervautz. 1996. "Studierstube" - An Environment for Collaboration in Augmented Reality". *Proc. of CVE (Extended abstract)*, 19–20.

Toward Utopian Multi-Surface Interactive Systems

Cornelius Toole
Department of Mathematics &
Computer Science
Tougaloo College
Tougaloo, MS 39174, USA
ctoole@acm.org

Abstract
Once a designer or developer decides to build an interactive experience that involves two or more systems, they now have the challenge of trying to make one place out of multiple. This research is aimed at identifying tools that support cross-surface interactions without requiring designers and developers to prematurely address the friction of cross-device communication. I believe this can achieved (1) by providing high-level abstractions that can be mapped onto multiple configurations of concrete interactive system components and (2) by providing low-level infrastructure to combine concrete resources in very flexible ways. I propose that interaction with multi-device ecologies can be further propagated in the wild when we use such approaches to increase the probability that our individual contributions can be integrated with the contributions of others both in research and industry.

Presented at the Cross-Surface '15 workshop, in conjunction with ACM ITS'15

Author Keywords
Tangible, multi-device interaction, multi-surface interaction, distributed user interfaces, distributed heterogeneous user interfaces

ACM Classification Keywords
H.5.2. User Interfaces

Introduction
The physical and digital objects within our interactive world are diverse and represent great, underutilized potential in the ways in which they can be combined. We have gotten quite use to the interactions that can be had on these devices. But we face friction when we try to interact across these devices, especially when these devices differ in their type (e.g. smartphones and smart-watches, tangibles and interactive surfaces) and/or underlying technology platform (e.g. Web vs. native, iOS vs. Android, tangibles vs. wearables). My

Utopian Multi-Surface Parameters (heavily influenced by MCR_{it}[15])

- **U:** the set of users
- **M:** the set model objects represented by system
- **R:** the set of representations in the system
- **C:** the set of controllers to link models and representations

Real World Multi-Surface System Parameters

A concrete multi-surface interactive system model would include **U,M,C,R** plus

- **D:** the set of computational domains systems components would be deployed
- **L:** the set of locales system entities could be distributed across
- **P:** the set of real device platform properties

research position on interaction with multi-device ecologies is that we need to work toward providing post-desktop interaction developers and designers with ways to build multi-surface interaction without having to prematurely commit to technical decisions about the implementation of computation and communication across these devices. I believe we can model the characteristics of utopian[1] cross-surface interaction as well as design an architecture by which running instances of that model can be built (see sidebar for the properties under consideration).

For remainder of this document I will describe efforts toward realizing a more utopian multi-surface interactive world and the problems that motivated that research. At a high-level, the ideas discussed here should be applicable to the many subgroups in the ITS community (tangibles, tabletops, wearables, etc.), but my research has largely been from the perspective of toolkits for tangible user interfaces.

Prior Research

Many of my prior research experiences in this space involved the development of tangible user interfaces (TUIs or tangibles) in support of science application users carrying out visual data analysis tasks (see **Figure 1:** Visualization Exploration Tangibles). Early on, any change in the technology (e.g. the hardware underlying the tangibles, the software that drove the visualization, the communication infrastructure, etc.) required significant reengineering and design. These experiences led me to this line of questioning: What would it take for a tangibles-driven visualization application to be able to vary in its system configuration parameters[3] without having to change code that defines the application's high-level interactive structure and behavior?

These questions along with questions of the how to make network programming more accessible to post-desktop interaction developers and designers led me to seek out communication infrastructure that would yield more flexible, yet performant multi-device systems [4]. I also sought ways to expose this infrastructure through network transparent APIs. Early iterations of this work were used in two semesters of a undergraduate course in interface design and technology [16] and in research systems involving multi-device, multi-modal interaction in remote distributed contexts [5]. Insights from these experiences went into the development of TUIKit, a tool for developing tangibles with distributed, heterogeneous components[13].

From there I worked on trying to generalize the ideas in TUIKit and examine what properties systems based on this approach would have. This resulted an architecture based on a candidate design pattern called the Proxy Tangible Interactor (PTI). PTI-based applications consist of collections of objects that act as proxies for actual resources that provide implementations of the functionality exposed by the proxies' APIs. PTI employs adapters to mediate the proxy-resource bindings by translating between the generic event and command messages of interactive system resources and the schema for data and control of specific resource

[1] Here we mean utopian as in *no place*, not necessarily an *ideal place*.

[3] By configuration we mean the number, type and locality of the users, computational systems and UI elements.

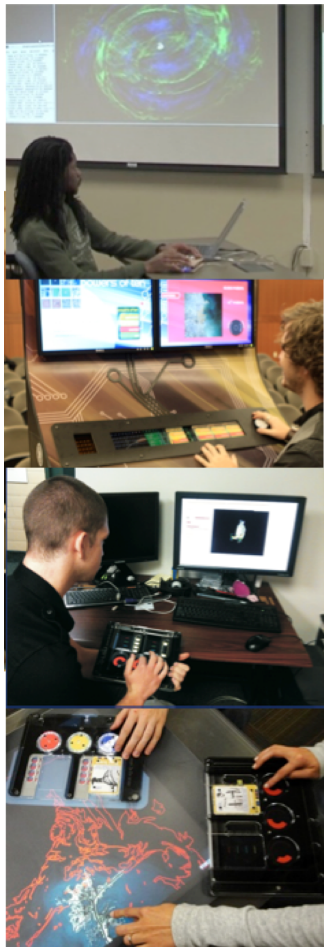

Figure 1: Visualization Exploration Tangibles

implementations. For more details on the design goals and qualities of the PTI architecture see [14].

Related Work

Using the banner of reality-based interaction to unify several post-desktop interaction paradigms (e.g. tangibles, interactive surfaces, etc.), we find several examples of the toolkits and frameworks that address the challenges of RBI-based application development [6]. Klemmer and Landay introduce the *tangible interactor* abstraction to hide details of input device management and enable the developer to focus on the task of defining what the interaction facilitated through the device would mean within the target software. Our framework extends the interactor abstraction for use in distributed systems. Several frameworks address the development of the distributed RBI-based systems. Several use a dataspaces-based approach [1,7,12]; while others user a client-server based approach [8,9,10,18]. This work is very much inspired by several ubiquitous computing toolkits [2,17]. These systems represent innovations that progressively decouple systems spatially, then temporally and eventually to some degree, semantically, enabling the spontaneous integration of interactive system resources in very fluid manners. We also find several frameworks in the literature represented concurrent and subsequent efforts toward support for flexible RBI-based systems. From Fielding's work on the network application architectural styles, we adopt a software engineering technique to evaluate the existing design and guide further development of TUIKit [3].

Future Work

The work on TUIKit and PTI resulted in support for configurable applications that can integrate various interaction devices (see **Figure 2:** Devices supported by TUIKit). Plans for future work include:

- To further develop a utopian multi-surface interaction model along with functional mappings onto actual devices
- To publish specifications to allow others to add support for their own hardware and software within this ecosystem
- To adopt the TUIO protocol for greater interoperability with existing tangible and interactive surfaces systems.
- To develop an implementation of TUIO with support beyond point-to-point communication patterns for greater flexibility
- To develop adapter and proxy interfaces to integrate more interaction techniques with emphasis placed on motion-tracking and more output modalities
- To explore the use PTI/TUIKit to implement other models of multi-surface interaction such as ROSS or Proxemics [11,18].
- To develop support for progressive enhancement of interaction across surfaces with varied capability.

Acknowledgements

This work has been supported in part by NSF MRI-0521559, IIS-0856065, EPSCoR RII-0704191, La. BoR LEQSF (2008-09)-TOO-LIGO Outreach, and the Howard Hughes Medical Institute. Thanks to my advisor Prof. Brygg Ullmer, LSU, the CCT and Tougaloo College.

References

1. Ballagas, R., Ringel, M., Stone, M., and Borchers, J.iStuff: a physical user interface toolkit for

Figure 2: Devices supported by TUIKit

ubiquitous computing environments. *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (2003), 544.

2. Dey, A.K., Salber, D., and A, G.D.A.conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction (HCI) Journal 16*, 2-4 (2001), 97–166.
3. Fielding, R.T. and Fielding, K.Architectural Styles and the Design of Network-based Software Architectures. *Technology*, (2000).
4. Hintjens, P.ZeroMQ - The Guide. 2015. http://zguide.zeromq.org/page:all.
5. Hutanu, A., Schnetter, E., Benger, W., et al.Large scale problem solving using automatic code generation and distributed visualization. *Scalable Computing: Practice and Experience 11*, 2 (2010).
6. Jacob, R. and Girouard, A.Reality-based interaction: a framework for post-WIMP interfaces. *Proceedings of the …*, (2008), 201–210.
7. Johanson, B.Extending tuplespaces for coordination in interactive workspaces. *Journal of Systems and Software 69*, 3 (2004), 243–266.
8. Julià, C.F., Earnshaw, N., and Jordà, S.GestureAgents: An Agent-based Framework for Concurrent Multi-task Multi-user Interaction. *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, ACM (2013), 207–214.
9. Kaltenbrunner, M. and Bencina, R.reacTIVision: a computer-vision framework for table-based tangible interaction. *In Proc. TEI'07, ACM Press*, (2007), 69–74.
10. Kobayashi, N., Tokunaga, E., Kimura, H., Hirakawa, Y., Ayabe, M., and Nakajima, T.An Input Widget Framework for Multi-Modal and Multi-Device Environments. *Third IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS'05)*, (2005), 63–70.
11. Marquardt, N., Diaz-Marino, R., Boring, S., and Greenberg, S.The Proximity Toolkit: Prototyping Proxemic Interactions in Ubiquitous Computing Ecologies. *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, ACM (2011), 315–326.
12. Marquardt, N. and Greenberg, S.Distributed Physical Interfaces with Shared Phidgets. *Proceedings of the 1st international conference on Tangible and embedded interaction*, ACM (2007), 20.
13. Toole, C., Ullmer, B., Liu, K., Sankaran, R., Dell, C., and Branton, C.W.Toward Toolkit Support for Integration of Distributed Heterogeneous Resources for Tangible Interaction. *TEI'11 Works-in-Progress Workshop*, (2011), 85 – 90.
14. Toole Jr, C.Software Architectural Support for Tangible User Interfaces in Distributed, Heterogeneous Computing Environments. 2012. http://etd.lsu.edu/docs/available/etd-06052012-162857/.
15. Ullmer, B. and Ishii, H.Emerging frameworks for tangible user interfaces. *IBM Systems Journal 39*, 3 (2000), 915–931.
16. Ullmer, B., Sankaran, R., Liu, K., et al.Casier. *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction - TEI '11*, ACM Press (2011), 229.
17. Varela, G., Paz-Lopez, A., Becerra, J., and Duro, R.Decoupled Distributed User Interface Development Framework for Ambient Intelligence Systems. *DUI@ EICS*, (2013).
18. Wu, A., Mendenhall, S., and Jog, J.A nested APi structure to simplify cross-device communication. *Proceedings of the Sixth …*, (2012).

A Reality Checklist for Multi-Device Systems in the Wild?

Henrik Korsgaard
Department of Computer Science
Aarhus University
korsgaard@cs.au.dk

Clemens Nylandsted Klokmose
Department of Computer Science
Aarhus University
clemens@cs.au.dk

Steven Houben
ICRI-Cities & UCLIC
University College London
s.houben@ucl.ac.uk

Abstract
This position paper proposes the development of a reality checklist for multi-device systems in the wild. The checklist will help researchers evaluate designs, design ideas or design specifications for a system before it is deployed in the wild.

Author Keywords
Checklists, multi-device ecologies, in-the-wild

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

Introduction
In our research community we have a strong interest in understanding how technology can augment our everyday activities whether at home, at work, in the city, or in the local library. The technological advancement now allows us to explore systems or applications that span multiple devices both personal devices such as smartphones, tablets and laptops, and shared devices such as interactive wall or tabletop displays. We know from the literature that enticing people to interact with a system in a public or semi-public setting is challenging. It has been explored and discussed in work on tabletops [10, 7], public displays and interactive walls [8, 3], media architecture [4], and in combinations

of public displays and personal devices [5, 9]. In our own work with building and deploying multi-device systems (e.g. [11, 13]) we have experienced that the challenges multiply with the complexity of the ecology, particularly when involving peoples' personal devices, spatially distributed devices and/or networked infrastructure. Ironically, while most of us automatically reach for our smartphone as soon as there is a break in our busy life, getting us to take our phone out of the pocket to interact with a system in the wild is *hard* and people are reluctant to install applications on their device(s) on the spot [5]. For instance, Müller & Krüger [9] describe a system of 20 public advertisement displays installed in shops in a large German city running over the course of a year where people could use their phones to get rebate coupons. *Only 37 coupons were redeemed*. Another example is the CHI 2013 Interactive Schedule [13]. It was similarly based on a number of large (semi) public displays showing video previews for upcoming paper sessions. The system allowed for advanced interaction through the conference smartphone app over the local network. Papers featured on the large displays could be added to users' personal schedules and it was possible to create custom playlists for display on the large display. The WiFi at the conference center was (as always at large conferences) unstable, therefore obstructing the smartphone to public display interaction. *No one noticed, as no one got their phone out of their pockets at the displays*. In both cases there can be many explanations to why so few (or none) interacted with the system as intended by the designers.

With this position paper we propose the need for analytical tools to help us as researchers to systematically reality check designs, design specifications or design ideas for multi-device systems that are to be deployed in the wild, to catch our blind spots for potential interaction show stoppers before the system meets actual use.

Checklists and walkthroughs

Analytical tools in HCI and interaction design serve as means for evaluating a design or a specification for a design before involving real users. The *cognitive walkthrough* [12] is a well-known example. The method forces the analyst to break down and question each step involved in completing a task with the given user interface, to verify if a potential user will be able to select an appropriate action at each step. The cognitive walkthrough takes its starting point in an engaged user that is motivated to interact with the system. We are interested in analytical means to assess the situation that precedes and surrounds the actual interaction with the system. The *activity walkthrough* [2] is an extension of the cognitive walkthrough that emphasizes a *contextualization* of the use situation. This includes considering the activities the use situation is part of, the users' previous experiences with similar user interfaces, and previous experience with realizing their activity without the given user interface. Similarly, the *activity checklist* [6] is intended to elucidate the most important contextual factors of a user interface including a focus on user goals and social and physical aspects of the environment. We will in the following present a set of themes that we believe could form the initial foundation for a reality checklist for multi-device systems.

Outlining a Checklist

In the following we outline a checklists based on a preliminary analysis of our own cases and the related work. The checklist is based on themes divided into individual focus areas, which are accompanied by specific questions. A *no* to each of these questions should be seen as a red flag requiring further investigation.

Physical properties and visibility	
Distance	Is the system visible to potential users? Is the system discernible at the distance users first encounter it? Is the users' view to the system clear of obstructions?
Movement	Does the system align with the way people would move or sit in a space? Is the system aligned with the flow and existing layout of the physical space?
Orientation	Does the system align well with how people orient themselves in the physical space (e.g. opposite of doorways and openings)? Does the system utilize how the existing physical layout directs attention?
Quality	Does the system support the existing functions, configuration and purpose of the physical environment? (Should it?)
Material form	Does the system resemble what it is or indicate use through its form (e.g. interactive table/public display)? Is it easy to see what is part of the system and what is not?
Signs & instructions	Is the system accompanied with symbols or instructions explaining what it is? Are there symbols/text indicating connectivity (e.g. WiFi/Bluetooth) or basic functionalities?
Spatial distribution	Does the system relate to other physical objects close by? Are other artifacts in proximity of the system distinguishable from it (information displays, signs, installations)?

Understanding interaction	
Input & output	Is the interaction understandable (without other people using the system)? Is it easy to discern how to interact with the system at first encounter? Is it easy to understand the input/output relation? Does the system tease potential users with animations, examples, graphics, audio etc.?
Skill & time	Is interacting with the system easy for new-comers? Is it comprehensive how much time interaction will take? Does interaction 'end'?
Reward	Will interacting with the system reward potential users? Is there a clear outcome?
Social dynamics	Is interaction with the system aligned with social norms or existing social interactions? Is how the system expose or draw attention to users deliberately thought through?

Physical properties and visibility

The first step, which may seem banal, is to ensure that potential users are aware of the system. This entails looking at the context wherein the potential use will occur: How do people move and orient themselves in the physical space, distance, orientation, obstructions etc. (see e.g. [1])? How is the system is represented in the physical space, the material form, position, signs and spatial relationship? How should potential users might recognise elements of the system and draw on initial familiarity? After all, a display might just be a display in the crudest sense and input/out devices might be more or less hidden or the connection between them might be less obvious.

Understanding interaction

Once potential users have some awareness of the system, making sense of and identifying how one might interact with the system is the next step. Understanding how one interacts with a system might prompt actual interaction, understanding what one signs up for and/or participate in is key part of the decision process leading to initial interaction, and understanding some of the social dynamics of interacting with a table or pulling out a personal device is equally important [8, 3].

Motivation	
Goals & Incentives	Does the goal of using the system align well with potential goals of potential users? Are there incentives for walking up and using the system?
Investment & time	Is the required degree of personal investment for interaction acceptable for the users? Do the users have the time for it? Do the users want to get their smartphones out of their pockets for this?

Motivation
The motivation for approaching and initiating interaction is key in understanding how potential users negotiate whether they want to try, look and/or continue their activities. Extrinsic and intrinsic motivation both play a role. Systems within work settings or systems offering essential information offer extrinsic incentives and other potential users are driven by intrinsic motivation (playful, curious, fascination with technology etc.). Understanding motivation from a user's perspective will help reveal shortcomings or just a basic challenge in the design.

Technical obstructions
Lastly, when we have convinced a potential user, they need to be able to participate and use the system. For systems that utilise personal devices, some initiation and configuration is often needed, and in systems that relies on user information (login, profiles etc.) there is some overhead in setting up or logging into the system. Putting a personal device on an interactive table or logging into a public display also involves issue of trust and might make potential users reconsider interaction even when they have their smartphone out of their pockets.

Technical issues	
Connection & Compatibility	Is it quick to connect to the installation? Is the system compatible will all kinds of user devices? Is a password or a login required?
Configuration	Can users avoid installing anything on their personal devices? Can they avoid changing settings? Is setup simple and consisting only of very few steps?
Trust & Security	Does the system keep personal data safe? If interaction is logged, is it for the immediate benefit of the users? Can the user remove her profile/history/device from the system? Can others access users' data, if so is this made clear to the users?

Summing up
The themes and tables presented above outline a very preliminary reality checklist for multi-device systems when deploying these in the wild. We hope this proposal will spark discussion at the workshop and that the participants will help us move towards a more complete reality checklist.

REFERENCES

1. Till Ballendat, Nicolai Marquardt, and Saul Greenberg. 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. DOI: http://dx.doi.org/10.1145/1936652.1936676
2. Olav W Bertelsen. 2004. The activity walkthrough: an expert review method based on activity theory. In *Proceedings of the third Nordic conference on Human-computer interaction*. ACM, 251–254.
3. Harry Brignull and Yvonne Rogers. 2003. Enticing people to interact with large public displays in public

spaces. In *Proceedings of INTERACT*, Vol. 3. 17–24.

4. Martin Brynskov, Peter Dalsgaard, Tobias Ebsen, Jonas Fritsch, Kim Halskov, and Rune Nielsen. 2009. Staging urban interactions with media façades. In *Human-Computer Interaction–INTERACT 2009*. Springer, 154–167.

5. Nigel Davies, Adrian Friday, Peter Newman, Sarah Rutlidge, and Oliver Storz. 2009. Using bluetooth device names to support interaction in smart environments. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*. ACM, 151–164.

6. Victor Kaptelinin, Bonnie A Nardi, and Catriona Macaulay. 1999. Methods & tools: The activity checklist: a tool for representing the âĂIJspaceâĂİ of context. *interactions* 6, 4 (1999), 27–39.

7. Paul Marshall, Richard Morris, Yvonne Rogers, Stefan Kreitmayer, and Matt Davies. 2011. Rethinking 'Multi-user': An In-the-wild Study of How Groups Approach a Walk-up-and-use Tabletop Interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 3033–3042. DOI: http://dx.doi.org/10.1145/1978942.1979392

8. Jörg Müller, Florian Alt, Daniel Michelis, and Albrecht Schmidt. 2010. Requirements and design space for interactive public displays. In *Proceedings of the international conference on Multimedia*. ACM, 1285–1294.

9. Jörg Müller and Antonio Krüger. 2009. *MobiDiC: context adaptive digital signage with coupons*. Springer.

10. Rune Nielsen, Jonas Fritsch, Kim Halskov, and Martin Brynskov. 2009. Out of the Box: Exploring the Richness of Children's Use of an Interactive Table. In *Proceedings of the 8th International Conference on Interaction Design and Children (IDC '09)*. ACM, New York, NY, USA, 61–69. DOI: http://dx.doi.org/10.1145/1551788.1551800

11. Anna Maria Polli, Matthias Korn, and Clemens Nylandsted Klokmose. 2013. Local area artworks: collaborative art interpretation on-site. In *Proceedings of the 2013 ACM conference on pervasive and ubiquitous computing adjunct publication*. ACM, 79–82.

12. Peter G Polson, Clayton Lewis, John Rieman, and Cathleen Wharton. 1992. Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International Journal of man-machine studies* 36, 5 (1992), 741–773.

13. Arvind Satyanarayan, Daniel Strazzulla, Clemens N Klokmose, Michel Beaudouin-Lafon, and Wendy E Mackay. 2013. The CHI 2013 interactive schedule. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2987–2990.

Ad hoc adaptability in video-calling

Sean Rintel
Kenton O'Hara
Behnaz Rostami Yeganeh
Roman Rädle
Microsoft Research
Cambridge, UK
serintel@microsoft.com
keohar@microsoft.com
t-behros@microsoft.com
t-rorad@microsoft.com

Presented at the Cross-Surface '15 workshop, in conjunction with ACM ITS'15.

November 15, Funchal, Madeira, Portugal.

Abstract
In this position paper we explore ad hoc adaptability across devices in video-calling. We note the current difficulty of even simple combinations, discuss design issues, briefly report on a study of ad hoc screen mirroring, and note future directions.

Author Keywords
Video-mediated communication; multi-device interaction; adaptability; intelligibility; collaboration

ACM Classification Keywords
H.4.3 [Communications Applications]: Computer conferencing, teleconferencing, and videoconferencing

Introduction

Many opportunities for sharing resources in a conversation or meeting cannot be planned. As Suchman [10] noted, even when tasks themselves are planned, the achievement of those tasks is often a series of situated actions, the nature and needs of which change with the dual retrospective-prospective view that we take of successive actions. Many tasks are achieved through a combination of prepared resources which are brought to bear as situationally appropriate, discovering the need for unexpected resources, and developing resources within the task itself. Ad hoc sharing, then, is crucial to task fulfillment.

Despite a long history of research into the needs and methods of supporting ad hoc shared access to multiple information resources across domestic and institutional video-mediated communication [2,3,5], the dominant free and commercial systems on the market are still largely funneled and siloed. Ad hoc-ness is limited or in some cases impossible.

These limitations are the result of the 'one username per device per call' model that pervades most video-calling architectures. Even simple combinations are cumbersome. A collocated group will tend to cluster around a single endpoint, funneling participation into serial sequences of displays unless everyone locally joins the call, which is unlikely and unwieldy. Even a single person at one endpoint cannot expect all their devices to be aware of one another's' presence, state, and capabilities, let alone make use of those capabilities in parallel. As such opportunities for enabling rich conversations (e.g. Figure 1 and Figure 2) and work (e.g. Figure 3) are lost because the socio-technical transaction costs of combining the capabilities of devices is too high.

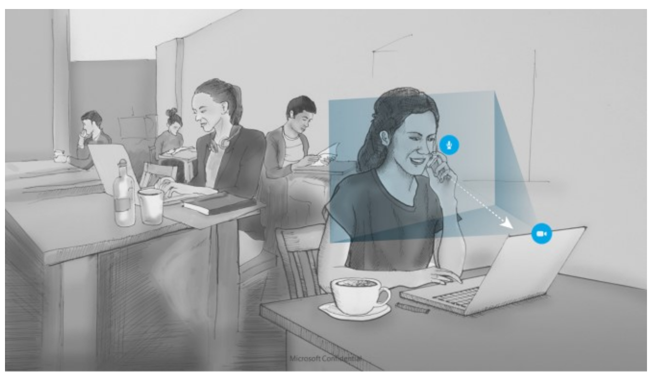

Figure 1: Personal combinations: Using a smartphone for audio and laptop for video in a noisy environment.

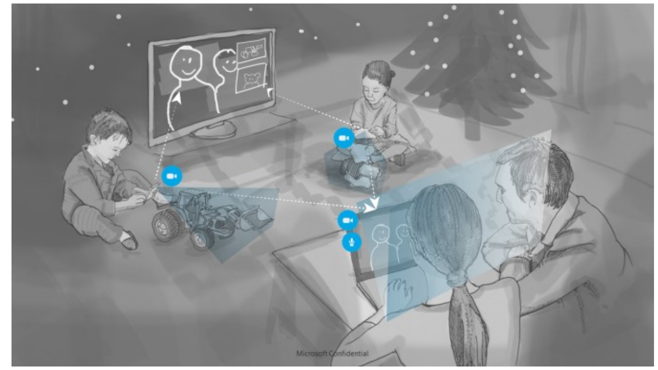

Figure 2: Domestic combinations: Using additional personal or shared devices to allow children to play in their own space while parents converse with grandparents from a laptop.

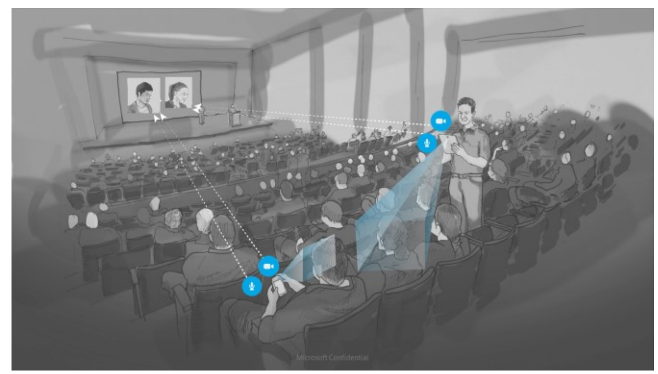

Figure 3: Work combinations: Using additional personal devices to be seen and heard while asking questions in an auditorium.

Designing for ad hoc adaptability

The issues of designing for the kind of ad hoc video-calling scenarios such as those above have been well articulated, especially in the work of Edwards and colleagues Speakeasy system [1] and, in the video-mediated communication field specifically, Neustaedter

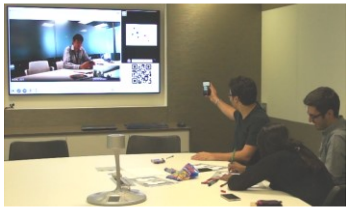

Figure 4: Pointing a smartphone at the QR code provided a seamful way for participants to understand that one participant would soon be mirroring.

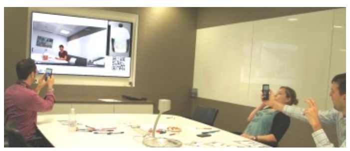

Figure 5: Local participants could acquire the QR code in parallel and then negotiate serial access to the displayed mirror.

and colleagues' Peek-a-Boo system [8]. Any 'recombinant' system needs to design for awareness of opportunities, intelligible system status, accountable control of status, recoverability and history, flexible and context aware combinations, feedback, simplicity of use and learning, and security and privacy.

Of these, accountability and intelligibility are the paramount drivers of designing for ad hoc adaptability. Accountability, in the ethnomethodological sense, refers to the way in which social order is achieved in the moment through treating social reasoning is observable and reportable in the actions of oneself and others. System accountability should operate in the same way, manifested through intelligibility, which is an ongoing awareness of the state of the system. The actions of the system and its users are thus holistically subject to practical moral reasoning. The other design principles noted above then feed into such reasoning. Further, as with the ad hoc inclusion of capabilities themselves, this moral reasoning will often need to be accomplished on the fly, even if there are pre-established and persistent policies for certain combinations.

As has been argued for home networking [4], we would also argue that intelligibility relies on providing users with access to both simple and detailed depictions of the connections and policies currently invoked in an ad hoc system. To a certain extent, then, we disagree with the 'it just works' market trend promoting the value of invisible, magical, seamless connection. Connections should certainly be easy to accomplish, but there is value, too, in visible 'seamfulness', such that users are not confused or surprised by any given connection. Seamfulness should not be a barrier to action – endless notifications and requests or convoluted specification of all the steps involved in connection – rather it should provide for expectable experiences.

We have explored some of these issues in a small-scale study of screen mirroring [9]. Screen mirroring is limited in many current setups. Access to screen mirroring tends to be restricted to the person driving the host computer. Swapping control is cumbersome and bringing to bear materials from a broader ecosystem of mobile devices and physical information surfaces even more so. The work-around of joining the video-call as an additional participant still involves social negotiations about taking the (displayed) floor [7], as generally only one device can mirror its screen at a time and often the mirroring takes over the majority of the display.

By adding a secondary window with a QR code that allowed all users to mirror their mobile screens – whether they showed material on the device or used the live camera – we found that mobility allowed users to contribute to the video call from their place in the room but also they were able to move around the room and even beyond. Individual work for preparing material to be was carried out in parallel to the overall discussion around the shared display. This enabled a more fluid interleaving of individual, subgroup, and full group sharing activities.

Most importantly, in terms of accountability and intelligibility, we found that participants sometimes negotiated among themselves as to who would take the floor. This could be just-in-time or organized serially (bidding for a place such as 'you go first and then I'll go'). Further, since acquiring the QR code on the camera was often a visible action requiring a clear

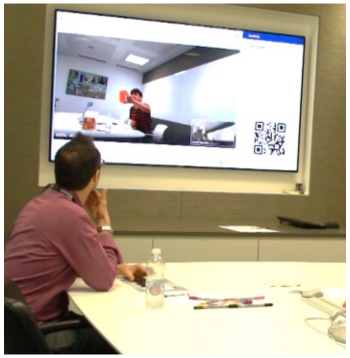

Figure 6: A local participant watches a remote participant acquire the QR code to establish mirroring, and thus knows to wait before establishing his own mirror.

pointing of the smartphone at the shared display, negotiations about upcoming likely sharing needs could be initiated by the same attention to embodied actions that we use to understand gestural onset [6]. However, this useful seamfulness worked best for local users. Remote users' bids for mirroring would sometimes be missed unless local users were watching the screen as the remote user acquired the QR code (Figure 6). Seamfulness, then, must be carefully designed take advantage of local conditions while also recognizing the asymmetry of remote access.

Conclusions

The increasing capabilities of web applications, web media stream standards such as ORTC, and IoT connection standards such as AllJoyn hold promise for the end of host-centric architectures and a bright future for realistic and robust cross-platform cross-device ad hoc adaptability of video-mediated communication systems. The design of such complex systems will need to need to foreground accountability and intelligibility to balance simplicity with seamfulness.

References

1. Edwards, W.K., Newman, M.W., Sedivy, J.Z., and Smith, T.F., 2009. Experiences with recombinant computing: Exploring ad hoc interoperability in evolving digital networks. *ACM Trans. Comput.-Hum. Interact. 16*, 1, 1-44. DOI= http://dx.doi.org/10.1145/1502800.1502803.
2. Finn, K., Sellen, A., and Wilbur, S., 1997. Video-mediated communication Erlbaum, Mahwah, NJ.
3. Gaver, W.W., Sellen, A., Heath, C., and Luff, P. One is not enough: multiple views in a media space. In *Proc. Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems 1993*, ACM (1993), 335-341.
4. Grinter, R.E., Edwards, W.K., Chetty, M., Poole, E.S., Sung, J.-Y., Yang, J., Crabtree, A., Tolmie, P., Rodden, T., Greenhalgh, C., and Benford, S., 2009. The ins and outs of home networking: The case for useful and usable domestic networking. *ACM Trans. Comput.-Hum. Interact. 16*, 2, 1-28. DOI= http://dx.doi.org/10.1145/1534903.1534905.
5. Harrison, S., 2009. Media space 20+ years of mediated life Springer-Verlag, London.
6. Mondada, L., 2007. Multimodal resources for turn-taking: pointing and the emergence of possible next speakers. *Discourse Studies 9*, 2 (April 1, 2007), 194-225. DOI= http://dx.doi.org/10.1177/1461445607075346.
7. Müller-Tomfelde, C. and O'Hara, K., 2010. Horizontal Surfaces in Media Space Assemblies. In *Tabletops: Horizontal Interactive Displays*, C. MÜLLER-TOMFELDE Ed. Springer, London, 435-456.
8. Neustaedter, C. and Judge, T.K. Peek-A-Boo: the design of a mobile family media space. In *Proc. Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing - Adjunct 2010*, ACM (2010), 449-450.
9. Sørensen, H., O'Hara, K., Gosset, P., and Kjeldskov, J., 2015. Wireless Smartphone Mirroring in Video Calls. In *Human-Computer Interaction – INTERACT 2015*, J. ABASCAL, S. BARBOSA, M. FETTER, T. GROSS, P. PALANQUE and M. WINCKLER Eds. Springer International Publishing, 410-417. DOI= http://dx.doi.org/10.1007/978-3-319-22698-9_28.
10. Suchman, L.A., 1987. *Plans and situated actions: the problem of human-machine communication*. Cambridge University Press, Cambridge, MA.

Challenges in Managing Multi-Device Notifications

Abhinav Mehrotra
University of Birmingham
University College London
United Kingdom
a.mehrotra@cs.bham.ac.uk

Jo Vermeulen
University of Birmingham
United Kingdom
j.vermeulen@cs.bham.ac.uk

Robert Hendley
University of Birmingham
United Kingdom
r.j.hendley@cs.bham.ac.uk

Mirco Musolesi
University College London
United Kingdom
m.musolesi@ucl.ac.uk

Presented to the *Cross-Surface '15* workshop, in conjunction with ACM *ITS'15*. November 15, Funchal, Madeira, Portugal.

Abstract

Many applications and online services on our mobile phones notify us about messages, emails, or status updates. These notifications can be disruptive. People currently use multiple mobile and wearable devices that can run the same application or service simultaneously. This aggravates the issue of disruptive notifications. For example, a new email notification can trigger vibrations or sounds on a tablet, laptop, smartphone and smartwatch at the same time. Additionally, notifications can be sent from one device to another (e.g., an internet-connected thermostat that notifies a user on their smartphone). In this position paper, we discuss the key challenge of handling and managing multi-device notifications. We outline open issues in addressing this problem and discuss opportunities for future work.

Author Keywords

Notifications, Interruptibility, Mobile Sensing, Multi-Device Interaction, Wearables

ACM Classification Keywords

H.5.m. [Information Interfaces and Presentation (e.g. HCI)]

Figure 1: Mark is chatting with Susan via an instant messaging application. When Susan replies, he receives multiple notifications on his smartwatch, laptop and smartphone (sketch by Lindsay MacDonald).

Introduction

People currently tend to own and use multiple personal devices, including smartphones, tablets, laptops, wearables and IoTs. While the perpetual availability of information through these devices and increased connectivity is generally considered to be beneficial, people also get interrupted through constant notifications on their devices. One critical problem reported by early adopters of smartwatches is being constantly interrupted by notifications. Despite this, early adopters also report the ability to unobtrusively receive information in social situations as an advantage of wearing a smartwatch [5]. Another study found that some smartwatch compulsively check their watches and even check their bare wrists when they were not wearing their devices [2]. Some reported regularly feeling a phantom buzz, notifying them of an imagined incoming message.

Another issue is that our personal devices tend not to be aware of each other. Notifications often trigger vibrations or sounds on all of our devices at the same time. In Figure 1, Mark is discussing a problem with his colleague Susan via an instant messaging application on his laptop. Mark has the same instant messaging application installed on both his smartwatch and smartphone. When Susan replies to Mark's message, notifications arrive on all three devices to alert Mark for Susan's incoming message.

This demonstrates the need for more intelligent multi-device interruption management. If applications are aware of a user's different devices, they could determine the best device to use to alert the user. In this case, the notification could just have appeared on Mark's laptop. If Mark was walking around the office to get a coffee, however, the notification could have been delivered to his smartwatch instead.

In order to achieve this kind of flexibility, an intelligent notification approach should not only infer the user's receptivity to notifications in a particular context, but also learn how the user interacts with notifications on each of their devices in that context. In other words, multi-device applications should consider *when* and *whether* to deliver a notification and additionally *where* (i.e., on which device) this notification should be delivered. Multi-device interruption management can benefit applications that trigger push notifications by reducing interruptions from multiple simultaneous notifications.

Related Work

Previous studies have found that initiating interactions at inopportune moments can cause interruptions [10]. These interruptions can adversely affect task completion time [6], error rate [4], and can influence people's affective state [3]. Additionally, people may feel anxious about missing out when they cannot check their devices for incoming notifications [11, 9]. Studies suggest that some people tend to receive hundreds of notifications per day [8].

Researchers have proposed different mechanisms for managing interruptibility. Adamczyk and Bailey [1] propose a method to predict interruption timings based on a user's cognitive load during the task execution. the location, time of day, activity) to infer opportune moments to deliver notifications. Mehrotra et al. [8] show that machine learning classifiers lead to a more accurate prediction of a user's interruptibility when trained with both the content of a notification and the user's context. On the other hand, PrefMiner [7] proposes a novel interruptibility management solution that learns users' preferences for receiving notifications based on automatic extraction of rules by mining their interaction with mobile phones. However, all of these studies focus on interruptions caused by a single device and do not consider multi-device interruptions.

Challenges

In this section, we discuss a number of research challenges for multi-device interruption management.

Deciding Where to Interrupt?

An important question for intelligent multi-device notification management is to decide *where*, *when*, *whether* and *how* to interrupt the user. A naïve approach would be to detect when people are actively using one of their devices (e.g., by monitoring input events) and direct all notifications only to that particular device. However, there are other aspects that are important to consider in determining the most suitable device to use. For example, some devices can be more suitable to use in a certain context than others (e.g., while driving).

Inferring Engagement with a Device
Different techniques can be used to determine whether people are actively using their device. Input events can be monitored to infer how actively engaged the user is with a device (e.g., key presses on a laptop).

Smartphones and smartwatches tend to turn off their displays after a certain period of inactivity to optimise battery usage. The status of the display (on or off) can be tracked and used as a simple measure of engagement.

Gaze tracking can be an rich, albeit more complex and resource-intensive, way of measuring engagement by tracking whether the user is actually looking at the device. Some Android smartphones already feature built-in gaze trackers. One application of gaze tracking is to verify that users have seen a certain notification without requiring them to interact with it first (e.g., clicking on an email to mark it as unread).

Predicting Which Device Notifications Will Be Handled On
Daily use of personal devices together with the increasing popularity of wearables and the presence of connected embedded sensors in many artefacts and the fabric of cities themselves (i.e., the "Internet of Things") allows us to extract and model some inherent patterns of human behaviour. More specifically, user behaviour in specific sensed contexts observed over a period of time can be used to build predictive models. In order to predict the device on which the user will handle a notification in a given context, an application could learn the patterns of the user's receptivity to information on different devices in different situations.

Context prediction has been investigated in the past for forecasting future locations of users [13], communication patterns [12], interruptibility [8] and preferences [7]. Similar techniques can be used to construct a machine learning model that learns the patterns of the user's interaction with notifications on different devices in different contexts. The model would gradually start making sensible predictions about the device on which the user will handle a certain notification in the current context. A key challenge is to be able to train the system over multiple devices. Certain devices might be used rarely by a user in a certain context and this can make the learning task very challenging. There is a fundamental problem related to bootstrapping the learning components. A possible approach is to adopt models extracted over multiple users and refine these, for example, using Bayesian approaches.

Privacy and Shared Devices

People might be using devices that can be seen by others (e.g., notifications appearing while giving a presentation). In order to maintain privacy, personal notifications might have to be suppressed on a particular device, even though it is the active one. Additionally, some devices such as tablets might be shared between family members.

Providing User Control

Another important challenge is providing a balance between automatic notification routing and end-user control over the dispatch of notifications to certain devices in a particular context. There are limitations to what can be correctly sensed, which means notification routing algorithms will inevitably make inappropriate judgements. Users could configure rules to route certain types of notifications to certain devices, or the application could defer to the user's judgement when it cannot infer which device is the most appropriate. An important consideration is that asking the user where to deliver notifications can again cause interruptions.

Conclusion

Cross-device interaction enables people to use different devices together and have access to information and complete tasks on several devices. A key challenge in this is interruption overload. All these devices and applications are constantly competing for the user's attention, with several devices that buzz or beep at the same time. In this paper, we have discussed opportunities and challenges for intelligent multi-device notifications. In order to address the issue of multi-device interruptions, applications should be able infer the user's engagement with different devices and learn the patterns of the user's receptivity to information on different devices in different contexts. A key issue in designing multi-device notification management is training the machine learning components, given the fact that particular interactions with certain types of notifications may rarely occur.

References

[1] Adamczyk, P. D., and Bailey, B. P. If not now, when?: the effects of interruption at different moments within task execution. In *CHI'04* (2004).

[2] Andre Spicer. Apple Watch and the Internet of me, Mar. 2015. http://www.straitstimes.com/opinion/apple-watch-and-the-internet-of-me.

[3] Bailey, B. P., and Konstan, J. A. On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Computers in Human Behavior 22*, 4 (2006), 685–708.

[4] Bailey, B. P., Konstan, J. A., and Carlis, J. V. Measuring the effects of interruptions on task performance in the user interface. In *SMC'00* (2000).

[5] Cecchinato, M. E., Cox, A. L., and Bird, J. Smartwatches: The good, the bad and the ugly? In *CHI EA '15* (2015), 2133–2138.

[6] Czerwinski, M., Cutrell, E., and Horvitz, E. Instant messaging and interruption: Influence of task type on performance. In *OZCHI'00* (2000).

[7] Mehrotra, A., Hendley, R., and Musolesi, M. PrefMiner: Mining User's Preferences for Intelligent Mobile Notification Management. In *UbiComp'16* (September 2016).

[8] Mehrotra, A., Musolesi, M., Hendley, R., and Pejovic, V. Designing Content-driven Intelligent Notification Mechanisms for Mobile Applications. In *UbiComp'15* (2015).

[9] Mehrotra, A., Pejovic, V., Vermeulen, J., Hendley, R., and Musolesi, M. My Phone and Me: Understanding People's Receptivity to Mobile Notifications. In *CHI'16* (April 2016).

[10] Miyata, Y., and Norman, D. A. Psychological issues in support of multiple activities. *User centered system design: New perspectives on human-computer interaction*

(1986), 265–284.
[11] Oulasvirta, A., Rattenbury, T., Ma, L., and Raita, E. Habits make smartphone use more pervasive. *Personal and Ubiquitous Computing 16*, 1 (2012), 105–114.
[12] Pielot, M. Large-scale evaluation of call-availability prediction. In *UbiComp'14* (2014).
[13] Scellato, S., Musolesi, M., Mascolo, C., Latora, V., and A., C. Nextplace: A spatio-temporal prediction framework for pervasive systems. In *Pervasive'11* (2011).