An Activity-Centric Approach to Configuration Work in Distributed Interaction

Steven Houben

PhD Thesis in Human-Computer Interaction

PIT LAB

Pervasive Interaction Technology Laboratory
IT University of Copenhagen

Author:
Steven Houben
An Activity-Centric Approach to Configuration Work in Distributed Interaction
PhD Thesis In Human-Computer Interaction,

Institute:
IT University of Copenhagen
Software Development Group
Pervasive Interaction Technology Laboratory

Supervisor:
Professor Jakob E. Bardram

Committee:
Professor Susanne Bødker, *Aarhus University*
Professor Aaron Quigley, *University of St Andrews*
Associate Professor Thomas Pederson, *IT University of Copenhagen*

This dissertation is submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Ph.D.) at IT University of Copenhagen.

Submitted, November 2014 and *Defended, March 2015*

“Since the real world of experience will never fail to afford material and reality to our ethical investigations, nothing will be less needful than to take refuge in negative conceptions void of content, and then somehow or other make even ourselves believe that we are saying something when we speak with lifted eyebrows of absolutes, infinites, supersensibles and whatever other mere negations of this sort there may be, instead of which it would be shorter to say at once Nephelokokkygia: we shall not require to serve up covered empty dishes of this kind.**”**

– Arthur Schopenhauer [162]

Abstract

The widespread introduction of new types of computing devices, such as smartphones, tablet computers, large interactive displays or even wearable devices, has led to setups in which users are interacting with a rich ecology of devices. These new device ecologies have the potential to introduce a whole new set of cross-device and cross-user interactions as well as to support seamless distributed workspaces that facilitate coordination and communication with other users. Because of the distributed nature of this paradigm, there is an intrinsic difficulty and overhead in managing and using these kind of complex device ecologies, which I refer to as configuration work. It is the effort required to set up, manage, communicate, understand and use information, applications and services that are distributed over all devices in use and people involved. Because current devices and their containing software are still document- and application-centric, they fail to capture and support the rich activities and context in which they are being used. This leaves users without a stable concept for cross-device information management, forcing them to perform a large amount of manual configuration work.

In this dissertation, I explore an activity-centric approach to configuration work in distributed interaction. The central goal of this dissertation is to *develop and apply concepts and ideas from Activity-Centric Computing to distributed interaction*. Using the triangulation approach, I explore these concepts on a *conceptual*, *empirical* and *technological* level and present a framework and use cases for designing activity-centric configurations in multi-device information systems. The dissertation presents two major contributions:

First, I introduce the term *configuration work* as an abstract analytical unit that describes and captures the problems and challenges of distributed interaction. Using both empirical data and related work, I argue that configuration work is composed of: curation work, task resumption lag, mobility work, physical handling and articulation work. Using configuration work as a problem description, I operationalize Activity Theory and Activity-Centric Computing to mitigate and reduce configuration work in distributed interaction. By allowing

users to interact with computational representations of their real-world activities, creating complex multi-user device ecologies and switching between cross-device information configurations will be more efficient, more effective and provide better support for users' mental model about a multi-user and multi-device environment. Using *activity configuration* as a central concept, I introduce a framework that describes how digital representations of human activity can be distributed, fragmented and used across multiple devices and users.

Second, I present a technical infrastructure and four applications that apply the concepts of activity configuration. The infrastructure is a general purpose platform for the design, development and deployment of distributed activity-centric systems. The infrastructure simplifies the development of activity-centric systems as it presents complex distributed computing processes and services into high level *activity system* abstractions. Using this infrastructure and conceptual framework, I describe four fully working applications that explore multi-device interactions in two specific domains: office work and hospital work. The systems are evaluated and tested with end-users in a number of lab and field studies.

Full Publication List

Concepts, figures, systems and studies presented in this dissertation have appeared previously or will appear in the following publications. Some passages and figures in this dissertation have been quoted verbatim from the following publications:

Preprint Manuscripts

1. **Steven Houben** and Jakob E. Bardram. Activity-Centric Support for the Distributed Configuration Problem in Clinical Work.
   To appear in the **iCarenet Book, Springer**.
   (Chapter – 30 pages)
2. Jakob E. Bardram, Steven Jeuris and **Steven Houben**. Activity-Based Computing: Computational Management of Activities Reflecting Human Intention.
   To appear in special issue of the **AI Magazine** on PAIR and activity context-aware system architectures.
   (Full paper – 9 pages)

Published Manuscripts

1. **Steven Houben** and Nicolai Marquardt. WatchConnect: A Toolkit for Prototyping Smartwatch-Centric Cross-Device Applications.
   In Proceedings of the 2015 SIGCHI Conference on Human Factors in Computing Systems **(CHI'15)**. ACM, 2015.
   (Full paper – 10 pages)
2. **Steven Houben**, Mads Frost and Jakob E. Bardram. Collaborative Affordances of Hybrid Technologies in Medical Work.
   In Proceedings of the 2015 Conference on Computer-Supported Cooperative Work and Social Computing **(CSCW'15)**. ACM, 2015.
   (Full paper – 13 pages)

3. **Steven Houben**, Paolo Tell and Jakob E. Bardram. ActivitySpace: Managing Device Ecologies in an Activity-Centric Configuration Space.
In Proceedings of the 2014 ACM international conference on Interactive tabletops and surfaces **(ITS'14)**. ACM, 2014.
(Full paper – 10 pages)
4. **Steven Houben**, Mads Frost and Jakob E. Bardram. HyPR Device: Mobile Support for Hybrid Patient Records.
In Proceedings of the 2014 ACM international conference on Interactive tabletops and surfaces **(ITS'14)**. ACM, 2014.
(Full paper – 10 pages)
5. Steven Jeuris, **Steven Houben** and Jakob E. Bardram. Laevo: A Temporal Desktop Interface for Integrated Knowledge Work.
In Proceedings of the 2014 ACM Symposium on User Interface Software and Technology (**UIST'14**), ACM, 2014.
(Full paper – 10 pages)
6. **Steven Houben**, Søren Nielsen, Morten Esbensen, and Jakob E. Bardram. NooSphere: an activity-centric infrastructure for distributed interaction.
In Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia **(MUM '13)**. ACM, 2013.
(Full paper – 10 pages)
7. **Steven Houben**, Jakob E. Bardram, Jo Vermeulen, Kris Luyten, and Karin Coninx. Activity-centric support for ad hoc knowledge work: a case study of co-activity manager.
In Proceedings of the 2013 SIGCHI Conference on Human Factors in Computing Systems **(CHI'13)**. ACM, 2013.
(Full paper – 10 pages)
8. **Steven Houben** and Jakob E. Bardram. ActivityDesk: multi-device configuration work using an interactive desk.
In CHI '13 Extended Abstracts on Human Factors in Computing Systems **(CHI EA'13)**. ACM, 2013.
(Extended abstract – 6 pages)

9. **Steven Houben**, Jo Vermeulen, Kris Luyten, and Karin Coninx. Co-activity manager: integrating activity-based collaboration into the desktop interface. *In Proceedings of the 2012 International Working Conference on Advanced Visual Interfaces* **(AVI'12)**. ACM, 2012.
   (Short paper – 4 pages)
10. **Steven Houben**, Morten Esbensen, and Jakob E. Bardram. A situated model and architecture for distributed activity-based computing. *In Proceedings of the 2012 International Workshop on Model-based Interactive Ubiquitous Systems* **(MODIQUITOUS'12)**. Ceur, 2012.
    (Short paper – 5 pages)
11. Jakob E. Bardram, Sofiane Gueddana, **Steven Houben**, and Søren Nielsen. ReticularSpaces: activity-based computing support for physically distributed and collaborative smart spaces. *In Proceedings of the 2012 SIGCHI Conference on Human Factors in Computing Systems* **(CHI'12)**. ACM, 2012.
    (Full paper – 10 pages)

Included Papers

Part II of this dissertation consists of a collection of 7 published peer reviewed conference papers. For all papers, I am the major contributor as I performed all research, conducted the evaluations and studies and was lead author on the manuscripts. Author statements are included as appendices.

- **C1** **Steven Houben**, Jakob E. Bardram, Jo Vermeulen, Kris Luyten, and Karin Coninx. Activity-centric support for ad hoc knowledge work: a case study of co-activity manager.
  In Proceedings of the 2013 SIGCHI Conference on Human Factors in Computing Systems **(CHI'13)**.
  (Full paper – 10 pages)
- **C2** **Steven Houben**, Søren Nielsen, Morten Esbensen, and Jakob E. Bardram. NooSphere: an activity-centric infrastructure for distributed interaction.
  In Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia **(MUM '13)**.
  (Full paper – 10 pages)
- **C3** **Steven Houben** and Jakob E. Bardram. ActivityDesk: multi-device configuration work using an interactive desk.
  In CHI '13 Extended Abstracts on Human Factors in Computing Systems **(CHI EA'13)**.
  (Extended abstract – 6 pages)
- **C4** **Steven Houben**, Paolo Tell and Jakob E. Bardram. ActivitySpace: Managing Device Ecologies in an Activity-Centric Configuration Space.
  In Proceedings of the 2014 ACM international conference on Interactive tabletops and surfaces **(ITS'14)**.
  (Full paper – 10 pages)

C5 **Steven Houben** and Nicolai Marquardt. WatchConnect: A Toolkit for Prototyping Smartwatch-Centric Cross-Device Applications.
In Proceedings of the 2015 SIGCHI Conference on Human Factors in Computing Systems **(CHI'15)**.
(Full paper – 10 pages)

C6 **Steven Houben**, Mads Frost and Jakob E. Bardram. HyPR Device: Mobile Support for Hybrid Patient Records.
In Proceedings of the 2014 ACM international conference on Interactive tabletops and surfaces **(ITS'14)**.
(Full paper – 10 pages)

C7 **Steven Houben**, Mads Frost and Jakob E. Bardram. Collaborative Affordances of Hybrid Technologies in Medical Work.
In Proceedings of the 2015 Conference on Computer-Supported Cooperative Work and Social Computing **(CSCW'15)**.
(Full paper – 13 pages)

Acknowledgments

This research and dissertation would have been impossible without the encouragement and support of others.

First, I would like to thank my advisor *Jakob E. Bardram*. You have had a profound influence on my work and I am very grateful for your guidance, supervision and encouragement throughout the three years I worked on my PhD. I have been very fortunate to be able to do research in the extraordinary conditions that you created, and I realize that without your support many aspects of my work would have been impossible to complete in such a short time. Thank you!

Thanks to my committee members *Susanne Bødker*, *Aaron Quigley* and *Thomas Pederson* for the detailed feedback, provocative and interesting questions, and suggestions for future work. It was an absolute privilege to have you as the assessment committee of my dissertation and defense presentation.

I would like to thank all the past and present members of the Pervasive Interaction Technology (PIT) laboratory. Working in the PIT lab was an amazing experience and privilege and I cannot describe how much I have learned from everyone in the lab. Thanks to *Sebastian Buttrich* for all the technical assistance, music and moral support in times of stress. I would also like to thank *Afsaneh Doryab*, *Siemen Baader* and *Juan David Hincapié-Ramos* for the amazing welcome when I started in the lab and for their continuing advice and support. Special thanks to *Aurelien Tabard* for your amazing advice and support; it was a privilege to share an office with you! I would also like to thank *Morten Esbensen* and *Mads Frost* for teaching me the 'Danish way' and for the great discussions. Special thanks also to *Steven Jeuris* and *Paolo Tell* for the great ABC discussions and collaboration; I enjoyed sharing the lab island with you!

Thanks to *John Paulin Hansen* for your feedback on both my research and early version of this thesis. I would also like to thank *Shahram Jalalinia* for being my iCareNet travel buddy, and for your technical help and feedback on my work. Thanks to all the people I directly or indirectly worked with in the lab – *Søren Nielsen*, *Sofiane Gueddana*, *Tim Hankins*, *Mathias Schmidth*, *Dominik*

Grondziowsk, *Lars Yndal Sørensen*, *Thomas Snaidero*, *Pelle Krøgholt*, *Martin Poulsen*, *Simon Terkildsen*, *Jens Zeilund* and all the other people in the lab. I would also like to thank *Anna Vallgårda* and *Lars Toft Jacobsen* of the IxD Laboratory for letting me use their lab and for their technical support.

Some early work in this dissertation was done partly during a research internship in the HCI lab at Hasselt University. Thanks to *Karin Coninx* and *Kris Luyten* for providing me with an amazing opportunity to join your lab as an intern. I am very grateful for the excellent training I received during my research internship with you. Special thanks to *Jo Vermeulen* for the amazing supervision and collaboration. I was very fortunate to work with *Nicolai Marquardt* at the Interaction Centre at University College London (UCLIC). Thank you very much for hosting me and providing me with excellent advice and technical support for an amazing project. I am very much looking forward to continuing our collaboration. Also, thanks to all the iCareNet fellows - *Christian Weichel*, *Frank Bolton* and *Illapha Cuba Gyllensten* for the amazing discussions on 'science'.

A special thanks to my friends and family for their love and support during my studies. Most importantly I would like to thank my awesome girlfriend ***Evi Vlassaks***. None of this would have been remotely possible without your endless patience, support and love. Thank you!

Steven Houben, November 2014

Research Acknowledgments

During my PhD research, I have collaborated with a number of people. I am the major contributor of all research included in this dissertation as I performed the studies, built the software and wrote the papers. However, parts of the work presented in this dissertation and containing papers have been done in collaboration with other researchers and students, who have supported my research:

All work in this dissertation was done under supervision of **Jakob E. Bardram**. **Jo Vermeulen**, **Karin Coninx** and **Kris Luyten** were my supervision team during the development and implementation of co-Activity Manager. They collaborated on the writing of the manuscript and analysis of the results of the study. The development of the NooSphere infrastructure was done with the help of **Søren Nielsen**, who implemented parts of the cloud platform. **Morten Esbensen** contributed to the conceptual framing of the architecture of the infrastructure. I collaborated with **Paolo Tell** on the conceptualization and the writing of the ActivitySpace paper. The Hybrid Patient Record project was done in collaboration with **Mads Frost**, who helped conduct the initial workshops and the clinical simulation. **Mathias Schmidt**, **Thomas Snaidero** and **Vaidas Sirtautas** were students that helped to implement and test the second version of the Hybrid Patient Record (HyPR) device. I collaborated with **Nicolai Marquardt** on the design and ideas behind the WatchConnected toolkit and the writing of the manuscript.

All the work presented in this dissertation was supported by the EU Marie Curie Network iCareNet grant number 264738, IT University of Copenhagen and the Pervasive Interaction Technology Laboratory (PIT Lab).

Contents

List of Figures

List of Tables

1. Introduction

With the widespread introduction of new types of computing devices, such as smartphones, tablet computers, large interactive displays or even wearable devices such as smartwatches, the vision of Mark Weiser's Ubiquitous Computing [185] is increasingly becoming a reality. Setups in which users are interacting with a rich ecology of devices are becoming more common outside of research labs. This, together with the move of information from local storage to the ubiquitous accessible cloud, has shifted the focus from individual devices to an ecology of interconnected devices (such as seen in Figure 1.1) that are portals into a shared information space. These new device ecologies have the potential to introduce a whole new set of cross-device and cross-user interactions as well as support seamless distributed workspaces that facilitate coordination and communication with other users. More general, *distributed interaction* refers to a new paradigm in which interaction with a computer system is dynamically distributed over multiple (i) *people*, (ii) *devices* and (iii) *environments*. These three elements change over time and are in a dynamic and ad hoc way related to the task or activity people are doing. Information is no longer tied to one specific personal device but devices have become mediators to the ubiquitous shared information space supported by the internet.

Figure 1.1: The range of devices that are currently widespread among people.

Despite the important advantages of *distributed interaction*, prior studies [51, 136, 156] have highlighted that it also introduces important problems. Users are often overloaded with complex workflows that require them to move information or applications between a set of heterogeneous devices owned by different users. Despite the numerous available tools and systems, moving information between devices is often still a painful process that forces users to invent workarounds, spend a considerable amount of time setting up the devices and negotiate the tools with other users. Because of the distributed nature of this paradigm, there is an intrinsic difficulty and overhead in managing and using complex device ecologies, which I refer to as *configuration work*. It is the effort required to *set up*, *manage*, *communicate*, *understand* and *use* information, applications and services that are distributed over all devices in use and people involved. At its core, configuration work is related to the fact that devices are designed and perceived as individual entities and not as part of the complex ecology they are often used in. Moreover, because current devices and their containing software are still document- and application-centric, they fail to capture and support the activities and contexts in which they are being used. This leaves users without a *stable concept* for cross-device information management, forcing them to perform a large amount of manual *configuration work*.

Although configuration work in distributed interaction is a generic problem, the focus in this thesis revolves around two important and well documented use cases: (i) personal and collaborative work in an office setting, and (ii) mobile and collaborative clinical work in hospital wards.

Consider first the case of a group of interdisciplinary knowledge workers as seen in Figure 1.2. On average, these types of users own about three to six computing devices [51, 85, 156], including desktop computers, laptops, smartphones, tablets, e-readers and nowadays even smartwatches. Most of these devices are no longer dedicated to just one specialized task but have become multifunctional devices providing ubiquitous access to different sources of information. Together with traditional all-purpose computers, these new types of devices form a device ecology that provides access to an overlapping information space. Although knowledge work is often perceived as individual, it is in fact highly collaborative and includes many people and devices [51, 156]. Moreover, knowledge workers often collaborate on several simultaneous projects in partially overlapping subgroups (as indicated in Figure 1.2) that contain a different configuration of people and devices involved. Because of this dynamic and ad hoc characteristic of knowledge work, resources are often shared and used by several people on different devices. For these projects, the team members continuously collaborate on shared parallel work, which requires some form of coordination between individually performed work (e.g., sharing of files or awareness of each other's updates on those files). Although most devices, such as seen in Figure 1.1 and 1.2, have network connectivity, it is mostly still a tedious process to meaningfully connect devices and effectively exchange information. Previous studies [51, 136, 156] show that users encounter a number of fundamental problems when interacting with these distributed device ecologies, such as lack of transparency, control, intelligibility and context. Additionally, studies [10, 50, 74] have shown that knowledge workers often implicitly organize information, applications and other resources in thematically higher-level constructions, that reflect the work they are trying to do. Although these higher-level *tasks* are often individual by construction, their structure and properties influence other people that are part of the work contained inside these tasks. Because most current systems are *task-agnostic*, sharing information within these working contexts is very hard and requires users to do substantial configuration work.

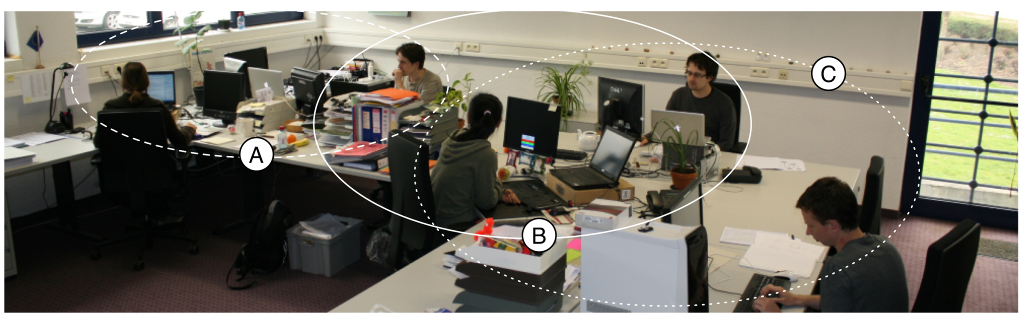


Figure 1.2: A group of knowledge workers that collaborate within a number of overlapping activities.

As a second example, consider the case of a clinical work in hospital wards. The workflow of clinicians in a patient ward can be described as *nomadic*. In addition to sitting in an office or other fixed locations for managing, archiving and preparing patient information, clinicians also roam through the hospital while doing their work [19]. This work typically includes collaborations with many people and requires the usage of physical tools and computing devices that are spread over multiple locations. Clinicians often move from one location to another while interacting with both mobile and stationary tools and devices, such as desktop computers (Figure 1.3A), large interactive whiteboards (Figure 1.3B) and mobile devices (Figure 1.3C). Despite the ubiquity and importance of these tools for the information flow in the hospital, most of these tools are not designed, tailored or suited for this nomadic use, resulting in a mismatch between the functionality of the system and the actual work done by the clinicians. Fundamentally, all these different systems and devices are intrinsically disconnected from each other, forcing clinicians to manually *reconfigure* the active work setting according to the situation. This results in work interruptions, information fragmentation and a disconnected workflow across several resources. Furthermore, current systems provide little support for the ongoing collaborative context, mobile situations or specific role of the device, thus, increasing the amount of configuration work.

Figure 1.3: Devices used at a medical ward in hospitals.

1.1 Research Context

The research context of this dissertation is situated within the field of *Human-Computer Interaction* (HCI), *"a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them"* [68]. The work presented in this thesis falls within the area of *Ubiquitous Computing*, a concept and discipline in Computer Science and HCI that focuses on the design and study of interactive systems that are comprised of multiple devices or sensors that integrate into the everyday environment of users as they "*weave themselves into the fabric of everyday life until they are indistinguishable from it*" [185]. More specifically, I explore how problems in *distributed interaction*, in which interactive systems are spread across multiple people and devices, can be addressed using *Activity-Centric Computing*.

The theoretical background of this work lies in Activity Theory, a socio-cultural framework that describes human activity as a relation among the subject (S) (human or group that acts in the world), object (O) (which is acted upon and motivates the activity) and the community (C) (or social strata in which the activity is engaged) [112]. Activity Theory provides us with a broad framework and vocabulary to reinterpret the core components of human-computer interaction in a distributed interactive setting. A number of previous work has explored the application of Activity Theory to system design [18, 113, 177], conceptualization of human-computer interaction [36, 37] and analysis of empirical data [14] leading to the Activity-Centric Computing paradigm. The work presented in this thesis builds on this previous work and extends Activity-Centric Computing with new concepts, insights and developments that are primarily situated at the *system design*, *concept* and *empirical study* level.

1.2 Research Questions

The central problem that is addressed by this dissertation, is mitigating *configuration work in distributed interaction*. I describe distributed interaction as interaction with a computer system that is dynamically distributed over multiple *people*, *devices* and *environments*. I define configuration work as the effort required to *control*, *manage*, *understand*, *communicate* and *use* information, applications and services that are distributed over all used devices and people. The central research question addressed in this dissertation is:

> *How can activity-centric computing be used to reduce configuration work in distributed interaction?*

This question can be subdivided into three core research questions:

Q1 Configuration Work - How do users *set up*, *manage*, *communicate*, *understand* and *use* information, applications and services that are distributed over all devices in use and people involved? What constitutes configuration work? What are the properties and processes of configuration work?

Q2 Activity Configuration - How can human activities be captured in a digital system configuration? How is such a system activity configuration defined? What are the properties and benefits of using activity configuration as an approach to reduce configuration work? How is activity configuration work performed? How are configurations constructed and shared between users and devices?

Q3 Technology - What are the technical and user requirements for technology that supports activity configurations? How can activity-centric technology be used to reduce configuration work in distributed interaction?

1.3 Objectives and Contributions

My thesis is, that by allowing users to interact with computational representations of their real-world activities, creating complex multi-user device ecologies and switching between cross-device information configurations will be more efficient, more effective and provide better support for users' mental model on a multi-user and multi-device environment. By explicitly using activities as fundamental first-class computational structures, these activity-related configuration states can be (i) constructed, (ii) shared and (iii) fragmented across devices and people, thus, reducing the configuration work required to change ongoing work.

The central goal of this dissertation is to *develop and apply concepts and ideas from Activity-Centric Computing to distributed interaction*. I explore these concepts on a conceptual, empirical and technological level and present a framework and use cases for designing activity-centric configurations in multi-device information systems. This leads to three fundamental research objectives and contributions:

Objective 1: Conceptual - *Operationalizing Activity Centric Configurations*
Because current devices and their containing software are still document- and application-centric, they fail to capture and support the activities and context in which they are being used. This leaves users without a stable concept for managing devices and other artifacts as part of their work. To provide users with a stable concept for distributed interaction that is grounded in their activities and use context, we need to build a better understanding of what this human activity and usage context entails and how these insights can be leveraged for distributed interaction. As devices are increasingly becoming portals into ubiquitous information spaces, it has become apparent that devices can no longer be considered as separate entities, but rather as part of larger device ecologies.

I describe the components and structure of configuration work and propose how Activity-Centric Computing can produce *activity configuration* as a *stable concept* for distributed interaction. Using *activity configuration* as a core computational concept, I further introduce a spatial dimension to configuration work, called *activity-centric configuration space*, and describe *activity-centric signifiers* which externalize the activity configuration to the physical world. Finally, I introduce an *activity configuration framework* to describe the relation between activities of users and the devices they use as tools to attain those activities. Previous work on Activity-Centric Computing has placed little emphasis on the role of computer devices and analog artifacts when supporting computational representations of activity. It is unclear how device ecologies fit into the activity abstraction or how devices are configured to be part of changing activity systems. I extend previous activity models with a new *activity configuration* model that explicitly includes a device and tool abstraction.

Objective 2: Technological - *System Design*
First, since there are no existing technological infrastructures or frameworks that support the design, implementation and evaluation of Activity-Centric Computing systems, we[1] designed and implemented NooSphere, an infrastructure and programming framework that mitigates the technical challenges in designing distributed interactive systems. Second, because the success of Activity-Centric

[1]The use of *we* in this dissertation refers to Steven Houben and all co-authors acknowledged in the publications related to the specific research projects.

Computing can only be measured in terms of how it solves the real user problems in distributed interaction, the work in this thesis is focused on two specific use cases: (i) ad hoc knowledge work in a situated office setting, and (ii) clinical work in a patient ward. Based on the conceptual and technical framework, we built a number of interactive systems to implement and explore the activity configuration framework in depth. Each of these systems were designed to tackle a specific sub-problem of configuration work in context of the use case. I extend cross-device system research within Activity-Centric Computing with three different approaches:

- *Replication and Fragmentation of activity configurations* that can be co-created, shared and redeployed across desktop devices that are operated by different users. Users can annex heterogeneous devices (such as tablets) to the activity workspace and use them as physical views on resources and applications.
- *Spatial mediation of activity configuration using a configuration space*, a digitally augmented physical action space that visualizes the activity of the user across all connected devices, using the surrounding space between the active devices. The space can be used to configure devices, interact with resources or share activities with other users. Furthermore, the space uses proxemics to automatically configure devices.
- *Instrumental activity configuration fragmentation* using a smartwatch that elevates the human hand to a reconfigurable instrument that can be used to fragment activity configurations and their content across screens through direct manipulation.
- *Activity configuration surrogates* that allow non-digital artifacts to be included into the distributed digital workflow of users. The surrogate augments non-digital artifacts with configuration, communication and awareness mechanisms to allow it to be used as activity signifier.

Objective 3: Empirical - *Field and User Studies*
To gather evidence of configuration work in distributed interaction and understand and relate it to the domain of knowledge work and clinical work, we conducted field studies, surveys, contextual inquiries and interviews with a number of users. These data were used for input to the system design and to form a conceptual understanding of the problem. Three systems were also evalu-

ated in a field study (co-Activity Manager), a lab study (ActivitySpace) and a clinical simulation (Hybrid Patient Record) to get a better understanding on how these systems affect users and how the introduction of activity configurations mitigates the observed problems.

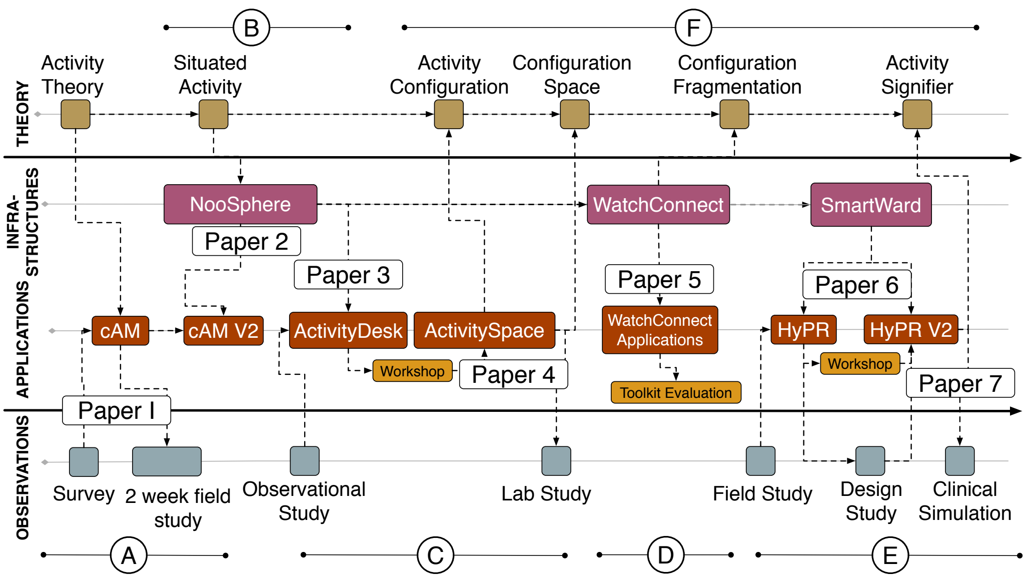


Figure 1.4: The triangulation approach taken in this dissertation.

1.4 Research Method

This dissertation follows the triangulation approach [117] described by Mackay and Fayard, who propose to take multiple perspectives towards approaching problems in Human-Computer Interaction. These perspectives include (i) a theory perspective, in which theory, models and frameworks can be studied and employed to understand or conceptualize a problem domain, (ii) an observation perspective, in which researchers study and observe how users interact with the world and (iii) a design perspective, in which the analysis of data from observations and theory are translated into new design artifacts, which in turn can be the source of observational study and theorizing. To delineate the difference between general technological frameworks and applications that are specifically designed for our use cases, the design perspective is split into an infrastructures and applications swim lane. Figure 1.4 provides a chronological overview of the projects and visualizes the different interconnections between the methods and perspectives used for each project:

A. **co-Activity Manager** - A study of ad hoc collaboration in a multidisciplinary software development team led to the design of co-Activity Manager (cAM). cAM is a multi-user activity-centric desktop interface that is designed to share activity configurations between different users. The system was deployed in a two week field study.
   Methods*: Survey, user interviews, iterative system design, application of theory and field evaluation of system.*

B. **NooSphere** - NooSphere is an infrastructure and programming framework for the design and development of activity-centric distributed interactive systems. The infrastructure was evaluated by implementing three fully working activity-centric applications. Grounded in a field study and demonstrating the multi-device configuration challenges in a hospital ward, we extended NooSphere to the SmartWard infrastructure, which is a multi-device patient management system and application that allows clinicians to reuse and contextualize patient configurations across devices.
   Methods*: Literature review, requirements specification, infrastructure design, application of theory and evaluation through example applications.*

C. **ActivitySpace** - Based on a study of device usage on desks of knowledge workers, we developed ActivitySpace, a multi-device interactive configuration space, that allows users to easily move resources between different devices in use. The system was evaluated in a scenario-based evaluation in a lab setting.
   Methods*: Quick and dirty field study, participatory workshops, iterative system design, conceptualizing and lab evaluation of system.*

D. **WatchConnect** - To explore how a smartwatch can be used as a mediating device that allows for easy fragmentation of configurations across devices, we designed, implemented and evaluated the WatchConnect Toolkit. The toolkit is designed to facilitate the design process of watch-centric cross device interaction techniques and applications. Leveraging the toolkit, we explore how activity fragmentation can be supported using a smartwatch.
   Methods*: Literature review, toolkit design and evaluation through example applications.*

E. Hybrid Patient Record (HyPR) - To mitigate the challenges in managing paper artifacts as part of cross-device information management, we explored the Hybrid Patient Record (HyPR). HyPR is a paper patient record that is augmented with an awareness and configuration device for connecting the paper documents to a distributed activity-centric patient management system. This system was evaluated in a medical simulation. ***Methods**: Field study, participatory workshops and clinical simulation.*

F. Activity Configuration Framework - Based on all systems, I developed an activity-centric framework that describes the creation, sharing, fragmenting and signaling of activity configurations in terms of devices and users. ***Methods**: Conceptualizing and application of theory.*

1.5 Dissertation Overview

The dissertation is organized into two parts. Part I provides an overview of the motivation, conceptual background of the work and description of the technology and systems. The goal of this first part is to provide a general overview and description of related work, the problem statement and the conceptual background. Part II consists of a collection of 7 conference papers that contain concepts, technology and user studies summarized and discussed in Part I. To maintain the consistency, readability and presentation of the original publications, the papers are included in their publicly available published formats.

Part I consists of 5 chapters:

Chapter 2 - *Related Work*
The first chapter of Part I provides an overview of work related to distributed interaction, Ubiquitous Computing and Activity-Centric Computing. Chapter 2 focuses primarily on systems and technologies and relates them to the research questions and problem area that is proposed in this dissertation.

Chapter 3 — *Configuration Work*
Chapter 3 analyzes configuration problems and challenges in distributed interaction, specifically in the domains of knowledge work and clinical work. Based on these challenges and problems, the chapter introduces

the concept of configuration work, provides a definition, describes the underlying processes and analyzes multi-device information management patterns.

Chapter 4 - *Activity Configurations*

Chapter 4 extends Activity-Centric Computing to describe *activity configuration* as a stable concept to mitigate configuration work in distributed interaction. The chapter details the types of device and user configurations that can be created and proposes a framework that considers user and device multiplicity in Activity-Centric Computing.

Chapter 5 - *Technology for Activity Configurations*

Using the activity configuration framework, Chapter 5 provides an overview of the different systems and papers that are included in this thesis. The aim of this chapter is to contextualize each paper from Part II in the activity configuration framework.

Chapter 6 - *Conclusion*

The last chapter of Part I summarizes the findings from the different papers, and describes future work in the field of distributed interaction and Activity-Centric Computing.

Part II consists of 7 papers:

Paper C1 - *Activity-centric support for ad hoc knowledge work: a case study of co-activity manager.*

In this paper, we introduce *co-Activity Manager*, an activity-centric desktop system that (i) provides tools for ad hoc dynamic *configuration* of a desktop working context, (ii) supports both explicit and implicit *articulation* of ongoing work through a built-in collaboration manager and (iii) provides the means to *coordinate* and share working context with other users and devices. Our study showed that the activity-centric workspace supports different individual and collaborative work configuration practices and that activity-centric collaboration is a two-phase process consisting of an activity sharing and per-activity coordination phase.

Paper C2 - *NooSphere: an activity-centric infrastructure for distributed interaction.*

As an approach to re-engineering problems in the design of distributed interactive systems, we introduce *NooSphere*, an activity-centric infrastructure and programming framework that provides a set of fundamental distributed services that enables quick development and deployment of distributed interactive systems. We describe the requirements, design and implementation of *NooSphere* and validate the infrastructure by implementing three canonical real deployable applications constructed on top of the *NooSphere* infrastructure.

Paper C3 - *ActivityDesk: multi-device configuration work using an interactive desk.*
In this paper, we present the design and explorations of the *ActivityDesk* system, an interactive desk that supports multi-device configuration work and workspace aggregation into a personal ad hoc smart space for knowledge workers. The main goal of *ActivityDesk* is to reduce the configuration work required to use multiple devices at the same time by using an interactive desk as a configuration space.

Paper C4 - *ActivitySpace: Managing Device Ecologies in an Activity-Centric Configuration Space.*
To mitigate configuration work, we introduce *ActivitySpace*: an activity-centric configuration space that enables the user to integrate and work across several devices by utilizing the space between the devices. This paper presents the conceptual background and design of *ActivitySpace* and reports on a study with nine participants. Our study shows that *ActivitySpace* helps users to easily manage devices and their allocated resources while also exposing a number of usage patterns.

Paper C5 - *WatchConnect: A Toolkit for Prototyping SmartWatch-Based Cross-Device Applications.*
In this paper, we introduce *WatchConnect*: a toolkit for rapidly prototyping cross-device applications and interaction techniques with smartwatches. The toolkit provides developers with (i) an extendable hardware platform that emulates a smartwatch, (ii) a user interface framework that integrates with an existing UI builder and (iii) a rich set of input and output events using a range of built-in sensor mapping. We evaluate the capabilities of the toolkit with seven interaction techniques and applications that reduce configuration work using the watch as a mediating device

Paper C6 - *HyPR Device: Mobile Support for Hybrid Patient Records.*
Based on design requirements derived from a field study, followed by a design study using a technology probe, we introduce the *HyPR Device*, a device that merges the paper and electronic patient record into one

system. We provide initial results from a clinical simulation with eight clinicians and discuss the functional, design and infrastructural requirements of such hybrid patient records. Our study suggests that the *HyPR device* decreases configuration work, supports mobility in clinical work and increases awareness on patient data.

Paper C7 - *Collaborative Affordances of Hybrid Technologies in Medical Work.*
This paper studies the use of the Hybrid Patient Record (HyPR). We report on two studies: a field study in which we describe the benefits and challenges of using a combination of electronic and paper-based medical records in a large university hospital; and, a deployment study in which we analyze how 8 clinicians used the HyPR in a medical simulation. Based on these empirical studies, this paper introduces and discusses the concept of *collaborative affordances*, which describes a set of properties of the medical record that foster collaborative collocated work.

Part I

Overview and Background

2. Related Work

> *"There is more information available at our fingertips during a walk in the woods than in any computer system, yet people find a walk among trees relaxing and computers frustrating. Machines that fit the human environment instead of forcing humans to enter theirs will make using a computer as refreshing as taking a walk in the woods."*
>
> — ***Mark Weiser*** [185]

This chapter provides an introduction to Ubiquitous Computing and distributed technologies that are related to the research questions and problem area described in the introduction. The purpose of this chapter is to provide a detailed sampling of existing technologies and approaches and use this overview to motivate and describe open challenges in Activity-Centric Computing. Section 2.1 first introduces the ubiquitous computing concept followed by an overview of smart space research and multi-device interaction techniques. Finally, Section 2.2 concludes this chapter by providing an in depth overview of prior work on Activity-Centric computing.

2.1 Ubiquitous Computing

In 1991, Mark Weiser provided a detailed account on the future of computing [185]. He described a vision of modern computing, which he referred to as *Ubiquitous Computing (UbiComp)*, in which computers would be integrated into the environment, allowing users to easily and seamlessly use computing power as part of everyday interaction with the world. Arguing that "*the most profound technologies are those that disappear*", he set a benchmark in the age of computing, which transcended the desktop computer designed for knowledge workers, and introduced a new generation of devices, called *tabs*, *pads* and *boards*. These devices were designed to support a range of specialized tasks and provided users with seamless interaction spaces that could intelligently respond to changes in the environment. Weiser envisioned a world in which hundreds of these devices would be interconnected to a *Ubiquitous Network* that would support *embodied virtuality*, in which the virtuality of data is "*brought into the physical world*" and computers are freed from their "*electronic shells*" [185]. At **Xerox Parc**, these ideas were operationalized into a fully working *roomware* system that consisted of a large interactive display, *LiveBoard* [60], a mid-sized hand-held device called the *ParcPad* and the smaller *ParcTab* [184] (Figure 2.1). These devices were interconnected and allowed users to seamlessly move information, input and output between all present devices. *Active Badges*, small devices that transmit an infrared identification signal, were used to track the location of users and devices, and leveraged this knowledge to create an intelligent environment that could, e.g., automatically open doors or forward input and output to devices in range. The Xeroc Parc systems essentially allowed users to grab and use any device in the environment, based on the *activity* and the *situation*. Weiser realized the impact of his vision on the interaction between humans and computer devices. Inspired by the work of Suchman [170] and Lave [109], he described the paradigm as "*a radical direction, for computer science, away from attention on the machine and back on the person and his or her life in the world of work, play, and home*" [187]. The central emphasis of Ubiquitous Computing is not on the machine, but on the role of the machine in the world of the user. At its core, Ubiquitous Computing shifted the focus towards understanding the "*place of today's computer in actual activities of everyday life*" [186], implying it should "*focus on the task, not the tool*" [188]. Furthermore, Weiser argued that the most challenging and profound changes induced by Ubiquitous Computing lies in the focus on *calm technology* [189] that engages both the periphery and center of our attention, and "*overcome the problem of information overload*" [185].

2.1.1 Smart Spaces

Among many other research topics, the seminal work by Weiser and his colleagues spawned a wide range of systems and infrastructure research that focused on supporting seamless information spaces or *roomware*, in which computers where integrated into the furniture and architecture of the room [169]. In these rooms, digital data is distributed across multiple screens, computing devices and even physical artifacts. One of the earliest approaches, **I-LAND** [168], attempted to blend the virtual and architectural space into one innovative workspace that

Figure 2.1: The Xerox Parc roomware allowed users to pick up and use pads and tabs in combination with desktop machines or large electronic whiteboards.

consisted of interactive walls, tables and chairs. I-LAND consisted of several large interactive displays (DynaWall) to support collaborative setups in which information was used by multiple people. It also included chairs (CommChairs) that were augmented with pen-based digital slates or a laptop docking system. Finally, I-LAND also included a number of interactive touch tables (InteracTable) that could be used for ad hoc collaborative work. All devices were connected through a custom infrastructure and software (COAST & BEACH) that allowed for seamless information exchange using Passage, a concept in which information is transported by connecting virtual data to physical bookmarks. These bookmarks were readable by the different I-LAND components, thus, allowing for physical and ephemeral binding of data to specific roomware devices. This was to ensure that it was "*no longer necessary to open windows, browse hierarchies of folders, worry about mounted drives, etc*" [168]. Similarly, the **iRoom** [90] system operationalized the UbiComp vision of Weiser [185] using a room-based smartspace system that consisted of interactive whiteboards, smaller PDAs, laptops and workstations, an interactive table and a number of peripheral interaction devices. iRoom was designed to facilitate moving data, control and applications between different available screens. Built around an Interactive Room Operating System (iROS), devices with different operating systems (OSs) could be interconnected in the room using a common data heap. iRoom also included scanners to integrate paper sketches into the general digital flow and provided simple wireless devices (e.g., sliders and buttons) to operate specific interface elements in the room. Using the Pointright [91] system, iRoom allowed for pointer and keyboard redirection to create a truly homogeneous workspace consisting of multiple machines and users. Similar to I-LAND [168], the goal of iRoom was to "*let the user remain focused on the work being done, rather than on the mechanics of interaction*" [90].

Inspired by the vision paper on 'the office of the future' [143], Rekimoto and Saitoh introduced **Augmented Surfaces** [147], a continuous workspace consisting of personal portable devices and pre-installed public computers. The core idea behind Augmented Surfaces was to create an environment that allowed for smooth and fast interchange of information between different collocated public and private computing devices. Augmented Surfaces employed a large camera-based interactive surface (INFOTABLE) as an extension to the workspace of smaller private devices, such as a laptop. By placing the private device on the surface, it was automatically connected to the surface but also to an interactive wall (INFOWALL). Other users could join the setup, by similarly placing their private devices on surface. To facilitate smooth and visual exchange of information, Augmented Surfaces provided a number of novel interaction techniques and visualizations, such as hyperdragging, that allowed users to drag and drop data across screen borders while visualizing the cross-device movement of the cursor using the anchor cursor visualization. Similarly to I-LAND [168], Augmented Surfaces allowed users to attach digital information to physical objects. Using visual markers to detect non-digital objects on the interactive surface, the system projected a visual aura around the object. This aura represented the virtual data spaces of the physical object and users could attach data to the object by dragging the digital information to the aura. Rekimoto and Saitoh showed how Augmented Surfaces allowed for a smart public environment that can be appropriated into different individual or collaborative configurations.

Realizing that existing systems and interfaces would continue to play a central role in human-computer interaction, a number of other systems focused on more explicitly overcoming their limitations and capabilities, to include them in distributed information spaces. **Project Aura** [69], e.g., was built around the *cyber foraging* concept in which *surrogate* servers were used as stable connection points of the Ubiquitous infrastructure. The surrogates allowed for nomadic and mobile use of PDAs and laptops, and integrated support for legacy systems, such as Windows and Linux. This essentially allowed users to incorporate of-the-shelve devices and operating systems (OS) into a distributed smart space system. To support users in expressing their intent, Aura included a task layer which was used to abstract applications and services into higher level tasks that could be moved between devices [69]. The **Gaia** meta-operating system [152] supported *active spaces*, which are defined a physical spaces that are augmented with technology to enhance the "*mobile users' ability to interact with and configure their physical and digital environments seamlessly*" [152]. Gaia OS included a number of meta services (such as IO, file system, communication, error handling, and resource allocation) in a distributed architecture that allowed both legacy computing devices and specialized interactive devices to be interconnected into one seamless space. The central goal of Gaia OS was to create a *user virtual space* that allowed users to perform a wide range of activities, which were stored in sessions that could be moved between different active spaces. Using Gaia [152] as infrastructure, **Aris** [30] is a window manager that allowed users to easily share legacy windows (from the Windows XP OS) across different heterogeneous connected devices. By leveraging existing windows and introducing a *world in miniature* overview of the smart space, Aris essentially supported direct manipulation for application relocation and the redirection of input across devices. **IMPROMPTU** [31] is an example of a multi-display interaction framework that specifically focused on collocated collaborative work using legacy applications. IMPROMPTU allowed users to share *task information* on shared displays to allow for multi-user interaction. Using a number of extensions of the Windows OS, users could share selected windows with the group or move applications to a large shared display. For all applications users could change the sharing mode to allow for read-only or full access to the content. Using IMPROMPTU, users could easily share resources that were part of collaborative activities and use these resources to create a shared awareness across all users. A final multi-screen approach that focused on integrating with legacy applications is **Window Brokers** [5]. Using an automatic, plug-in free window broker protocol, users could move windows between devices in different locations by using a display server that is connected to individual broker clients, that decided how the windows could be accessed and organized. By integrating legacy systems, these approaches essentially argued that the problem of creating multi-device environments lies in providing a solid infrastructure that provides a number of fundamental functions and services that allow existing applications and tools to break the boundary of their original creation. Users are able to translate their existing knowledge on computers to these type of distributed information spaces.

A number of other approaches moved away from legacy user interfaces, and introduced new interfaces and concepts for multi-display environments. **Dynamo** [88] is an example of a multi-user interactive surface that was specifically designed to support cooperative sharing and exchange of resources, that can be moved into the space. Using a custom user interface that focused explicitly on providing multi-user input and sharing capabilities, users could utilize and arrange parts of the surface for their own work. Users were enabled to *carve off* part of the interactive surface and share that surface with other users that are relevant to the activity. Legacy files and resources could be added to the surface using USB and would be parsed and visualized on the Dynamo UI. Dynamo utilized a display server and a number of central managers to handle distributed input, window management, telepointing and other services. Speakeasy [58] employed the concept of *Recombinant Computing* [59] in which they re-envisioned how interoperability between heterogeneous devices could be provided using a set of common interaction patterns that leverages mobile executable code. Using these interfaces and code, users are able to create device configurations and cross-device applications with very limited a priori knowledge on the different devices. These interfaces included data transfer, collections that described the relation between entities, metadata providing contextual information and control describing how users could affect the setup. Similar to Dynamo, **Xice** [4] is a distributed application toolkit that uses a scene-graph or presentation tree to model the user interface. These scene graphs are propagated across the network using a custom protocol to allow multiple devices to render and show the interface. Input is supported by allowing users to interact with the local scene and using that input to modify or change the tree on other devices. This allows Xice to support a wide range of input redirection, window interactions and cross-device applications, but also greatly limits its capabilities to a defined subset of interactions. Based around the concept of Instrumental Interaction [24], **Shared Substances** [73] employed a data-oriented approach in which functionality and data are loosely coupled, allowing for more flexibility in the design and implementation of large scale multi-display environments. As demonstrated in the WILD room [25], the framework and underlying concepts scale to large display sizes, heterogeneous devices and big data, which facilitates exploration of scientific data, as suggested by Rogers [151].

As described by Saha and Mukherjee [153], these pervasive systems require four fundamental technical components: *devices* ranging from traditional computers to fully embedded sensors, a *pervasive network* that can be used as the backbone of the system, a *pervasive middleware* that is used as a distributed shell to interface between different devices and *applications* that leverage the infrastructure and network. Based on these components, they described six fundamental challenges for Ubiquitous Computing [153] and distributed interaction:

- C1 *Scalability*: How to deal with an increasing number of devices, applications and systems that are distributed across different users and geographical spaces?
- C2 *Heterogeneity*: How to support cross-domain systems and applications that are interconnected through the network and support different platforms and devices?
- C3 *Integration*: How to integrate existing devices and systems into different pervasive systems and middleware? How to coordinate information and data across distributed spaces?
- C4 *Invisibility*: How to balance the relation between human intervention and self-tuned smart environments? How to automatically support connection of devices and applications?
- C5 *Context Awareness*: How can computer systems sense the use context to become intelligent environments that support user experience?
- C6 *Context Management*: How can context information be translated into an interaction that is meaningful and understandable to the end user?

2.1.2 Context-Awareness

The core argument behind these smart space systems is that the focus should be set on the task or activity of users, and the role of computing devices in the everyday life of the user. This argument, however, implies that user is able to express *intent* to these complex environments by translating this intent, ideas and configurations to the machine. As argued by Abowd et al. [2], machines often fail to understand the dynamics of implicit human interaction and fail to include *implicit situational information* or *context* into the human-computer interaction. Schilit et al. [157] provided one of the first definitions on context-awareness, describing it as "*systems that adapt according to the location of use, the collection of nearby people, host and accessible devices, as well as to changes to such things over time.*" Context-aware systems include three types of context: "*where you are, who you are with, and what resources are nearby*" [157]. Based on this early definition, Abowd et al. [2] expanded the notion of context to: "*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.*"

	Manual	**Automatic**
Information	Proximate selection and contextual information	Automatic contextual reconfiguration
Command	Contextual commands	Context-triggered actions

Table 2.1: Dimension of context-aware applications [157].

In their early paper on context-awareness, Schilit et al. [157] described four main categories of context-aware systems that reside on the intersection of manual or automatic work, and information and command input (Figure 2.1). Proximate selection is a technique in which nearby information sources are emphasized and selected in an effort to reduce the configuration time of the environment. Important within this category are nearby or collocated computing devices, people, physical artifacts and the places of interaction. In contrast, automatic contextual reconfiguration uses this location information to automatically re-configure devices. Using contextual commands, explicit input from users can be contextualized based on the ongoing situation. However, this can also be automated using context-triggered actions, in which implicit input by the user (e.g., movement or location) is used to automatically trigger actions that reconfigure the space. Based on early research on the Active Badges projects [183, 185], a large body of work focused on location aware systems that supported automatic reconfiguration of applications and devices [1, 44, 110]. Although these definitions and dimensions are useful to describe the close connection between *manual configuration* done by the user and automatic context-aware configuration done by an intelligent system, Schmidt et al. [158] argued that "*there is more to context than location*". By expanding the notion of context to a wide range of features, Schmidt et al. [158] introduced a clear differentiation between context in human factors and physical environment. Human-centered context included the user's physical condition, emotional state, social context of interaction and the tasks or activities. The physical environment focused on location in the broad sense, infrastructure and physical conditions. Recently, the notion of context has been refined further into "proxemic interaction" [8, 76]. Inspired by the theory of *Proxemics* by Hall [78], Ballendat et al. described how devices could have "*fine-grained knowledge of nearby people and other devices – their position, identity, movement, and orientation*" [8]. This fine grained context model, that describes a granular spatial relation between devices, can be leveraged to support advance cross-device interaction techniques and approaches.

2.1.3 Multi-Device Interactions

With the increasing widespread use of different types of devices, the ideas and concepts discussed in early roomware systems (described in Section 2.1.1) are becoming increasingly relevant. In recent years, we have seen an explosion of novel devices, ranging from small devices, such as watches and phones, to mid-sized devices, such as tablets and notebooks, all the way up to very large interactive displays and tabletops. This spectrum of new devices has led to an '*ecosystem of displays*' [175] that supports a wide variety of social setups ranging from individual use, up to a many to many social group interactions [175]. The expanded notion of context-awareness of humans and their computing devices [158] has initiated a number of new solutions for cross-device interaction problems in these types of multi-device ecosystems. Captured in the '*Gradual Engagement Pattern*' by Marquardt et al. [118], many interaction techniques and approaches have been proposed to mitigate the distributed configuration problem by using context as a mechanism to support users in (i) getting an overview and awareness on the available devices and their capabilities, (ii) understanding what information can be used on each device, and (iii) seamlessly transfer these resources across devices. As summarized by Marquardt et al. [118], these systems and interaction techniques focus on three important challenges within multi-device environments: (i) device pairing, (ii) awareness and revealing of information and resources, and (iii) transferring information from one device to another.

There are different approaches reported in literature to pair and annex devices. Hinckley [84] introduced interaction techniques that allowed users to physically bump different devices together to pair the devices into a shared workspace. Bluetable [190] used computer vision of interactive surfaces with Bluetooth to detect and pair mobile devices to a large display. Similarly, Tide [164] allowed users to pair a mobile phone to a large interactive table using computer vision. Schmidt et al. [159] extended this method and were able to detect fine grained movement and placement of the edges of the phone on an interactive surface using a combination of computer vision and IMU data. A final example, is the F-formation detection by Marquardt et al. [119], in which top mounted 3D cameras were used to detect fine grained movement of devices that could be annexed into one seamless space. Multi-device environments have opened up a large space for cross-device resource management mechanisms and interaction techniques. Some early work such as Pick and Drop [146] explored how users could apply direct manipulation to multi-device environments by picking up information on one device using a pen and dropping it on the screen of another device. Touch and Interact [79] uses physical connection between a phone and a large display to perform selections. Similarly, using Phonetouch [159], Schmidt et al. [160] introduced a suit of interaction techniques that allowed users to interact with a large interactive tabletop, using the phone as input device. Touch and Point [38] uses a combination of mid-air gestures for social disclosure, with touch for precise selection to support ad hoc group work around a large interactive display. The approach allows users to interact in mid-air with the shared display, while providing fine grained selection through a personal touch-enabled device. Deep shot [42] utilizes the camera of a smartphone to capture the context and state of the applications on another device, and moves this to the mobile application.

This approach allows users to create a snapshot of a task and resume the task on a different device. Recently, Duet [46] described a suite of interaction techniques, specifically designed for a combination of smartwatch and tablet, in which users could configure and share sensors and information across the devices in use. Many of these systems and techniques, however, operate under the assumption that data, applications and services are already set up and available for use. How these techniques and ideas fit into *how* people actually use devices in information dense environments, is unclear.

2.2 Activity- and Task-Based Computing

Long before the introduction of Ubiquitous Computing, Bannon et al. [10] realized that within desktop computing, "*current human - computer interfaces provide little support for the kinds of problems users encounter when attempting to accomplish several different tasks in a single session*". Inspired by *Activity Theory* [63, 98, 181], this early work initiated a new research direction, *Activity-Centric Computing*, that revolved around using context models that are a reflection of human intent. These computational activities are used to structure, organize and share information within a defined workspace. Activity-Based Computing, a term originally coined by Donald Norman at Apple Research [130], is an interaction paradigm that provides explicit support for users' activities, rather than focusing on the tools needed to perform those activities. These activities are presented as a computational context model that encapsulate all information and resources relevant for that specific activity. The core idea behind Activity-Centric Computing is, thus, to better support human activity in computer-mediated interactions. Figure 2.2 provides a historical overview of recent Activity-Centric Computing systems.

2.2.1 Activity-Centric Desktop Computing

One of the earliest systems that, perhaps unintentionally, introduced concepts from Activity-Centric Computing into the desktop interface was the **Rooms** system [82]. The Rooms system mitigates a number of window management issues by introducing virtual workspaces. Each virtual workspace, or room,

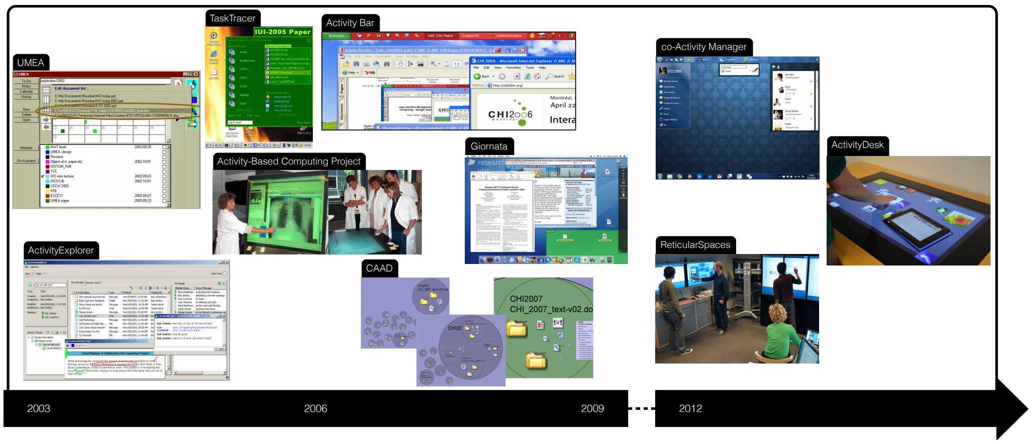


Figure 2.2: An overview of the main Activity-Centric Computing systems researched in the last decade.

essentially provides the user with a clean canvas that can be allocated to a specific task. By switching between rooms, the window configurations of the previous room is stored and removed, while the windows of the selected room are restored on the screen. Users can name and organize each room according to the specific task it represents. Using a number of built-in mechanisms, users can share or replicate windows across rooms using *doors*. The Rooms system also provides an overview screen that lists all defined rooms with their name and pictorial representation of the window layout. This early work resulted in the widespread adoption of virtual desktops in many Linux distributions, Apple OS X and recently even in the Microsoft Windows 8 OS, demonstrating the relevance and impact of virtual spaces.

Task Gallery [148] extended the Rooms metaphor to a three-dimensional view. Multiple documents and windows can be hung side-by-side on the walls of the 3D space. By exploiting spatial memory and using an art gallery metaphor to organize documents and windows, Task Gallery was designed to ease information retrieval by organizing these resources into tasks, which are "*collections of documents and applications organized around a particular user activity* [148]." Users can move tasks around within the 3D space and can even segment the space into separate rooms, thus, allowing for grouping of task windows. Using a number of tool palettes and window banners, users can create and manage tasks, interact with windows and navigate the space. Interestingly, Task Gallery [148] explores how spatial memory can be leveraged to reduce mental load when switching between different tasks (or activities) at hand.

One fundamental limitation of the Rooms [82] and Task Gallery [148] systems is that the content of the task is confined within one device and screen. To mitigate this space problem, **Kimura** [116, 179] expanded the display space by including a projector based interactive secondary display that can be used for "*perusal, manipulation and awareness of background activities*" [116]. The peripheral display is used to assist users in managing working contexts or *montages* by providing an overview of previously automatically collected activity logs. Each montage depicts the content of an individual room and is essentially an external visual room overview [82]. Montages help users recall previous actions that were part of an activity, thus providing awareness on background activities. The secondary display can be used for peripheral awareness, but also to directly interact with the montages. Since the display runs Flatlands [126], montages can be moved, manipulated and deleted. Montages can be organized into spirals based on their significance, spatially organized stacks or be interrelated with other activities. In Kimura, the tasks as described in Rooms [82] are extended "*to "activities" that include more than just the documents and application windows currently being used*" [116].

The first approach that explicitly incorporated communication mechanisms into a task or activity model, was **Taskmaster** [26]. By organizing email into tasks, the focus of the user shifts from individual emails to a collection of relevant emails. Because many individual messages represent tasks, they can be interrelated to other tasks. Introducing the *thrask* concept, incoming messages are analyzed and grouped based on their relevance. Taskmaster includes a to-do list with thrasks that helps users to get an overview of the groups. Thrasks essentially correspond to threads of activities. Users can rename and modify the thrasks or design their own activity threads. By selecting a thrask, the interface provides a mechanism to inspect attached resources. By adding meta-information to thrasks, deadlines and reminders can be set that trigger automatic notifications. To increase recognizability in thrasks, users can also add color or icons to the representation.

To place a more explicit focus on supporting the intention behind the activity, **Umea** [97] moved away from virtual desktops, but rather supported a desktop tool that captured events from applications to automatically build projects. Projects are visualized on a calendar-style visualization that allowed users to create, manage and delete projects. For each project the relevant resources, such as documents, folders, URLs and contacts are easily accessible through a floating windowed toolbar. Selecting a resource, automatically restores its previous state. Umea actively tracks (a selected number of) applications to build interaction histories that are added to the currently selected activities. This allows users to create basic configuration for their activity and use the system to automatically build up a broader context. In this way, the intent of users can easily be captured and developed over time. The central underlying idea of Umea is "*minimizing overhead and making the benefits of creating project environments apparent to the user*" [97].

Groupbar [165] extended the standard Windows XP taskbar, that allows for window switching, with a mechanism to organize and interact with a group of spatially organized windows. Users can drag and drop window buttons on the taskbar into a group, that represent a task or activity. This group visualization can be opened and inspected to allow for traditional interaction with individual windows, but users can also interact with the entire group. Using a context menu, users can open, close, restore and layout the entire group. In contrast to Rooms [82], restoring the window group does not remove any other windows, but rather pushes the windows of the selected group to the foreground, allowing the user to easily move them into focus. Groups can also be collapsed or minimized to save space on the taskbar. In contrast to the application grouping mechanism that is used in current Windows OS, Groupbar allows for the grouping of windows of different applications. Groupbar demonstrates how the taskbar (or toolbar) of a desktop manager can be augmented and extended to support activity-centric workflows.

Extending on the concept of automatic context inference on desktop systems, **Tasktracer** [54, 163] is a task-based desktop interface that monitors users' interactions and activities to allow users to access previous activities and quickly change the task context. Within Tasktracer, a task "*is a set of information resources (documents, electronic messages, contacts, etc.) and tools (computer applications, phone*

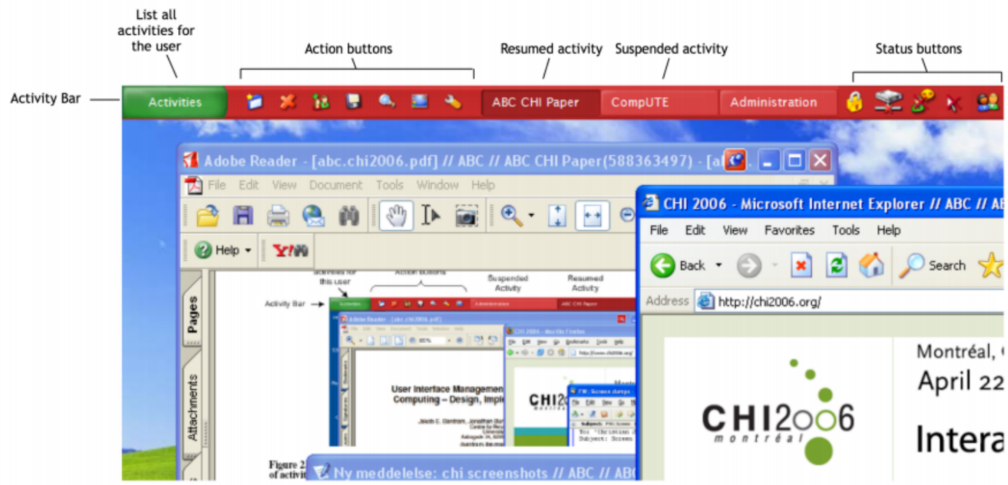


Figure 2.3: The activity bar allows users to interact with activities on a taskbar [13].

line) employed to access and manipulate these resources" [54]. Using a data collection framework, Tasktracer captures users' interactions with applications in order to infer task profiles. The goal of Tasktracer is to leverage these task profiles to support users in recovering from interruptions and allow users to reuse knowledge between tasks. To visualize tasks to the users, Tasktracer supports three different visualizations: a stand-alone window listing all the tasks, a context menu built into the standard taskbar, a list of activities integrated into the start menu and a stand-alone window, TaskExplorer, that provides an overview of the resources in each task. Selecting a task provides users with quick access to the resources associated with that task.

Similar to Tasktracer [54, 163], the Context-Aware Activity Display (**CAAD**) [144] automatically monitors interactions on the desktop interface to construct *context structures*. The structures represent users' actual work activities, which are defined as "*the sets of task-relevant information and people*" [144]. The collected structures are visualized on a secondary awareness display (the activity display) as clusters of activities that can be edited and managed by users. By visualizing groups of related information, the system provides users with an awareness on their ongoing and previous work.

Expanding on the idea of using a taskbar to visualize activities, the **Activity Bar** [13] (Figure 2.3) is used as a central access point to select, represent and interact with activity representations on a desktop. Using the Activity-Based Computing (ABC) framework, "*the basic computational unit is no longer the file (e.g., a document) or the application (e.g., MS Word) but the activity of a user*." Within the desktop interface, the Activity bar extends the standard task bar, since the focus is no longer on applications but on activities. Each activity is confined within a virtual desktop that allows users to organize information and resources within that activity. The Activity bar visualizes all open activities and allows for easy switching between activities by simply clicking the button associated with the activity. The bar also includes action buttons that allow users to create new activities, close existing ones or save the activity to a file. The interface also provides an overview of all open activities on the desktop workspace in a miniature view. Similar to window management, users can utilize shortcuts to easily switch between activities.

Based on earlier work by Bernstein [29], who suggested that distributed workflows could be structured around tasks, **ActivityExplorer** [7, 70, 121, 125] supports an informal, ad hoc and easy to initiate collaborative mechanism organized around activities. Similar to Taskmaster [26], Activity Explorer (Figure 2.4) allows users to create *activity streams* in which messages, chats, files, folders, annotated screen shots and to-do items can be shared among a group of people. Activities are shared objects that "*hold one piece of persistent information, and they define a list of people who have access to that content*" [125]. By including both synchronous and asynchronous collaboration into one persistent activity, which is represented as a dialog thread, Activity Explorer allows for an ad hoc collaboration model. It provides dynamic membership, meaning that users can easily add and remove team members to a specific activity. The interface is an email-like overview of all activities and includes details on the activity threads, chat windows, an overview of all contacts and details about the activity itself. In contrast to previous described systems, Activity Explorer moves away from dedicated workspaces and rather focuses on using activity as a mechanism for a stand-alone collaboration manager that can be used as part of standard desktop use.

Similar to the Activity Bar [13], **Giornata** [177, 178] (Figure 2.5) extended the concept of Rooms [82], and Kimura [116, 179] and introduced an *Activity Theory*

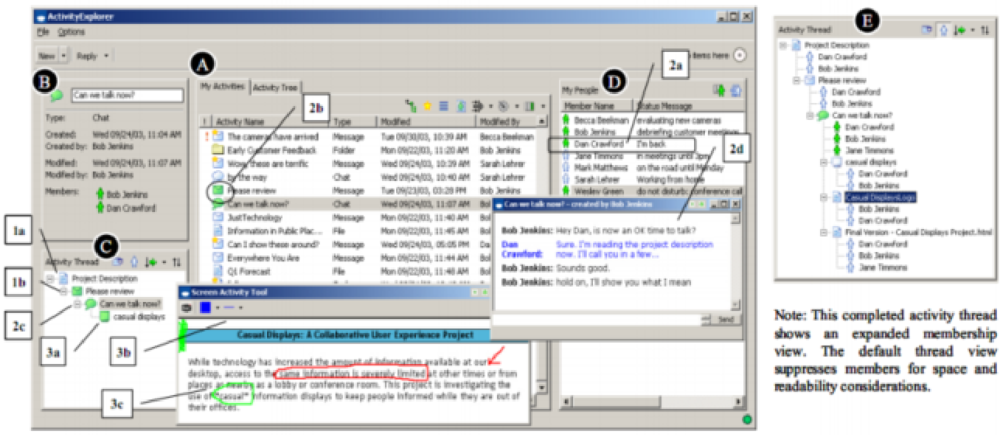


Figure 2.4: The activity explorer user interface is composed of an email-like interface that presents activity threads used to share objects between users [125].

informed desktop interface that allowed users to organize information and windows related to the same activity into dedicated workspaces. In contrast to the work of Bardram et al. [13], in which activities were relatively rigid structures, Giornata focuses on providing a much more lightweight mechanism that allows users to appropriate different approaches when interacting with activities. Rather than explicitly defining activities, Giornata allows users to seamlessly create anonymous activities which can be retrospectively named by adding tags to the activity. Using the command + tab shortcut, users switch between open activities, which are visualized with a name and snapshot of the workspace. As one of the first systems, Giornata also included contacts management into the activity desktop abstraction. Users can organize their contacts into groups that, similarly to the windows and resources, are swapped when switching between activities. This notion of "*activity-aware collaboration*" allows users to move parts of the communication processes into the activity abstraction.

Critiquing the rigidness and lack of plasticity of using virtual spaces to represent tasks or activities, **TAGtivity** [133] uses a flexible tag-based model to connect windows and application on a desktop interface to a specific activity. Focusing specifically on activity management, which they define as "*supporting users in managing their applications and application windows*" [133], users are provided with an interface that (i) augments each existing window with a TAGtivity bar (which

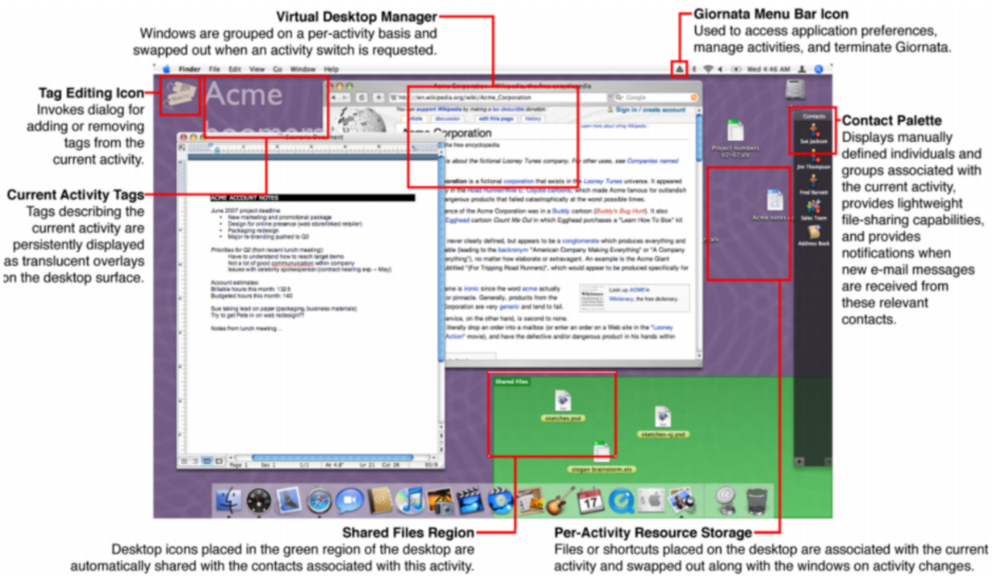


Figure 2.5: Giornata allows users to organize their desktop using flexible and lightweight activities that encapsulates windows and resources but also contacts [178].

is attached to the bottom of the window), and (ii) the TAGtivity manager, that provides an overview of all open activities. Using the toolbar, users can add color, tags, names and other meta information to the window, thus associating it with a new or existing activity. Using the bar, users can easily modify or update the relation of the window to any open activity. All existing activities are visualized in the manager, which provides users with an interface to easily browse the windows and applications associated with that activity, and launch them if needed. The central idea behind TAGTivity is to provide more fine grained activity management on the level of each individual window. This allows users to quickly and easily create or modify the association between the resources and windows.

2.2.2 Activity-Centric Multi-Device Environments

As mentioned earlier, most of the existing multi-device smart space systems essentially point towards the need to focus on the task or activity of users, and the role of the computing devices in the everyday life of the user. Early on, Lamming et al. [107] described a need to focus on *Activity-Based information retrieval* in Ubiquitous Computing systems. The central vision behind this idea was that users would carry portable memory aids that supported them in "*browsing through autobiographical data to find episodes corresponding*" [107]. By essentially linking context information to episodes or activities, users are aided in information retrieval. In their reflections on the "*Past, Present, and Future Research in Ubiquitous Computing*", Abowd and Mynatt extended this idea of using activities as a unit of analysis for *everyday computing* [3]. Although they leave us without an explicit definition of the concept of *activity*, they exemplify them as "*orchestrating tasks, communicating with family and friends, and managing information*". Activities do not have a defined temporal dimension as they are continuous, meaning they can be interrupted by internal or external sources. Because of this, multiple activities operate concurrently and users can suspend and resume activities based on their interest and intent [3]. They also argued that applications should use associative models of information to support activities within Ubiquitous systems [3], because these "*associative and context-rich models of organization support activities by allowing the user to reacquire the information from numerous points of view.*" By supporting associative computing models that reflect human intent, users are given a flexible and extendable mechanism to handle information within the current context. As summarized by Abowd and Mynatt: "*as computing becomes more ubiquitously available, it is imperative that the tools offered reflect their role in longer-term activities*" [3].

Prekop and Burnett [141] operationalized this idea into an *activity-centric context*. Because it is important "*to understand the properties of context and the relationships between context and other closely related concepts, especially, tasks or activity and users or agents*", Prekop and Burnett proposed to view context specifically as a subset of human activity. In their model, activity is defined as "*something being done by the agent*", providing a wide scope ranging from 'getting a cup of coffee' to 'working on a specific project'. The model places an explicit focus on human agents that perform activities within a specific context. One of the central problems that they tried to address is to get a better understanding of what information is relevant to the activity performed. This includes resources such as information, computing devices, applications and people. These resources are used as part of a *process*, which describes how resources are related to the specific activity.

Activity-centric context emerges from using resources of a specific activity within a specific context.

Drawing upon early research on an activity theoretical analysis of cooperative work in hospitals [12, 16], Christensen and Bardram applied the notion of *Activity-Based Computing* [48] to distributed work in hospitals. They proposed to "*model work activities as first class objects in the computing infrastructure thereby lessening the gap between the health care tasks and the work done using the computer*" [48]. Crystallized into the **Activity-Based Computing framework**, Bardram [18] proposed a pervasive infrastructure to support clinical work across users, devices and spaces, using activity as a central computational concept. The system allows clinicians to organize patient-related information and workflows into activities, that can be accessed from different workstations and PDAs running specialized medical software. Using the Activity Bar [13] (Section 2.2.1), clinicians can switch between patient cases that are represented as activities. These activities are synchronized through an infrastructure to allow access from other workstations. The system also includes a number of collaborative functions to allow clinicians to start a synchronized desktop sharing session, in case remote collaboration is required.

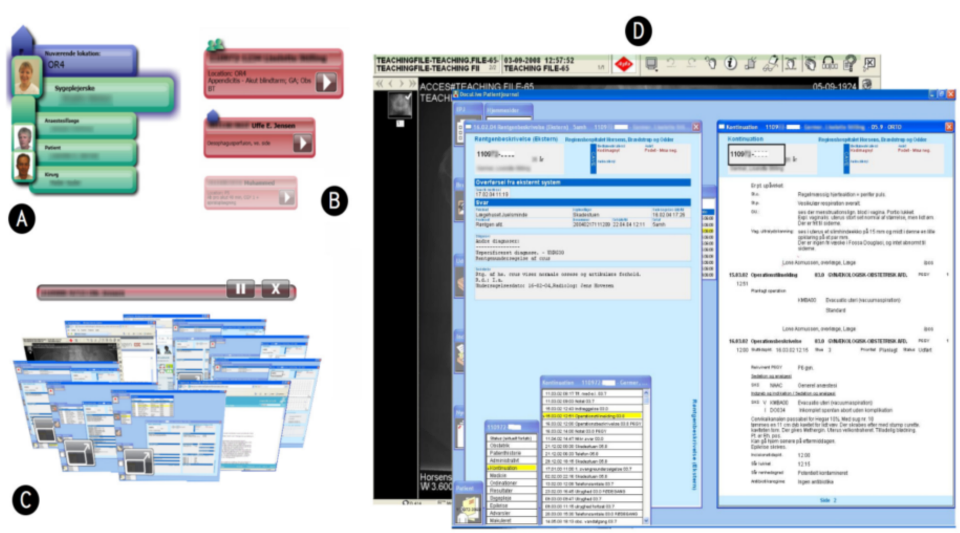


Figure 2.6: Clinical surfaces is designed to support activity-based access to medical information on public interactive displays [21].

In an effort to expand this paradigm to large interactive displays across a large building, **Clinical Surfaces** [21] used the Aexo [39] infrastructure to support the distribution of clinical activities across multiple public displays. The goal of Clinical Surfaces is to facilitate the use of fixed and mobile public interactive displays to navigate large amounts of clinical data. The interface running on the displays (Figure 2.6) visualizes the context, such as current location and nearby people, an overview of all activities and the applications, documents and resources of the selected activities. Using a basic relevance algorithm, based on location and nearby people, the interface adapts the activity view to push the most relevant activities to the foreground. Resources (such as documents, URLs or database files) can be launched using a local application. Clinical Surfaces includes a privacy model that allows clinicians to operate the displays in public, personal or private mode. In the default public mode, activities are presented based on all people nearby; in personal mode, the view is adapted for one specific clinician; and private mode can be activated by inserting a USB stick containing private activities. Because activities are well defined within a clinical context (e.g., patient data and related clinicians), the system places little emphasis on the creation and maintenance of activities, but more on the access model.

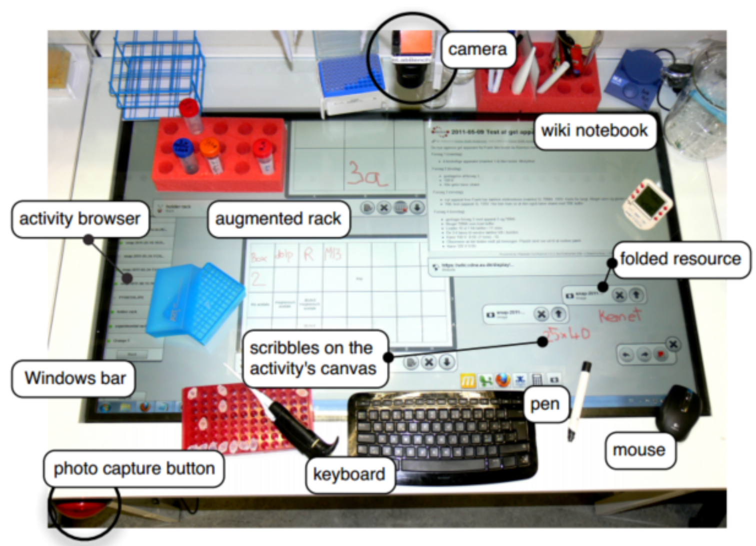


Figure 2.7: The eLabBench allows biologists to move activities between an interactive desk and their personal computer in the office [172].

The **eLabBench** [171, 172] uses activities as a central mechanism to distribute resources between a desktop computer and an interactive laboratory bench. In order to support molecular biologists in performing experimental work, the eLabBench allows them to easily construct, design and analyze exploratory biology work using an augmented tabletop that is used as a work bench. The activity-based infrastructure allows biologists to easily move the information about the experiments between their personal computing device in the office and the eLabBench. The interface of the eLabBench (Figure 2.7) includes an *ActivityDock* that is used to switch between different ongoing activities. The rest of the space is used to visualize resources that are part of the selected activity. This includes a digital notebook, scribbles and native applications, but the desk also recognizes physical artifacts. Augmented tube racks are recognized and visualized on the space. Biologist can annotate and include the rack descriptions into the digital workflow. Although very limited, the system demonstrates how physical artifacts can be included into the activity abstraction and used beyond the local space.

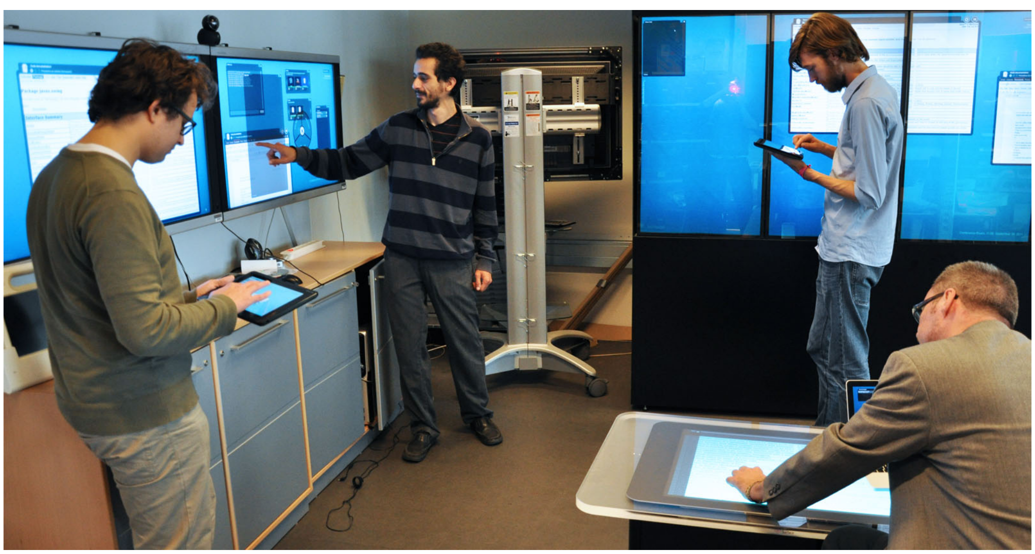


Figure 2.8: ReticularSpaces is a multi-device smart space that includes activity-based support for large displays, tabletops and tablets [15].

Finally, in an attempt to build a complete activity-cased smartspace system (similar to the original Parc system [185]), **ReticularSpaces** [15] provides a first cross-device user interface concept using activity as a first class visualization

(Figure 2.8). Using an activity interface that was the same on all connected devices, the system provides a unified distributed view. The ReticUI interface allows users to switch between the activity view, showing all activities and their interdependencies, and an action view that contains all the resources of that activity. The resources, such as websites or images, are encapsulated into touch-enabled windows that can be moved and organized on the space. Any changes to the location or content of each resource is synchronized with all other devices. Similar to Clinical Surface [21], the system uses a location tracker to adapt the activity view based on the people that are present in the room. The interface includes a number of real-time communication mechanisms, including video and chat, but also provides asynchronous methods using the activity log.

2.3 Summary

This chapter provides an overview of related work in the fields of smartspaces in Ubiquitous Computing (Section 2.1.1) and the relation with Context-Awareness (Section 2.1.2). Based on these early multi-display environments, Section 2.1.3 then briefly discussed a number of interaction techniques for the pairing and distribution of information across devices. Finally, Section 2.2 provided a detailed overview of system research within activity-centric desktop and pervasive computing. Detailed related work for each project is described in the papers in Part II.

3. Configuration Work

"Present-day IT infrastructure, "the real ubicomp," is a massive noncentralized agglomeration of the devices, connectivity and electricity means, applications, services, and interfaces, as well as material objects such as cables and meeting rooms and support surfaces that have emerged almost anarchistically, without a recognized set of guiding principles."

— Antti Oulasvirta [135]

In this chapter, I introduce the concept of ***configuration work*** by discussing and analyzing a range of configuration problems and challenges in distributed interaction. Section 3.1 first introduces the general configuration problems in distributed interaction. Section 3.2 and 3.3 provide an overview of configuration problems in knowledge work and in clinical work. In these sections, I summarize findings from our empirical work described in papers **C1**, **C3**, **C4**, **C6** and **C7** and relate their main findings to previous literature. Based on these analyses of our own work and previous studies, I define *configuration work* in Section 3.4 to describe the overhead of managing activities, devices and resources in distributed interaction.

3.1 Overview

In the last decade we have observed two major trends in end-user computing. First, there has been a widespread introduction of a whole new set of computing devices, ranging from mobile portable devices such as tablets and smartphones, to modern versions of the traditional desktop computer, such as hybrid touch-enabled laptops, large interactive displays and recently even a new set of wearable computers, such as the smartwatch and head mounted displays. Second, connected to this device multiplicity, resources and services are increasingly moving from traditional local storage to a ubiquitous cloud. The focus has shifted from individual devices to an ecology of interconnected devices that provides universal, shared and social access to users' personal information cloud [123]. Since most of these devices nowadays overlap in the type of functionality or tasks they support, users can hand pick which devices to use for what task, based on the *functional*, *interaction* and *social requirements* of the activity. Because of this, setups in which users are engaged with multiple devices at the same time are becoming more common. The user-device-resource mapping is changing from a one-to-one to a many-to-many paradigm in which complex device ecologies are constructed and shared between users to access a collaborative distributed information space. We refer to this computing paradigm as *distributed interaction*, in which *interaction with digital information is distributed over multiple people, multiple environments and multiple digital devices*. The setup of these four aspects changes dynamically over time and is in an ad hoc way related to the activities people perform.

Despite the many advantages of distributed interaction, it also poses users with a number of challenges. Although there are many tools available to support a wide range of multi-device scenarios, studies [51, 93, 103, 136, 156] show that users encounter a range of fundamental problems, such as lack of control, context, transparency and intelligibility when interacting with multiple devices and their containing resources. In essence, employing multiple devices to perform a particular task requires users to put a significant effort in what I call *configuration work*, the effort required to *set up*, *manage*, *communicate*, *understand* and *use* information, applications and services, which are distributed across several devices and people. Within distributed interaction I identify three distinct overlapping *problem spaces* that introduce configuration work:

1. **Resources** – *What resources are people using?*
   Managing resources (such as files and applications), services, communication and collaboration processes across multiple devices, owned by multiple users.
2. **Devices** – *What devices are people using?*
   Aggregating and pairing multiple devices into one seamless space. Awareness and understanding of the role of each device in a multi-device setup.
3. **Activities** – *What tasks or activities are people doing?*
   Managing one coherent work task or activity across several connected devices and people.

To get a better understanding of how these problems related to activities, resources and devices are manifested in both the office and clinical work setting, the next sections discuss configuration problems in office and clinical work using these three concepts.

3.2 Configuration Problems in Office Work

In this section, I take a general perspective on the *configuration problems* by summarizing and analyzing our observations and studies. These studies include a survey with 145 participants [**C1**], observational studies of a multidisciplinary software development team [**C1**], a quick and dirty field study of desk usage [**C3**] and a literature review [**C4**]. I relate these studies to previous studies and observations around multi-device workflows in individual and collaborative work. The results of the analysis are summarized in Table 3.1.

Type	***Problems***
Resource	- *Synchronization work* across devices - *Lack of transparent mechanisms* to distribute resources in the cloud - Users perform *workaround* using email and ftp - *Resources* are *locked* in devices or file types - *Curation work* to handle and archive files
Device Ecology	- *Moving resources* between devices - Handing off *use context* between devices - Handling *changing configurations* of devices - No support for *device roles* given by users
Activities	- Organize work into *higher level units* or tasks - Handling changes in *device context* when performing same task - *Cross-device task allocation* or connecting applications on devices - *Collaborative* tasks including multiple users and devices

Table 3.1: Aspects of the configuration problems in knowledge work.

3.2.1 Resource Distribution

Our studies showed that many people struggle with handling files and information that are shared between different users and devices. Although many tools attempt to support users in automatically moving resources, such as files and other data, between different devices, these tools often lack *control*, *transparency* and *integration*, making it remarkably hard to move information between devices. These tools are valuable for the technical distribution of information, but often do not immediately communicate which other users that have access to the shared data, nor whether the device can actually meaningful consume the data [156]. Managing and accessing information across different devices is still a significant problem [51]. To deal with this resource distribution problem, users employ a wide range of strategies to perform manual *synchronization work*. For overview, people use a combination of storage solutions that include (i) external storage, such as USB drives, SD cards or external hard disks, (ii) local storage, on the hard disk of the device, (iii) application storage, in which data is stored within application servers (e.g., email) and (iv) cloud storage or external services that provide storage or backup facilities [51, 156]. These different storage mechanisms are often used to separate working resources from older archived resources. Similar to our observations, one study [156] reported that active ongoing work was kept in open windows, applications and local storage, while recently closed projects where kept on cloud storage, and irrelevant old projects where archived (or disappeared) on hard disks or a back-up server.

Despite advances in cross-device interaction techniques (described in Section 2.1.3), transferring information between devices is still a hard problem that requires a combination of tools and workarounds. Although cloud storage is often perceived as a good solution for cross-device information management, it essentially moves the *curation work* of finding, organizing and relating files from the local file system to an even bigger cloud storage that merges data from different users and devices into one large online file system. Furthermore, users do not always trust automatic file sharing. They often find it difficult to understand what actions are applied to their information, and how they can reverse or undo these actions [51], thus, pointing at a lack of understanding about the functioning of the data synchronization. Users also point to other related issues such as reliability and privacy around their data [156]. Although cloud storage facilitates some technical challenges in distributing files and information, the disembodiment of data also introduces a number of intelligibility issues. As described by Odom et al. [132], cloud infrastructures "*break "the rules" of how we understand* *possession of material things*", implying that we need to rethink how cloud data can be visualized, represented and associated to a specific person.

Another frequently used approach is to appropriate existing technologies for file sharing. Our survey showed that email is the most used collaborative tool. Email is the prime example of a tool that is widely leveraged, and perhaps even misused for both task management and file sharing, as "*people sending attachments sometimes go so far as to say that they "FTP the document to someone," which shows how e-mail and file transfer have now become blurred to the point of confusion*" [55]. Studies [51, 156] show that in multi-device environments, email is used even more for file and resource exchange. However, this introduces a number of problems related to scale. The size of files that can be sent as attachment is limited, meaning that the approach breaks down in specific scenarios, causing the user to find a backup strategy. Additionally, making heavy use of email as file repository causes increasing email overload [64, 156], producing "*drawbacks such as cluttered inboxes and inefficient document transfer*" [156]. In essence, the widespread appropriation of email for collaborative work, points to a fundamental problem in supporting distributed workflows. As concluded by one of our participants: "*there is a lack of [a] cross-platform standardized way to share files and work*".

On smartphones and tablets, data are often owned by a specific application. Several participants in our survey mentioned that "*most devices have their own propriety sync[-chronization] software and rarely can you get them to effectively work together*". Moving them to other devices requires a user to first copy that file from that specific application to a shared repository such as Dropbox. Then, users simply synchronize the entire information hierarchy with all connected devices and browse that hierarchy to find the correct information. An attempt by existing tools to ameliorate this problem is to provide a simple view of the data, and only download information on demand. This approach however creates situations in which information is not accessible at all: when a device does not have an active internet connection, the data can simply not be downloaded. Specially in cases where devices are used *in concert*, this problem would break the operation consistency [156]. Even worse, some specialized devices, such as e-readers, allow annotation of books and documents, but lack a standard connectivity and sharing model, implying that information is trapped in the device. When sharing information, it is also often unclear to what degree a device can visualize and consume a specific resource [156]. Because resources and files are in essence pure data, they have to be analyzed and visualized by a local application or service. If this local *interpreter* is not installed, used or even available, the resource is not usable. Although many file types are standardized nowadays, there are still types of projects, documents or resources that can only be used on specific devices. There is a clear need for an additional control layer on top of these technical sharing and file distribution infrastructures that make these underlying processes more visible to the end-user.

3.2.2 Device Usage

Users on average own between three and six computing devices [51, 85, 156], including desktop computers, laptops, smartphones, tablets, game systems and e-readers. With the inclusion of additional devices, such as cameras, music players and smartwatches, this number can increase up to 10 devices per user [92]. Our survey showed that on average, participants owned between 1 and 9 devices (μ= *2.70*; σ= *1.59*) that they *actively use*. The *primary device*, meaning the device they use most, of 65.52% of participants was the notebook, while other devices (desktop computer 31.75%, tablet 0.69% and other 2.07%) are less popular as main device. Respondents spend between 1 and 16 hours a day using their devices (μ= *5,11*; σ= *3,13*). Device multiplicity provides users with the opportunity to choose the appropriate device based on its function, form factor, input and output bandwidth and available interaction techniques. Despite the fact that using multiple devices as information multiplexers has become a standard practice, devices are still designed for single-user/single-device user experience and not as part of a larger ecosystem [175]. This implies that devices are not aware of each other or their capabilities, unless they are equipped with special sensors [8]. As discussed earlier, basic operations such as moving files and resources are therefore cumbersome, as they require multiple steps and interactions with all devices in use [51].

Devices are chosen by users based on a number of criteria, including (i) form factor, (ii) task and (iii) device capabilities. First, the form factor of a device is an import motivator to switch to a different device. For mobile or stationary scenarios, users will employ different device configurations that leverage the situation and context. The physical design, such as display size, weight, portability and orientation, and affordances [129] of the device influence the choice of device [51]. E.g., when sitting at a desk, users will most likely use a combination of their smartphone with a traditional desktop interface computer. However, when at night they are reading documents in the couch, they will more likely use a tablet, or perhaps a smaller laptop. Second, the task will also greatly influence what device is used. When users, e.g., receive an email on a secondary device such as a smartphone that requires them to do substantial work, they will often switch to a device with better support for that particular task. Similarly, many users employ a tablet computer, rather than their traditional laptop, for active reading, note taking during meetings or sketching. Users also employ portable devices for on-site data collection [156]. Some modern hybrid touch-enable laptop device leverage this issue by allowing users to detach and use the screen as a tablet, while still being able to dock it into a keyboard for normal desktop use. Third, the device capabilities such as interaction techniques but also the available software will cause users to switch to a secondary device. E.g., reading a book on a high resolution full screen tablet using touch to scroll and zoom feels much closer to reading a physical book than using a laptop or phone. Similarly, using a CAD tool on a phone or tablet is simply unfeasible. Our rapid field study showed that many participants used multiple devices, such as tablets, notebooks, phones and also peripheral displays in their office setting, and that especially tablets have become a popular addition to the desktop computer or notebook. Furthermore, Figure 3.1 shows how desks are often used as a *spatial container* to organize devices and other non-digital artifacts.

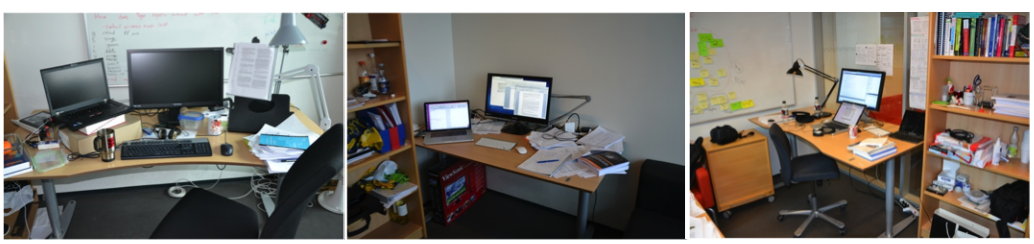


Figure 3.1: A set of desks used by office workers to spatially organize their devices.

Devices are used in different configurations and setups, based on the task and people involved. Device configurations are the "*set of devices and non-computational supported artifacts used in a situation, differentiating the active subset (those used at the moment) and the passive subset (those available in the room but not used)*" [136]. Throughout the work day, users actively manage the device configurations to handle ad hoc changes, caused by interruptions or spontaneous events, but also anticipate future use of the configurations. One study even reported that users switch device configurations in intervals smaller than 5 minutes [136]. However, incorporating and pairing new devices in ongoing work creates a device setup overhead, which influences and determines whether a device is used at all [136]. Maintaining workflows across different devices is hard, as they are not designed for parallelism [156]. To allow multiple devices to form one seamless distributed workspace, devices would benefit from mutual awareness about the information they contain, including their location and proximity, sensors and input capabilities [118]. This would also allow for more advanced interaction techniques such as cross-device drag and drop or push and pull information between devices [45, 79, 146].

Users frequently consider their devices as being either the *primary or master device* or being a *secondary or slave device* [92]. Most applications running on these devices, however, do not represent or incorporate this notion of a *device role*. Specially, in the recent shift of mobile devices from supporting specific tasks to becoming full information accessors, the changing role of these devices can play an important part in cross-device interaction. If devices can be aware of their role and use pattern, they could better facilitate cross-device application and resource management [51]. One example of such role-based functionality, is the ability of a laptop (used as primary device) to send SMS's from a desktop interface over a connected smartphone (attached as secondary device) [136]. Finally, Santos and Wigdor analyzed parallel device usage in distributed workspaces and found four different role patterns [156]. These patterns point to sophisticated strategies employed by users to handle information across devices by appointing different roles to their devices depending on the work context.

General	**%**	**Work**	**%**	**Cross Device**	**%**
Internet	97.24%	Communication	88.28%	Web Surfing	88.97%
Work	91.03%	Office tasks	83.45%	Email	86.90%
Social	76.55%	Research	53.10%	Calendar	55.86%
Multimedia	69.66%	Software	36.55%	Multimedia	51.03%
Accounting	58.62%	Accounting	28.28%	Reading	44.83%
Writing	42.76%	Design	23.45%	Communication	44.14%
Games	31.03%	3D Modeling	2.76%		

Table 3.2: An extended overview of tasks performed on the primary device for general and work purpose, as reported in our survey with 145 participants [C1]. The last column presents an overview of tasks performed on multiple devices.

3.2.3 Activity and Task Management

Our survey showed that the desktop or notebook is often used for the primary activity while mobile devices are used as peripheral devices that are configured to create interruptions (such as calls, emails and messages) or even operate as a two-way communication device, that might change or influence the work on the primary device. Table 3.2 provides an extended overview of the different tasks and activities performed using the primary and other devices, as reported in our survey [C1]. For general purposes the main device is primarily used for internet (97.24%) but also for social networking (76.55%), multimedia (69.66%) and games (31.03%). Interestingly, 91.03% of participants claimed to use their main device *also* for work. Tasks performed as part of work were communication with colleagues (88.28%), office tasks (83.45%), research (53.10%) and to a lesser extent software development (36.55%), design and 3D modeling (26.21%) and accounting (28.28%), thus, reflecting on a broad range of different knowledge tasks. For multiple devices, 88.97% of participants reported browsing the web on all devices. Email is a second frequently used cross-device task (86.90%). Other tasks included reading books, articles or news (44.83%), calendar management (55.86%) and more general communication (44.14%) including instant messaging and Skype.

This fragmentation and overlap causes users to migrate information or context from one device to another device. In highly mobile situations, users might want to use their mobile devices to perform a task or interact with data, but when returning to an office, that mobile configuration might not be ideal for continuing the task as better devices are available. In stationary scenarios, users might employ a secondary device for peripheral or background activities as described by Santosa and Wigdor, who observed that users "*valued having a hand-held device dedicated to peripheral or support activities, despite drawbacks*" [156]. This is in line with the findings of Grudin [77] who found that users with multiple monitors use their second monitor for secondary or peripheral tasks that are often used in direct support of the primary tasks. Furthermore, multiple devices are sometimes even used simultaneous for one activity. Examples of this include scenarios in which one device does not have all the functions or data needed, or when users reserve one device for, e.g., personal activity and the others for their work [136]. However, current devices are not designed with "*migration in mind*" [96, 136]. This holds for information applications and resources, but perhaps even more for communication and collaboration tools. Since many secondary devices are used explicitly for these types of communication tools, moving these to another device is either impossible or creates a breakdown of work resulting in significant configuration work.

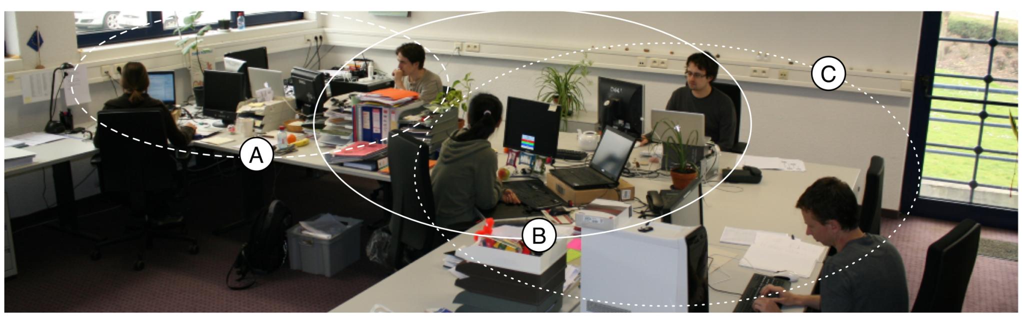


Figure 3.2: A group of knowledge workers collaborate within a number of overlapping activities.

Modern knowledge work is highly collaborative [81] and many knowledge workers collaborate on a number of simultaneous projects that include a subset of other users and devices (as seen in Figure 3.2). Within these subgroups, resources are often shared among multiple people and devices in order to coordinate and articulate work processes among the collaborators. Activities or tasks are fragmented across these users based on the individual role of each participant. In the example of Figure 3.2, the team consists of three software developers, a historian and a designer. Although each of them collaborates on the same projects, they all use a specific subset of resources, services and devices for their part of the work. This delineation is, however, not always very clear, implying that a subset of resources are used by multiple people but also that there exists a set of interdependencies between individual users [142]. This interdependent work "*involves a number of secondary activities of mediating and controlling these cooperative relationships. Tasks have to be allocated to different members of the cooperative work arrangement: which worker has to do what, where, when?*" [161]. The distribution of these activities have to be articulated to project members [166]. Furthermore, with the increasing mobility of workers, they often lack the social, technical and informal ad hoc resources needed to articulate this activity distribution. This forces them to perform *mobilization work* [140], which is the meta work that has to be performed to articulate work activities while being mobile.

Distributing tasks across devices creates a distributed workflow that can span multiple people and devices [51]. One of the central problems in multi-device environments is that current devices are designed with an explicit focus on applications and files, and not on the task or activity people are using the devices for [51]. Studies [10, 50, 74] have shown that users often organize their work and resources into thematically higher level constructions that are a representation of the ongoing work. As reported by Gonzalez et al.,"*throughout their day, individuals are constantly moving from one topic to another and managing information streams from a myriad of sources*" [74]. Additionally, to achieve overview about ongoing tasks and coordinate information with other people, users perform *meta work*. This meta work is needed to structure and organize their workspaces for daily work. Because the overhead of manually switching between multiple tasks is significant [50], tools and software to manage and control this meta work are needed. Users' tasks are, thus, not confined or limited to one device and user, but rather span across multiple devices that are setup in different configurations, based on the active work condition and task [51]. This greatly amplifies the project fragmentation problem [28] as projects are now not only distributed over applications, but also devices and users. There are currently no tools to directly support to migration and sharing of tasks across devices. As described by Oulasvirta: "*users essentially need new and more efficient ways to interoperate devices,* *plan action in the face of "seams", understand and manage technological complexity, plug their data into other devices, and align use fluently with everyday activities*" [135].

In "*Beyond the desktop metaphor in seven dimensions*" [100, p. 350], Moran and Zhai summarized that there is a shift from the classic application and file-centric view to a multidimensional perspective in which devices are portals into a shared activity-centric information space that allows for (i) diverse representation on data, (ii) device multiplicity by providing *transformational* interfaces, (iii) new interactions and modalities to unlock the full potential of mobile and distributed interaction, and (iv) incorporate intrinsic support for social interactions and collaborative workflows. Providing more explicit support to seamlessly move or fragment users' tasks across multiple devices will allow users to better appropriate the different interactive input and output capabilities of each device that is used to work on the ongoing task or activity. As concluded by Dearman and Pierce, "*support for users' activities should be neither application- nor device-centric*" [51].

3.3 Configuration Problems in Clinical Work

The second domain discussed in this thesis, is clinical work. Although clinical work in hospitals encapsulates many of the properties, and by extension also the problems of multi-device knowledge work, the clinical domain introduces a number of additional challenges. In this section, I therefore describe configuration problems (summarized in Table 3.3) that are part of clinical work. Similar to the previous section, I relate the field studies described in paper **C6** and **C7** to prior studies and observations. The field study applied observations, contextual inquiries and interviews over a period of several weeks and was conducted to get a better understanding of work at patient wards. Details on the study are elaborated in paper **C7**.

3.3.1 Resources and Applications

Information and resource management related to patient care in medical work in hospitals is organized and centered around medical records. These records are legal documents containing detailed information about the patients' personal

Type	*Problems*
Resource	- Hard to access centralized *digital record* - Handling *parallel paper* documentation - Files and resources are kept in applications - Resources and information kept on whiteboards
Device Ecology	- Using and sharing *public devices* - *Collaborating* through devices - Including *non-digital artifacts* into digital workflow - No support for *connected* device ecologies
Activities	- Create computational *representations of activities* - *Mobility work* when moving between different locations - *Coordination* and *communication* across activities

Table 3.3: Aspects of the configuration problems in clinical work.

information, medical history and ongoing treatment. Within clinical work, the medical record is used as a central organizational and coordinative repository that allows clinicians to share, communicate and manage complex medical procedures during day to day work in the hospital [27]. To increase the efficiency, safety and quality in care, the Western world is investing a significant amount of resources into digitizing processes within health care, with a special focus on creating an integrated unified Electronic Health Record (EHR) [66]. EHRs have a number of important advantages over traditional paper documentation including higher quality of health care, efficiency in use, and a higher level of patient safety [43]. Often, EHRs encapsulate a number of digital resources, such as files, database entries and even applications or services. These are mainly Health Information Systems (HISs) such as a Radiology Information System (RIS), an Electronic Medication System (EMS), a Patient Administration System (PAS), a Blood Bank System (BBS) and many others. These resources and applications are important for daily operation of a hospital, as they are primarily used to create new medical information. Clinicians often access these HISs through a system portal or EHR, which contains an overview and links to all applications and information.

Next to the EHR, many hospitals also use paper medical records (PMRs) as part of the clinical workflow [173, 192]. During the study, we observed how the PMR played a central coordinative role in daily hospital work. Although the de-

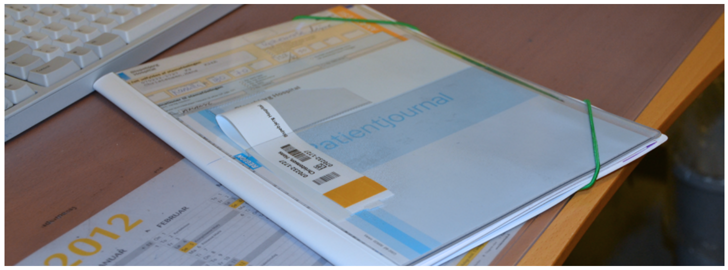


Figure 3.3: A PMR consists of a plastic folder that is marked with the name and ID of the patient. The PMR provides color-coded sections for nurse notes, treatment history or other forms and observations.

tailed procedures differ between different departments of the hospital, the main structure is standardized across the entire hospital in a unique PMR. Figure 3.3 depicts the PMR, which consists of a plastic folder that contains a bar code and sticker with the name and ID of the patient. The record has color-coded sections for a range of different types of documentation, including patient information, the narrative treatment record (or 'continuation'), nursing documentation, observations, test results (e.g., radiology examinations) and messages from other medical professionals. It is a legal requirement that this record is at all times present at the ward that is currently treating the patient.

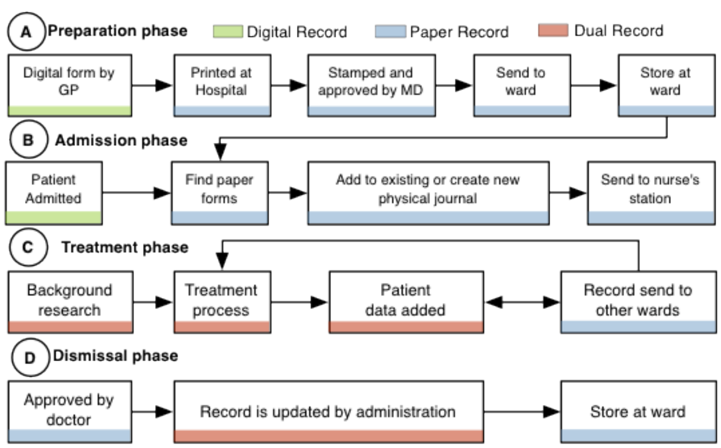


Figure 3.4: The life cycle of the PMR consists of a (A) preparation phase, (B) admission phase, (C) treatment phase and (D) dismissal phase. (GP: general practitioner; MD: medical doctor)

Figure 3.4 shows the four main processes involving the medical record. The color code indicates whether only the PMR (blue), only the EHR (green) or both records (red) are used for that part of the process. The patient is referred to the hospital (e.g., for surgery) by the general practitioner (GP), who fills in an online form. This form is printed by the administration of the hospital when setting up an appointment for the patient. The form is stamped and approved by a doctor and sent to the ward responsible for the patient (Figure 3.4A – *preparation phase*). In the hospital, the general workflow surrounding the PMR is primarily managed by the ward secretaries and the nurses. When a patient is admitted to the hospital, the ward secretary locates the PMR. Most patients are readmitted to the same department and this 'home' department, hence, physically stores the PMR in the storage room. However, if a patient was previously treated at another department, locating the PMR can be a rather cumbersome process. Once located, the referral letter (e.g., from the GP) is added to the PMR. If a new patient without PMR is admitted to the ward, one is created. The record is then sent to the nursing station the day before (or on the morning) the patient arrives. During the morning conference between the doctors and nurses, the record is used to prepare the arrival of the patient and to plan the treatment (Figure 3.4B – *admission phase*). Once the daily treatment and care of the patient has ended (Figure 3.4C – *treatment phase*), the medical continuation is updated by a ward secretary while nurses update the nursing record, the medicine scheme and add relevant examination results to the record (Figure 3.4D – *dismissal phase*). Once the patient is discharged from the department, the PMR is finalized and stored at the ward. This implies that hundreds of archived records are at the department.

This double medical record introduces a number of configuration problems related to finding, using, and managing both the paper and electronic representation of the patient record. First, the usage of both electronic and paper records causes synchronization problems between both representations [176], forcing clinicians to deal with the paper and digital information simultaneously. Since digital information is often only available through desktop computers, it requires clinicians to sit at a desk when interacting with patient data (Figure 3.5). Second, since many hospitals require the use of a unique paper record, it is often transferred between different departments and wards as patients and clinicians move throughout the hospital. This causes paper records to be physically misplaced in the ward or even lost between departments.

Figure 3.5: Clinicians often use both digital applications and resources as well as printed paper-based records.

3.3.2 Devices and Tools

This widespread use of paper-based *working documents* can be framed within a larger *web of coordinative artifacts* [20]. Clinicians use a large amount of both non-digital artifacts and devices to coordinate and share information as part of their daily work activities. To maintain a close and direct feedback loop between all involved clinicians, patient information is often managed on large (sometimes interactive) whiteboards [32]. These boards include initials of the responsible nurses, a brief summary and future outline of the treatment, the room in which the patient is situated and the responsible and attending physician. This information is often marked with specific colors or magnets to, e.g., indicate which nurse or doctor has made the changes, or is taking care of the patient. Furthermore, using symbols, nurses use the board to coordinate the ad hoc order of activities. E.g., nurses mark a specific patient with a diagonal line to indicate that the patient needs to be attended by a doctor. After checking up with the patient, the doctor will add a second diagonal line, creating a cross, indicating that the activity was completed.

Clinicians often move from one location to another while interacting with both mobile and stationary tools and devices, such as desktop computers, large

Figure 3.6: Clinicians use a range of different devices at a medical ward in hospitals, including (a) desktop monitors, (b) large interactive displays and (c) mobile device.

(interactive) whiteboards and mobile devices [22] (Figure 3.6). Large static display that are situated in different wards are, e.g., used to visualize the surgery schedule in an effort to create a better shared overview of the planning [23]. Large interactive whiteboards were introduced to replace the traditional whiteboard in an effort to make them more stable and uniform across different wards [83]. In some hospitals, mobile devices, such as PDAs, are used to increase patient safety by bringing redundant information closer to the patients. For example, some PDAs were equipped with a bar code scanner, so that the administering nurse could verify whether they were giving the patient the correct medication. Most of these devices are also public, meaning that any clinician can pick them up and use them during a work day. With exception of some workstations in offices, there is no explicit ownership over any device at a patient ward. However, next to this ecology of digital devices and tools, clinicians also still use many non-digital artifacts. The PMR is a prime example of a non-digital artifact that still plays an important and central role in daily clinical work [67].

Clinicians use a wide range of tools to coordinate patient information with other clinicians. Despite the ubiquity and importance of these tools for the information flow in the hospital, most of these tools are not designed, tailored or suited for this nomadic use, resulting in a mismatch between the functionality of the system and the actual work done by the clinicians. There is little support for the ongoing collaborative context, situation or specific role of the device and non-digital artifacts. Artifacts, such as the PMR, are completely disconnected and obsoleted from the rest of the digital workflow. Fundamentally, all these different systems, non-digital artifacts and devices are intrinsically disconnected from each other, forcing clinicians to manually *reconfigure* the active work setting according to the situation. This results in work interruptions, information fragmentation and a disconnected workflow crossing several information resources. As concluded by Morán et al. [122], "*these working conditions call for a new computing paradigm for hospital work, one that supports collaboration and coordination, mobility, seamless interaction with heterogeneous devices, and frequent task switching*".

3.3.3 Activity and Task Management

Clinical work is highly organized into a set of "*well-defined tasks or activities that must be carried out and are known and agreed upon by all clinicians*" [48]. Activities in medical work can be subdivided into (i) patient care and clinical assessment, (ii) coordination and communication with other clinicians and (iii) administrative work [122]. Patient care involves all processes that explicitly involves interaction with the patient. Coordination work revolves around articulating the patient case and plans with other clinicians through direct communication or shared artifacts. Administrative work encompasses knowledge work processes and includes examples such as patient information curating, ordering special tests or procedures or simply inspecting the medical journal. These three types of work are not explicitly separated but rather overlap during daily clinical work. Clinical work is temporally organized as many activities have to be executed in a specific order according to a specific plan [17]. Despite the highly planned nature of clinical work, the actual daily work activities are highly volatile and ad hoc in nature. As concluded by Bardram, "*on the one hand, due to the contingencies of the concrete work situation work has an ad hoc nature. Plans are not the generative mechanisms of work, but are merely used to reflect on work, before or after. On the other hand, we find that plans, as more or less formal representations, play a fundamental role in almost any organization by giving order to work and thereby they effectively help getting the work done*" [16].

A patient ward (as depicted in Figure 3.7) consists of a number of rooms and spaces allocated for patients and clinicians. Patient rooms, living- and bathrooms and other communal areas are freely accessible by patients. Rooms such as the meeting rooms, offices, nurse stations and medication storage rooms are used for clinical work. A classic example of clinical activity on this type of patient ward is the daily *ward round* [124]. During this daily procedure, one or two doctors,

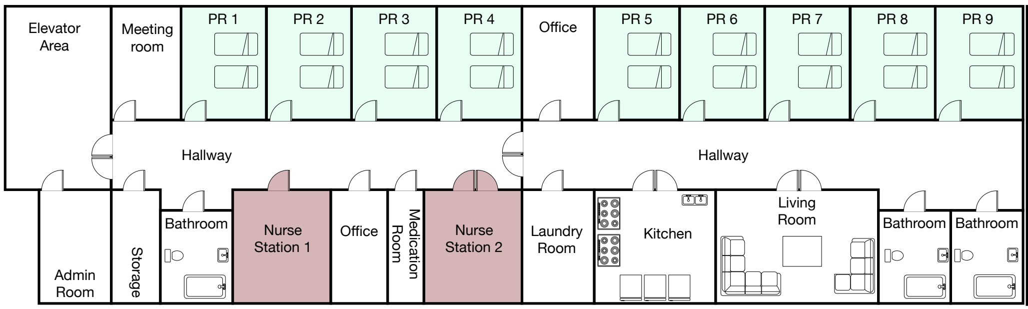


Figure 3.7: The physical layout of one of the wards from our field study. A typical ward consists of an administration desk, a number of patient rooms (PR), stations for the nurses, offices for the doctors, storage and medication rooms for medical equipment and finally bath- and living rooms for patients.

accompanied by a number of nurses, visit all patients at the ward to check up on their status, and discuss the progress of the medical treatment with the patient. During this round, the team of clinicians move from one room to another, often carrying a number of documents, paper records and other information needed to assess and discuss the patient case. Doctors are also frequently allocated to multiple bed wards. This means that throughout their shift, they are responsible for patients that are distributed among different wards that are physically disconnected from each other. Nurses at the ward also continuously move between the nurse station, the physical location where information is synchronized and patient care is organized, and the patient rooms, where the treatment is provided and the care is executed. Over the years, highly specialized medical treatments, such as different types of surgery, but also specialized test, such as MRI or CT-scans, have been centralized in specific parts of the hospitals. This implies that patients are often moved between different departments whenever they require any of these specialized tests and treatments. The workflow of clinicians in a patient ward can be described as *nomadic*: throughout the workday both clinicians, and their tools and equipment, frequently move between different areas of the same ward, different departments and even different buildings. Similar to the earlier discussed *mobilization work* [140], the highly mobile character of clinical work introduces *mobility work* [19], the spatial extension of articulation work [166] and the overhead of aligning and organizing people, artifacts and information during work.

Clinical work is also highly interdependent on a large amount of clinicians with different roles and specializations [145]. Activities are shared among these clinicians but "*individuals engaged in different activities will have different perspectives on the same information*" [145]. Therefore, collaboration and communication are an important and intrinsic part of daily work in the hospital. During clinical work, clinicians almost constantly communicate relevant information with other clinicians and departments, and coordinate complex patient treatment procedures and changes in patient treatment using *common information spaces* [9]. This highly collaborative aspect of clinical work causes clinicians to "*receive multiple interruptions, either face-to-face from colleagues, or through the paging and telephone systems*" [49]. This forces clinicians to frequently stop their ongoing activity and switch their attention to another one. Studies show that because of these interruptions, "*clinicians reduce the time they spend on clinical tasks [...], and may delay or fail to return to a significant portion of interrupted tasks*" [138]. However, interruptions are necessary as clinicians interact with each other to coordinate work, align priorities and reorient their focus [75]. The face to face communication processes are important as they help clinicians to create a shared understanding of their workflow and articulate their activities. The problem lies in finding a way to support this type of ad hoc collaborative activities that are distributed among different people and artifacts, while reducing the overhead of switching between these different work contexts.

3.4 Configuration Work

All these aforementioned problems introduce *configuration work*, which is the time, effort and work required to *set up*, *manage*, *communicate*, *understand* and *use* information, applications and services, that are distributed across several devices and people. It is work needed to configure the environment and setting in order to perform tasks or activities.

3.4.1 Processes of Configuration Work

Depending on the setup of the distributed interactive systems, *configuration work* can be composed of a number of defined subprocesses in any arbitrary

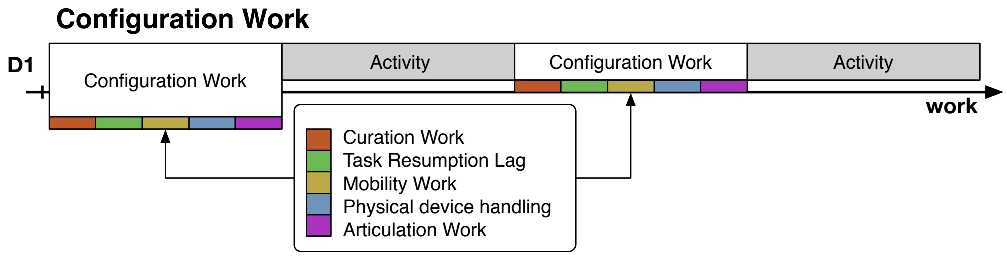


Figure 3.8: Configuration work in distributed interaction can be composed of curation work, task interruption and resumption lag, mobility work, physical handling of the devices, and articulation work.

order, duration or frequency (as depicted in Figure 3.8). These processes include curation work, task interruption and resumption lag, mobility work, physical handling of the devices, and articulation work. The different processes are highly interrelated and have a direct influence on each other. The cumulative effect of these interactions are the cause of configuration work.

Curation Work

Curation work (in literature often also referred to as metawork) includes setting up file-, task-, contact-, and communication tools within a specific device. It is the "*work that enables work*" [155], and includes processes to set up and prepare a specific device for actual work. Curation work are activities needed to reflect on ongoing work and prepare the system for new tasks. This curation work is generally performed after the completion of other activities [74]. Within distributed interaction, curation work remains an important aspect of multitasking as it linearly expands with every other device in use.

Task Interruption and Resumption Lag

When users switch between different ongoing tasks, there exists a delay associated with handling, understanding and processing the interruption that cause the task switch and work required to restore the resources and content needed to continue working on the new task. Interruption lag refers to the process of cleaning up the workspace of the devices and "*rehearse the primary task problem representation*" [154]. This interruption lag leads up to a task switch, after which users recall the new primary task and set up the environment using curation work to continue work on the new task [154].

Mobility work

As "*mobile workers are often impoverished in terms of social, informational and technical resources*" [140], they have to perform additional work to mitigate these limitations. Bardram and Bossen [19] describe this problem as the effort of moving people, devices and non-digital artifacts as part of creating an environment that allows users to perform a task. Within distributed interaction, mobility is a fundamental intrinsic aspect of work, implying that considerable amount of mobility work is needed to achieve a workable environment.

Physical Device Handling

The physical handling of devices to create a workable device configuration, is a fundamental problem in multi-device environments [136]. Creating, maintaining and organizing physical device configurations, meaning being able to place, power, configure and operate devices in a specific environment, is a central precondition for the performance of tasks or any other work on any type of computer device. Without a functioning, ergonomically organized device setup, work is impossible.

Articulation Work

Articulation work (detailed in [161, 166]) is "*a kind of supra-type of work in any division of labor, done by the various actors*" [167]. In order to handle interdependencies between people and cooperate on distributed activities, workers need to perform a set of secondary activities that involves how, when, why and where these shared distributed activities are performed. "*Workers have to articulate (divide, allocate, coordinate, schedule, mesh, interrelate, etc.) their distributed individual activities*" [161]. Articulating what activities people are doing, is an important and intrinsic aspect of distributed interaction, and part of *configuration work*.

3.4.2 Multi-Device Use Patterns

Configuration work plays an important role in the overhead of managing multiple devices and artifacts at the same time. Based on the discussion of our empirical studies on managing devices, activities and resources in distributed interaction, I present 4 generic usage patterns that signify how multi-device configuration is done. These patterns are parallel use, alignment, fragmentation and replication.

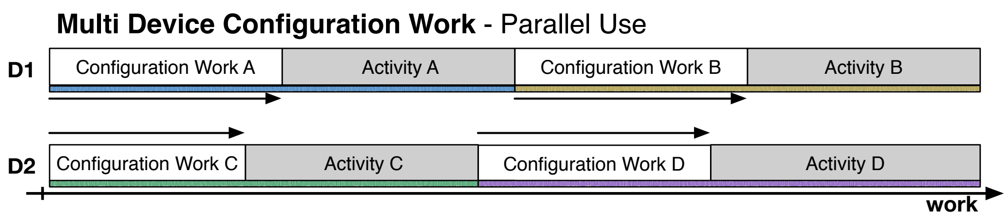


Figure 3.9: Parallel multi-device usage with no cross-device configuration work.

Parallel Use

Often, multiple devices are used in parallel by the same or different users running multiple activities. For these types of *parallel device usage*, the activities on each device are clearly delineated and do not overflow or interact with activities on other devices. As such, the configuration work for each of those activities is independent and confined within each device (Figure 3.9). As described earlier, an example of such a parallel device usage, is the case where a user employs different devices for both work and personal usage. In this case, one device is specifically focused on activities related to work, while the other devices are only used for personal purposes. There is no information overflow between devices and no cross-device configuration work is needed. A second example is in the case where the group of knowledge workers (Figure 3.2) are working on two parallel activities with no explicit overlap. During the study, we observed how users swapped between these disconnected parallel activities whiles others would work on shared activities.

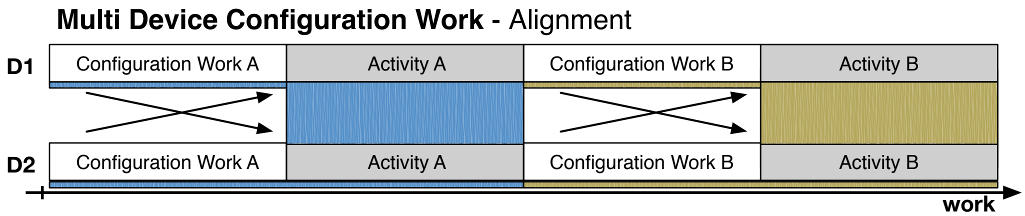


Figure 3.10: When devices are used in tandem, the configuration work needs to be aligned.

Alignment

When different devices (and non-digital artifacts, which I here include in the 'device' definition) have a distinct role or contain specific information, they are used in tandem. Take the example of the paper and digital patient record, that is distinctly used at the same time to get an overview of all information related to a patient case. In this situation, both devices (e.g., the personal computer and the paper folder) need to be configured in the correct way, aligned with the goal of the activity, to create an environment in which the task can be performed. If the paper record or digital record are not available, the shared structure is incomplete and the activity cannot be performed. A second observed example is a teleconference call between two team members. In order for both members to engage in the distributed communication activity, the configuration work of both instances need to be set up in context of the activity. The configuration work of both devices need to be aligned as they are intertwined (Figure 3.10). This means that configuration work of one device directly impacts the work for another device. The activity can only be performed if configuration work on all used devices is done.

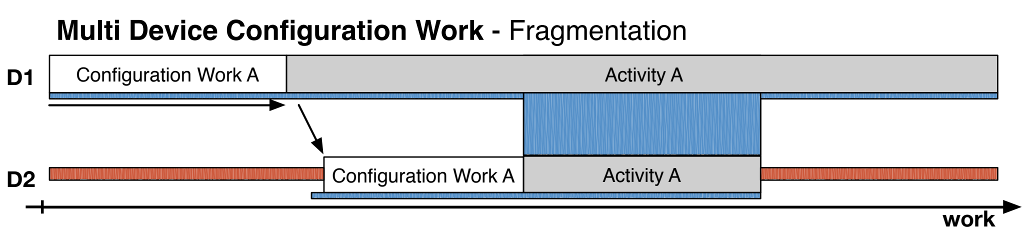


Figure 3.11: When activities are fragmented across devices, configuration work is needed to include the new devices.

Fragmentation

As reported in previous studies [51, 156] and observed in both the hospital and office case study, users often distribute tasks across multiple device in use. During, e.g., writing or work, users might employ a tablet to proofread the documents and use their phone to monitor emails or other IM tools. Even within a group of people, one activity might be fragmented into smaller pieces while keeping an active connection between the devices. Fragmenting activities across devices introduces additional configuration work, as the resources, data and applications need be communicated, set up and configured before they can be used (Figure 3.11). In the case of the multidisciplinary team, individual members would often share an entire project during a specific stage. When the designer, for example, was done creating initial designs, he would share the entire project with other members, so they could assess and use the designs for project reporting and software design. Because modern computing devices are not equipped with mechanism that allow for migration of an entire work context, each fragmentation includes a high degree of configuration work whenever the work across devices needs to be synchronized.

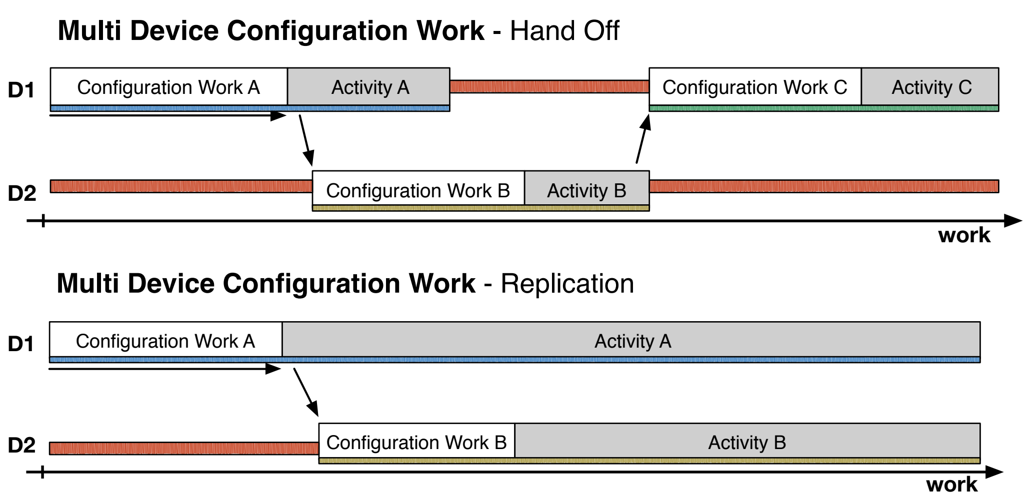


Figure 3.12: Activities can be handed off between devices owned by the same or different users. Replicating a work context on another device introduces configuration work.

Hand Off and Replication

Within teams, units of information are often handed off between different members. In a hospital setting, patient cases are passed along to different clinicians as part of the different processes in patient care. This can be done through physical non-digital artifacts, like the patient records that are sent with the patient to different wards and consultations. Similar, digital resources, such as lab tests or clinician assessments are handed off in the electronic record or by printing the document. Similarly, project teams often hand off project folders (including documents, contact information or other resources) between different team members when a project milestone is achieved. As in the fragmentation problem, handing off information induces configuration work, as there are no mechanisms to migrate work context or merge newer versions of that work context back into the previous version (Figure 3.12). Similar problems occur when users try to replicate a work context across multiple devices.

3.5 Summary

This chapter summarized and related our empirical work to prior studies and observations. After discussing the general configuration problems in distributed interaction in Section 3.1, I presented an in depth description of the configuration problems in knowledge work and in clinical work in Section 3.2 and 3.3. Finally, based on these empirical data, I provided a definition and description of the processes of *configuration work* in Section 3.4 to capture an overlapping concept that describes the overhead of managing activities, devices and resources in distributed interaction.

4. Activity Configurations

"The main benefits of continuing to import and develop theoretically based approaches into HCI is the construction of new accounts, frameworks, and concepts, which, in turn, have the potential for being developed into a more extensive design language that can be used in both research and design."
— ***Yvonne Rogers*** [150]

As observed in the empirical studies discussed in Chapter 3, configuration work is introduced when users *set up*, *manage*, *communicate*, *understand* and *use* information, applications and services that are distributed across several devices and people. Configuration work is, thus, introduced when users try to align and fit their artifacts to match the activity they are trying to perform. This chapter provides an overview of the theoretical and conceptual background of the work presented in this dissertation. Using Activity Theory, I re-analyze configuration work in distributed interaction using human activity as a unit of analysis and introduce artifacts and device ecologies into the Activity-Centric Computing paradigm.

Section 4.1 first briefly discusses the role of theory in the field of Human-Computer Interaction (HCI). Based on these reflections, Section 4.2 provides a general overview of Activity Theory, which is used to frame and reference Activity-Centric Computing in a theoretical background in Section 4.3. I then introduce *activity configuration* in Section 4.4 as a concept to describe computer-mediated activity across users and devices. Section 4.5 introduces a framework that describes how activity configurations can be distributed and shared across different users and devices. Finally, section 4.6 concludes this chapter by demonstrating how the activity configuration framework can be used to construct device ecologies.

4.1 Why (Activity) Theory?

In the first wave of HCI, early theories applied cognitive science as a method to analyze the interaction between humans and machines. Based on the early work by Neisser [128], who summarized that "*cognition is the act of knowing, and cognitive psychology is the study of all human activities related to knowledge*", Card et al. [40] introduced their *human processor model*, in which humans and machines were modeled as symmetric information processor units whose input and output processors where interconnected. Early on, this limited view on HCI was critiqued as a theoretical ground for analyzing and understanding the interaction between humans and computers [11, 170, 191]. Suchman [170] found that human actions are not planned and defined but rather ad hoc contextualized, flexible and situated as humans adapt to any situation based on their knowledge and experience. Situated action describes how human activities emerge from ad hoc moment to moment interactions between people and their environment. In her seminal book "*Plans and situated actions*", she concluded: "*to designate the alternative that ethnomethodology suggests more a reformulation of the problem of purposeful action, and a research programme, than an accomplished theory, I have introduced the term situated action*" [170]. Inspired by Garfinkel [115], Suchman introduced an ethnomethodological account on human-machine interaction *in the wild* that displaced the idea that humans could be modeled like a computer. Ethnomethodology is an anti-theoretical approach that rejects the idea of sociological theorizing [98], and, as Dourish described: it "*turned its analytic attention to the ways in which everyday social action was achieved.*" [53]. Using phemonology, Dourish extended this view on situated action to embodied interaction [52], in which he connected Ubiquitous Computing, Tangible User Interfaces (TUIs) and situated action to explain how users act through technology, not on technology. As part of this second wave of HCI, a number of post-cognitive theories (summarized by Rogers [150]), including Actor-Network theory [108], Activity Theory [34, 98, 112] and Distributed Cognition [86, 149] where introduced as theoretical accounts for HCI that encapsulated the complex analytical capabilities of the ethnomethodological situated action but also provides abstractions that allow for generalization and comparison within the theoretical framework. Although these theories differ on a philosophical level, they often portray similarities and it has even been suggested that Activity Theory and Distributed Cognition will keep informing each other and eventually might even merge into one theory for HCI [127].

As we are currently in the third wave of HCI, the focus has shifted from work to everyday experience. Computer devices have become, as Weiser predicted [185], ubiquitous and their use has blurred the boundaries between work, leisure and life itself. The focus shifted from purposeful work rational to aesthetics, emotion and experience, leaving us with an "*alternative agenda which focuses on designing UbiComp technologies for engaging user experiences*" [151]. This third paradigm, summarized by Harrison et al. [80] as *situated perspectives*, approaches interactions as phenomenologically situated activities that support situated action in the world. However, as argued by Bødker [35], many post-cognitive theories are still useful as analytical theoretical framing for the third generation of HCI. Within the third wave of HCI, users or workers need to become engaged people whose everyday life is part of design. Reflexive design prototypes need to play a role to explore operation, transparency and use, thus, operationalizing frameworks such as Mackay and Fayard's *triangulation approach* [117] that proposes a multi-perspective design approach. Finally, Bødker points to the rising importance of reconfigurability and tailorability, in which users are empowered to co-create, build and maintain components, configurations and mediators "*away from end user programming in isolation and towards configurations with multitudes of physical devices*" [35]. These aspects of HCI fit within the analytical power of these second generation theories, such as Activity Theory.

Kaptelinin and Nardi [98] argued that as community "*we need to compare, abstract and generalize*" to be able to communicate shared concepts [41]. Theory should both capture the important aspects of interactions, as done, e.g., in situated action, as well as be descriptive and generalizable to allow for comparison and development. We need a theory that allows us to meaningfully describe how humans act in the world, capturing both its context and its emergence; and more specifically, how humans act with technology. As argued by Kaptelinin and Nardi [98] such a theory can leverage design for (i) collaborative work using technology, (ii) mixed-reality environments that blend physical and digital data, (iii) more advanced activities, as suggested by the Ubiquitous Computing vision, and (iv) general human experience.

Compared to Distributed Cognition and Actor-Network theory, Activity Theory emphasizes an asymmetrical relation between human and material agency implying an intent or motive in human activity. Furthermore, this human intent is objectified and mediated by tools and signs that allows humans to act in the world. These two core assumptions of Activity Theory are a very important analytic match to the problems described in configuration work, which fundamentally revolve around creating, maintaining and sharing representations of these human intents across a stable device configuration. Although abstract in nature, Activity Theory can provide us with a conceptual framework to analyze, understand and design for computer-mediated human interaction. Furthermore, the descriptive power of Activity Theory provides us with a strong multidisciplinary framework of existing concepts, design artifacts and even technology. Relating back to Mackay and Fayard's *triangulation approach* [117], I consider Activity Theory a theoretical *perspective* that helps to inform and guide design and empirical study.

Within this line of argument, Activity-Centric computing is an example of a conceptual framework that originated from both empirical [3, 107, 130, 141] and activity theoretical research [10, 18, 98, 97, 178]. Both the empirical and theoretical perspectives point to the importance of shifting the unit of analysis of HCI to *human activity*. Within this thesis, I follow this approach and use Activity Theory, as introduced to the HCI community by Bødker [33] and summarized by Kaptelinin and Nardi [102], as a theoretical perspective on approaching configuration work in distributed interaction. This leads to a number of contributions in terms of new concepts, interactions and technologies within the Activity-Centric Computing paradigm.

4.2 Activity Theory

Activity Theory is a socio-cultural framework that describes and analyzes human beings and their sociocultural entities through the creation, development and processes of their activities in everyday circumstances [102]. In strong contrast to cognitive science approaches, Activity Theory states that the subject and object cannot be analyzed as separate components, but rather should be analyzed as part of one unit, human activity. Activity Theory has its foundations in the work by Lev Vygotsky on cultural-historical psychology [182], and was further developed and crystallized by Leontiev [112]. I here summarize the basic concept of Activity Theory and refer to the work of Kaptelinin and Nardi [98] for a full introduction to Activity Theory.

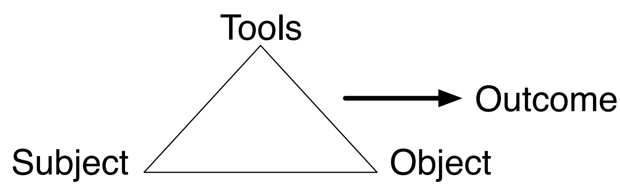


Figure 4.1: Human activity is a mediated interaction between the subject and object.

4.2.1 Activity

The basic unit of analysis within Activity Theory, is the *activity* (Figure 4.1). An activity is a set of purposeful actions of a (human or animal) subject in the world. It is a unit of life that is mediated by psychic reflections and real functions that orient the subject towards the objective world [111]. A human activity is the asymmetrical relation between the subject and the object that is mediated by both material and psychological tools. The asymmetrical relation is implied by the intrinsic *need* of human agency to act in the world as compared to material agency that may act but cannot experience human need or intention. The subject perceives the object world in terms of motives and goals and acts on the world through tools or instruments. This motive may be either material or ideal, thus, existing in perception or imagination [112]. Leontiev described activity as "*purposeful interaction of the subject with the world, a process in which mutual transformations of the poles of subject-object are accomplished*" [98]. This implies that no properties of the subject and object exist outside of the activity. Furthermore, activities are the source for development of both the subject and object. Therefore, the subject's properties are directly created, influenced and shaped by its interaction with the object world. Activity Theory describes that the human activity is directly influenced by sociocultural forces that produce and shape the human mind, implying the existence of an intrinsic dialectical interaction between the human and the world. In conclusion, an analysis of the mind is an analysis of the subject's act on the world.

4.2.2 Object-Orientedness

A human activity is always oriented towards their objects as everything humans do is connected to an underlying motivation or reason. This intentionality motivates humans to act on the object world. This world is highly organized and structured and provides affordances [6, 71] and resistances [102] that shape how humans act. Activities are differentiated based on their object [112]; human activities are objectified needs. The object is a central and necessary concept to understand human behavior. Although objects shape activities, they do not define them. Furthermore, objects can be either physical or intangible objects such as ideals or visions; they can be individual and collective; and are hence social and cultural, implying they are shaped by implicit and explicit rules. The object transforms the activity of the subject, which in turn can produce a reflection or image of the object. The dynamics between the subject and the object shapes human activities but also creates subjective phenomena related to the perception of the object.

4.2.3 The Hierarchical Structure of Activity

A human activity is defined as a mediated relation between the subject (S) and the object (O) that is driven by a motive that defines, shapes and structures the activity. The motive is an object that meets the need of the subject, implying that activities are caused by needs. Activities can be analyzed on three levels: activity, action and operation (Figure 4.2). The top level, activity, itself is a motive oriented structure that fulfills a human need or intention. Each activity can be decomposed into actions, which are smaller units that, although not directly connected to the motive of the activity, contribute to the activity. Actions form the conscious executions of individual objective goals that, combined, attain the motive of the activity. Before actions are performed, they are planned by subjects during an orientation step [106]. Actions consist of unconscious operations that are automatically performed within the structure and conditions of the action. Subjects are unaware of these operations and perform them as routine automated functions. Activities are why we are acting, actions are what we are acting and operations are how we are acting. The relation between actions and operations determines the skill of a subject for a particular activity. Through learning, conscious actions can be automatized into unconscious operations. However, operations that do not produce the desired effect can breakdown, forcing the subject to re-evaluate the goal and perform an action. These automatization and deautomatization processes point to transformations that can occur between the three levels of the activity hierarchy.

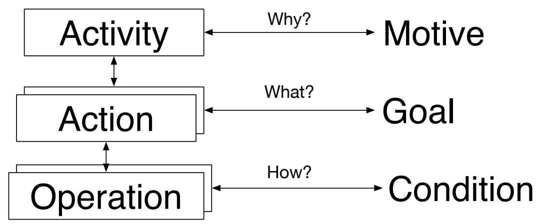


Figure 4.2: The levels of an activity consist of the activity itself which is driven by a motive, a number of actions which are conscious goal oriented processes that are themselves decomposed into automated unconscious operations that occur in specific conditions.

4.2.4 Internalization - Externalization

Activities are not separate units of analysis, but rather connected to the socio-cultural environment. First, human activities are distributed over both internal and external components. The interaction between the internal and the external activity explains how humans are shaped by sociocultural interaction. Internalization is the process of internalizing a previously external component. For example, when visiting a city for the first time, a person might need a detailed map to navigate it. However, after a few visits, knowledge about the main layout of the city might be internalized, implying that the map is no longer needed. The opposite process is externalization, which is the act of externalizing internal components to the object world. Sketching a drawing with directions for a lost tourist is an example of externalization that leads to the development of a new tool: the map. Both processes cause a redistribution of the components of an activity, based on the situation. Internalization is often the result of learning or practices, while externalization can be the result of a breakdown in internalized knowledge. These processes also play an important role in socially distributed activities. For instance, when teaching to another human, the teacher's individual internalized knowledge is transformed to social externalized knowledge. In the map example, one person creates a new artifact (the map) to externalize the internal knowledge to the lost tourist. There is thus a close connection between the internalization/externalization processes and the individual and social activity.

4.2.5 Mediation

Fundamental to Activity Theory is that human activities are mediated by tools or artifacts that allow humans to act on the world. All human activities are mediated by a complex system of both material and immaterial artifacts that influence and shape human interaction with the world. Tool mediation is a socially and historically developed objectified act in the world as tools represent the experiences of humans, which "*is accumulated in the structural properties of tools, such as their shape or material, as well as in the knowledge of how the tool should be used*" [99]. Artifacts or tools are used to convey sociocultural knowledge. Vygotsky originally made a distinction between external material tools, such as a hammer or a computer, and cognitive tools or signs, such as a map or mathematical formula. Both types of tools are mediators of human activity as they shape and effect the activity. Within Activity Theory, culturally developed artifacts are fundamental mediators of the subject-object relation that connect the human to the objective world and to the sociocultural history [98].

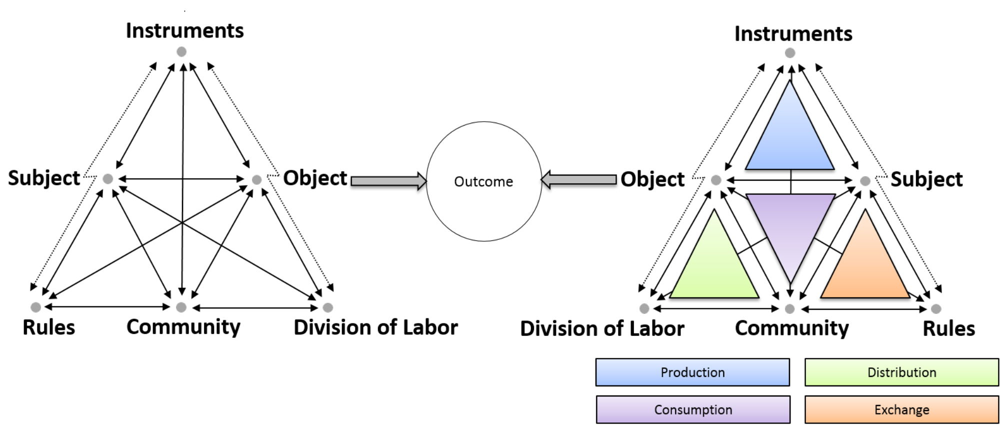


Figure 4.3: Collective activity is a mediated subject-object-community relation (adapted from [61, 62]).

4.2.6 Collective Activity

Although, throughout his description of Activity Theory, Leontiev clearly articulates that an activity can be composed of a collective subjects, he does not systematically explore the implications of a collective activity. Engeström [61, 63] therefore systematically extended the individual activity into a collective concept that introduces a third fundamental component, *community*. To highlight the highly social nature of a human activity, he proposed to embed all actions performed by humans in a collective activity that can be defined as the mediated subject-object-community relation. Within this model, the community represents the social strata in which the subject-object relation is embedded. As mediation is a fundamental concept within Activity Theory, Engeström introduced two additional mediators: rules, mediating the subject-community relation and division of labor, mediating the community-object relation. The activity systems created by the S-O-C relation are, thus, mediated by tools, rules and division of labor (Figure 4.3). First, tools provide the subject with a way to act in the world. They externalize the act in the world through enactment and are shaped by affordances and resistances. Second, rules define how the act of the subject is embedded in the social context. They socialize the act in the environment, culture and world. Third, division of labour structures the relation between the social strata and the object of the activity. It links the distribution of work among community to the hierarchical motive towards the object. Engeström [62] categorized four fundamental processes (Figure 4.3) that are interwoven into collective activity. These processes are: (i) production, (ii) consumption, (iii) exchange and (iv) distribution.

Activity systems are interconnected nodes and hierarchies that directly influence each other. Engeström described misfits or problems within and between different activity systems, which he called *contradictions*. He considers four different types of contradictions (Figure 4.4). Primary contradictions occurring within each node of the system point to instabilities or problems within a mediator; secondary contradictions which are tensions between different nodes in the system, cause a breakdown or problem; tertiary contradictions are problems between the form and structure of activity systems and the outcome, pointing to a fundamental mismatch between the system and the projected motive; and, quaternary contradictions reflect instabilities and problems within a network of activity systems caused by mismatches between different connected activity systems.

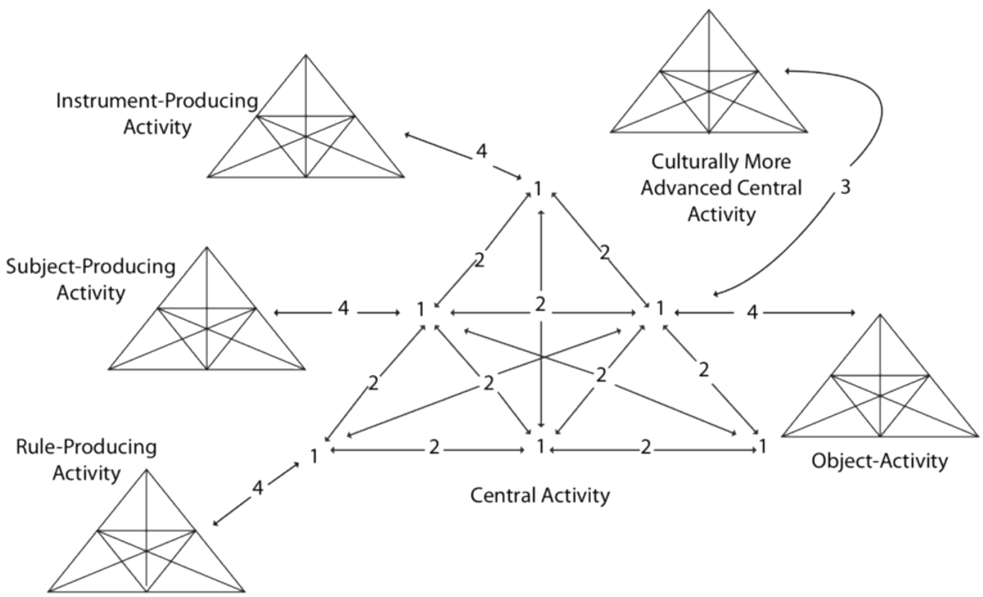


Figure 4.4: Contradictions in activity systems [41].

4.3 Computer-Mediated Activity

This perspective of using mediated subject-object-community relation as a basic inseparable unit of analysis differs substantially from the classic HCI view in which users and systems were analyzed and modeled as a formal closed and independent information processing loop [40]. The focus of Activity Theory shifted this narrow analysis of human-computer interaction to a broader perspective that considered the real life implications of human activity by including the sociocultural context into the unit of analysis. Because "*humans act through the interface*" [33] the focus within HCI moved to *computer-mediated activity* [94] in which computers are considered as a special type of mediating material tool, that is used and appropriated by subjects to act on the world. Humans rarely want to interact with computers directly, but rather use them to access information or processes relevant to the activity. Computers can thus be considered as general purpose tools that are reconfigured and appropriated for each activity. This tool mediation in Activity Theory is a dialectical relation between the tool and the user. To delineate this border between the individual and the tool, Kaptelinin [95] describes the concept of *functional organs*. A functional organ is a functionally integrated goal-oriented configuration of internal processes and external mediated resources. It describes how external tools can become an integrated part of individuals to augment human interaction. Notebooks augment human memory, a scissor allows users to more efficiently cut paper and glasses allow people to see better. Computers provide access to tools that are integrated into functional organs. The question is how computer tools can be integrated into human activity? [95] Computers do not have a fixed function but can be considered as meta-tools whose internal state can be changed to match a tool that integrates with a functional organ. The assimilation of these meta-tools is a continuous process that is related to both technological advancements reshaping the meta-tool possibilities, and, more importantly, the developing needs of the user. This assimilation process leads to frictions within tool configurations, especially considering the increasing number of devices.

4.3.1 Activity-Centric Computing

Activity-Centric (or Activity-Based) Computing is an interaction paradigm and Ubiquitous Computing approach that operationalizes the core argument of Activity Theory —human activity as the unit of analysis— into a computational platform. Activity-Centric Computing revolves around *computational activities* that encapsulate a number of services, applications and resources into a computational model that is a reflection of human intent. These activities can be created, shared and used by multiple participants. Bardram [18] captured most work described in Chapter 2 into a generic and reusable model for Activity-Based Computing (ABC). Figure 4.5 depicts the hierarchical model in which multiple participants interact with an activity that contains a number of services (applications or other tools) to access data and other resources. In this approach, activities become first class computational entities that users can directly interact with.

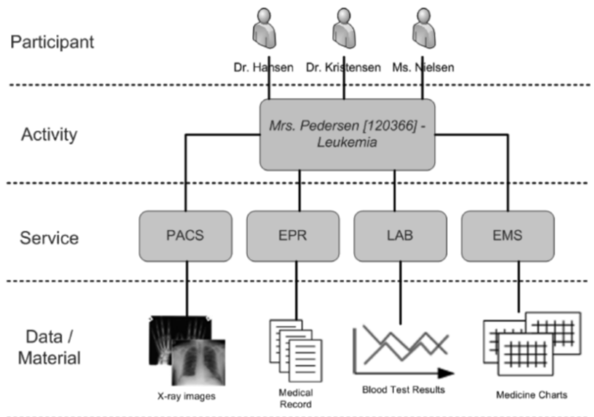


Figure 4.5: A computational activity organizes all related services, resources, and users into one context model that can be used by multiple users [18].

Based on both empirical and theoretical work, Bardram [18] proposes five high-level principles for Activity-Based Computing:

P1 **Activity-Centric** - Human work is organized and abstracted into *activities*, which are high level computational artifacts that encapsulate resources and applications into one goal-oriented interaction model. This model suggests moving away from the classic application-oriented interface and to focus on supporting activity-oriented workspaces. Users interact with activities rather than applications, resources and services.

P2 **Suspend and Resume** - Users can interact with multiple parallel activities. Switching between activities is done by suspending and resuming them. Suspending an activity stores the state and removes the visualization from the interface. Resuming re-initializes the activity and restores the applications and resources associated with the activity.

P3 **Roaming** - Activities are used as a computational abstraction on both the user interface and infrastructure level. This implies that activities can be accessed by multiple users. This allows users to suspend activities on one workstation and resume it on another.

P5 **Sharing** - Activities can be shared with other users. Users that are added to an activity as participants, can access and modify it. Sharing can occur both synchronously and asynchronously. When multiple users access the same activity, a collaborative and synchronized setup emerges that provides a real-time connection. Asynchronous interaction is possible by different users that suspend and resume the activity at different points in time.

P6 **Activity-Awareness** - Because activities can be used by multiple people, they are aware of their use context. The location, amount of users, but also the social setup can be sensed and used to automatically select the correct activity. Activity-awareness can also be leveraged to suggest tools or applications to users.

4.3.2 Artefact Ecology

The Activity-Centric Computing paradigm, however, places very little emphasis on the increasing amount of devices employed by humans. It is unclear how activities can support cross-device setups, interconnections among devices or how a user would include new devices into an activity. The model essentially falls short in capturing the dynamism between users and devices. These device connections and dynamics are described by Jung et al. [92], who proposed the notion of an *artifact ecology*, which is *" a set of all physical artifacts with some level of interactivity enabled by digital technology that a person owns, has access to, and uses"* [92]. Building on the *ecology* concept introduced by Gibson [72], they describe the existence of implicit and explicit relations among interconnected devices that are part of human life. Devices cannot be analyzed and perceived as individual entities but "*artifacts must be understood as part of an artifact ecology*" [37]. Based on ongoing work, device ecologies are adapted and changed in an ad hoc manner. However, users strive to stability in these configurations with maximal flexibility to handle change [136]. In a stable state, "*the artifacts of the artifact ecology have found their role*" [37]. Unfortunately, this definition carries two fundamental assumptions. First, it does not differentiate between computing devices with complex multiplexing capabilities (such as laptops, smartphone or tablets) and peripherals (such as mice, screens or printers). Second, the definition does not emphasize the *activity* of the user. Fundamentally, devices are used as dynamic tools to accomplish goals. Users do not act with computing devices, but rather through computing devices. Bødker and Klokmose [37, 36] therefore propose to relate this notion of artifact ecologies to human activities because *"the artifact ecology of an individual is highly dynamic"* [37] as it is constructed by the user through their activities. However, as described in Chapter 3, with an increase in meta-tools, there is also an increase in configuration work.

Although configuration work increases with every new device or user that is part of the activity, it is primarily a problem of configuring what is *on* the device. A computing device has both functional features (what can be done) and operational features (how it can be done) [34], and Bødker describes user interfaces as the operational aspects provided by the application running on the device. User interfaces are thus *how* actions can be done, implying that computer applications are externalizers. Based on this assumption, Bødker [34] distinguishes three aspects of interfaces: the *physical* aspects, which are operations and handling of the physical device; the *handling* aspects pointing to transparent operations and handling of user interfaces; and, *subject-object* aspects describing the operation conditions of the artifact. From an activity-centric perceptive, configuration problems can be described as breakdowns or misfit of tools within a specific activity system. Material meta-tools have an intrinsic property that they require some form of configuration to create a stable tool setup that allows the subject to act transparently through the artifact. Breakdowns can occur on multiple aspects within computer use [34]: in the handling of physical artifact, in the ability to understand or operate the artifact; and, in the match between the goal of the user and the current configuration or capabilities of the artifact. The tools used by the subject to interact with the world can break down and force the subject to perform configuration work, in which the tools are reconfigured and aligned with the motive of the activity. In this situation, the unconscious operations through which the subject acts fail, initiate a de-automatization process in which the act is shifted from the tools that are aligned with the functional organ, to direct interaction with the meta tool. For example, when clinicians have the incorrect patient information loaded on a tablet, their focus shifts from interaction with the patient to configuring the tablet to match the motive of the activity. Configuration work is thus the activity of stabilizing the tool configuration through re-automatization of the action; it is a tool creation activity.

4.4 Activity Configuration

To decrease the level of configuration work, I propose an activity-centric system approach that revolves around using context models that are a reflection of human intent. In summary, given that previous approaches have placed very little emphasis on the role of devices and material artifacts, I both extend the existing Activity-Based Computing model proposed by Bardram [18] with a device layer, and reconceptualize their basic notion of a *computational activity* into a broader *activity configuration*. I define activity configuration as *a description of an interaction and social context (including files, devices, applications, material artifacts, other meta information, and coordination and communication tools) that is a reflection of the real ongoing activity*. I make this distinction between a computer mediated *activity configuration* and an *activity* to clearly delineate the difference between the real human activity, and the part that is mediated and controlled by a set of computing devices. An activity configuration captures: which users are part of the activity, what resources are used, and what devices are used as part of the activity. Compared to the traditional Activity-Based Computing model [18], I introduce a new device and artifact layer. In addition to sharing or distributing an activity configuration among users, activities can be also *fragmented* across devices. Thus, a single activity configuration and its resources can span across several devices and material artifacts, and changes to its state are propagated and visualized on these attached devices. By explicitly using activities as fundamental first-class computational structures, such activity configurations can be (i) constructed, (ii) shared and (iii) restored across a device ecology. I describe users' device ecologies as *ad hoc and dynamic interrelations among interconnected devices that are part of the same motive-oriented activity*. By connecting the activity of users explicitly to their devices, users are presented with a cross-device representation that moves away from the predominant application and document-centric paradigm. Activity configurations are used to reduce the amount of configuration work when working in a distributed interactive setup, by allowing users to manipulate configurations across devices. Figure 4.6 depicts the structure of an activity configuration, which consists of subjects, devices, signifiers and digital configurations.

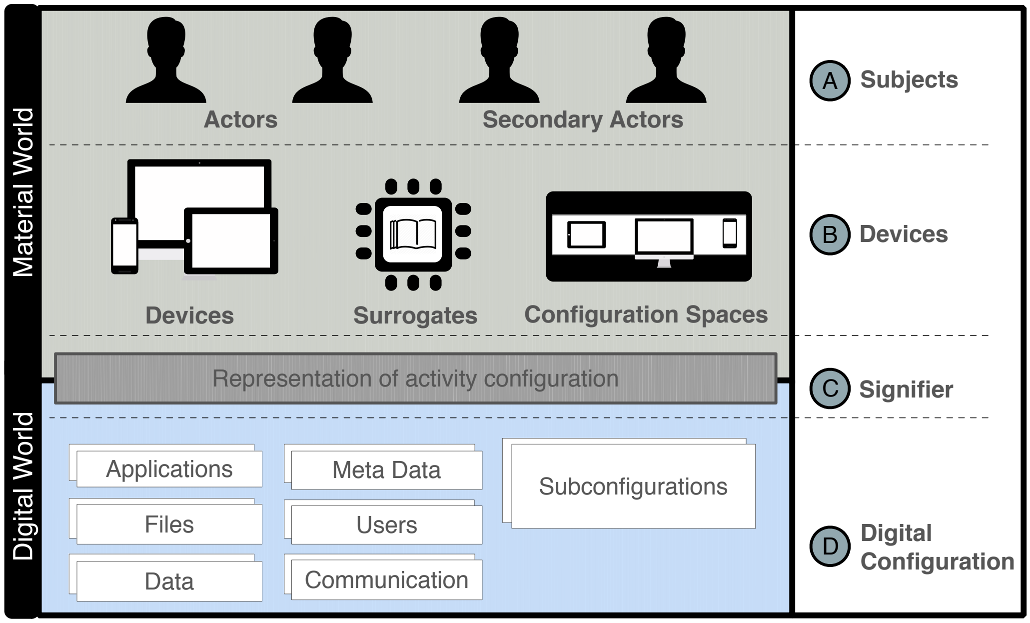


Figure 4.6: An activity configuration consists of subjects (A) that employ a number of computing devices, material artifacts that are augmented with surrogates and configuration spaces (B) to interact with a digital configuration of information and data (D) which is externalized to the user in the form of a signifier (C).

4.4.1 Subjects

Subjects (figure 4.6A) are human agents that are digitally represented in the system as part of the digital configuration. Subjects can be both *actors* and *secondary actors*. Actors interact with the system, motivated to complete the *objective* of their activities. They act on the system by using devices that are relevant and related to the activity. Actors own, shape, define, consume, and share activity configurations by interacting with the system. Secondary actors are not directly involved in the *situation* of the interaction of the activity, but contribute to its relevance. They represent external stakeholders that influence and define the object of the activity. Both the capabilities of actors and secondary actors are defined through roles, which describe what actions are accessible and executable by each subject.

4.4.2 Devices

Subjects employ devices to interact with digital information (figure 4.6B). Devices are computers used to *directly* interact with user interfaces that provide access to digital information. Devices can be used in *ensembles* in which digital configurations are fragmented across multiple devices. Next to standard computing devices, I also introduce two new classes of digital devices: *configuration spaces* and *surrogates*. Configuration spaces are devices or digitally augmented spaces that provide subjects with possibilities to *directly* or *indirectly* interact with digital information through a specialized mediating device. They are special computer devices that actively mediate configuration work between devices and users. Configuration spaces are active spaces that sense users and devices and provide a platform for cross-device configuration work. An example of a configuration space is the ActivitySpace system [**C4**] in which an interactive desk is used to mediate interaction across devices. Non-digital artifacts are included into the activity configuration using surrogates. Surrogates are specialized devices that augment non-digital artifacts with sensors and digital capabilities to include these artifacts into the digital representation of activity. They augment non-digital artifacts into TUIs. An example of a surrogate is the Hybrid Patient Record device [**C6**] that augments a paper patient record in order to connect it to a digital mobile device. Summarized, I distinguish between three types of devices:

Personal Devices - Personal computing devices include desktops, laptops, tablets, large interactive displays, smartphones and smartwatches that are used by subjects to directly interact with digital information.

Surrogate Devices - Surrogates are specialized devices that employ sensors to augment non-digital artifacts to connect them into the digital information.

Configuration Spaces - Configuration spaces are specialized augmented spaces that actively mediate digital information across a number of different devices.

4.4.3 Digital Configurations

Digital configurations are snapshots of digital information that are a reflection of a stable set of digital tools, such as applications, services or other digital representations of information, in a particular activity. These configurations include applications (or views on data, depending on the underlying paradigm), files, database items, access models of users, metadata and communication and coordination mechanisms. Similar to previous approaches [18, 178], all these information items are encapsulated into a stable digital representation of human intent. The configuration of digital tools is a device agnostic work configuration that can be appropriated, used, shared and fragmented between devices and subjects. This appropriation and distribution are captured in specific sub-configurations that describe how each device or subject is accessing or using the digital configuration, which allows for device or user specific interpretations of activity configurations. These digital activity configurations are not rigid models that dictate or force a specific work style as they neither enforce a specific interface paradigm nor reject existing approaches. The digital configuration mechanism is an abstract configuration tool that users can apply to organize information and communication across devices and artifacts.

4.4.4 Activity Signifiers

To externalize and socialize the activity configuration, it is represented as an *activity signifier*. An activity signifier is a psychological or cognitive tool, that is used by subjects as a social reflection mechanism to describe digital configurations [98]. They externalize, describe and reflect the configuration constructions into a simple tangible and stable sign. The sign can exist both in the digital world, e.g., as a user interface element, or in the object world, e.g., as a real physical object.

Digital activity signifiers have been frequently used in previous work in the form of Activity Bar buttons [13], Alt+Tab overview [178], floating touch-enabled UI elements [21, 15], overlapping document montages [116, 144] or Situated Glyps [104]. Each of these interfaces leverages basic signs, such as text- and color-labels, icons, pictorial representation or novel UI elements to convey information about the content of the digital configuration. Norman [131] describes these types of signs as social signifiers, he explains that: "*we search for significant signs in the world that offer guidance. In the social world comprised of people and technology, these cues are social signifiers*". Furthermore, digital signifiers can be used as a self reflection mechanism to handle large amount of digital configurations.

Physical activity signifiers are material instruments that use the physical dimensions, affordances and properties to signify activity configurations. Using shape changes, color, sound or other augmentations of physical objects, the content of the activity configuration can be shared across different subjects and devices. These material social signifiers have basic affordances, like any material artifact, but also have a set of *collaborative affordances* that support subjects in physically externalizing activity configurations with other subjects. Although basic affordances exist as a configuration of physical properties, its perceptible meaning is often dependent on the social strata and can thus change or differ among environments or social settings. Based on these prior interpretations of the social role of affordances [101, 105, 180], I describe *collaborative affordances* as 'physical' properties that afford collaborative perceptions and actions within a specific social context. Collaborative affordances contextualize basic affordances in a social structure, thus, allowing subjects to signal the activity configurations into the physical world. Collaborative affordances are mediated actions that emerge from social practice within a cultural setting and (re-)define social activity within that practice. An example of a physical activity signifier, are the reconfigurable colored lights on the Hybrid Patient Record device [**C6**].

4.5 Activity Configuration Framework

Activity configurations operate using device and user multiplicity: activity configurations can be used by one or more subjects that employ different sharing mechanisms using different device configurations. We distinguish between: three types of subject configurations, i.e., loose connected, semi-shared and shared configurations; and, four types of device configurations, i.e, replicated, fragmented, surrogated and mediated configuration. User and device configurations are combinable, thus creating a configuration matrix design space that visualizes the possible activity configurations. Figure 4.7 depicts the design space result-

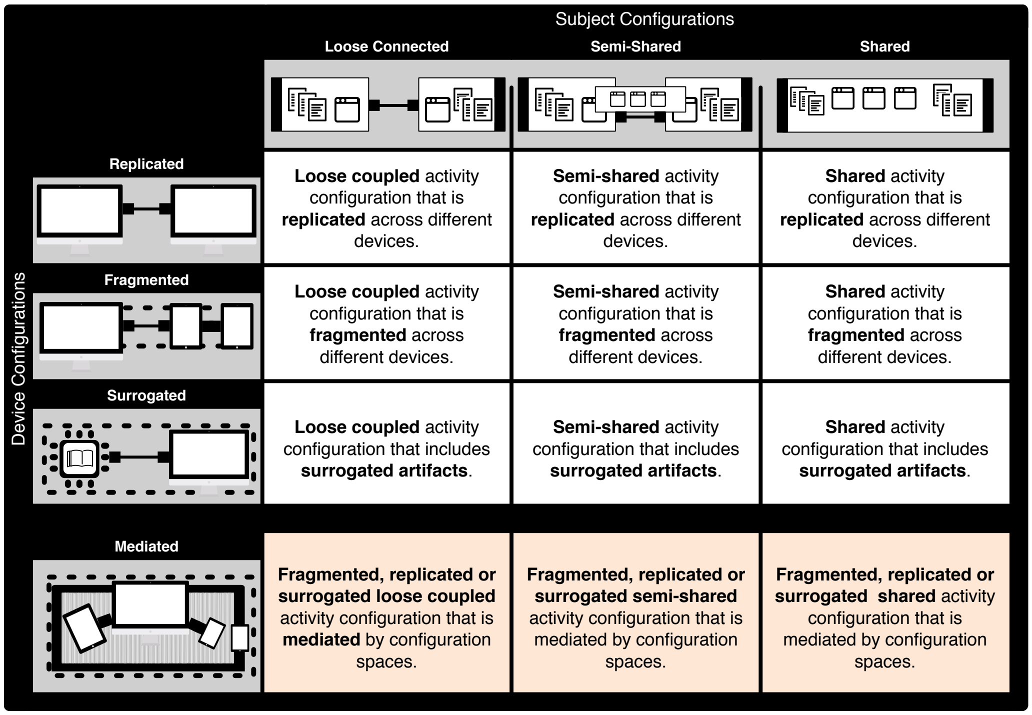


Figure 4.7: The activity configuration framework describes how activity configurations exist across subjects and devices. This leads to a configuration matrix of possible user and device configurations, in which device configurations can be replicated, fragmented and surrogated; and subject configurations can be loosely connected, semi-shared or shared. Each possible configuration can be mediated as indicated in the bottom row of the matrix.

ing from combining both types of configurations. The central purpose of this approach is to support users to:

1. **Produce** new activity configurations that capture a set of digital tools used during work, thus, creating digital configurations that are a reflection of human intention.
2. **Consume** existing activity configurations using a wide range of devices and locations allowing subjects to move their device agnostic activity configurations between different work settings.
3. **Exchange** activity configurations with other subjects and devices reusing other subjects' digital configurations to create a shared overview or template-based sharing mechanism.

4. **Distribute and Fragment** activity configurations across different connected devices in use supporting one seamless interaction space.

4.5.1 Configurations Across Subjects

Activity configurations operate using user multiplicity: multiple subjects can access, share and use the digital configurations. Subjects (either alone or in group) can create, manage and deploy activity configurations on a set of devices (discussed in the next section). Configurations can be used between subjects in three fundamental approaches: as (i) loose linked configuration, (ii) semi-shared configuration and (iii) shared configurations. Although these three approaches highlight the levels of shared usage, they can overlap into hybrid approaches or transform over time.

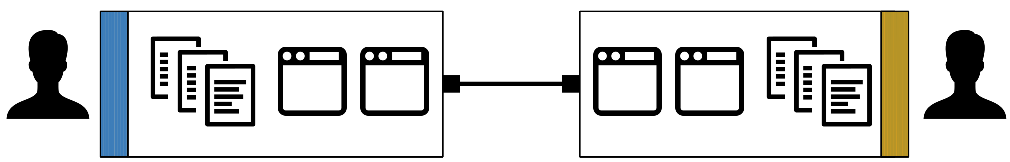


Figure 4.8: A loose connection model in which several activity configurations exist as separate disconnected configurations that contain communication and coordination tools that link the actors.

Loose Connection

Within a loose connected activity configuration (Figure 4.8), two separate digital configurations exist that have connected communication and coordination tools within the configurations. This allows multiple subjects to create individual activity configurations with separate activity signifiers (shown in the figure as blue and orange blocks) that contain a similar but modified set of digital tools and resources. Especially in work environments, such as office work, these types of activity configurations can be used in shared project models or highly collaborative project teams. This allows subjects to appropriate a generic configuration, such as a project template containing files, contact information and other resources, and create an individualized activity configuration that is tailored specifically for their role in the collaborative setup. These individualized configurations can be shared and reused by other subjects, thus creating an asynchronous configuration sharing mechanism. Although there is an overlap in resources and tools in use, they are not explicitly shared within the configuration model, but rather split up into individual activity configurations. This setup provides users with a high level of freedom on how to share resources from within the activity configuration.

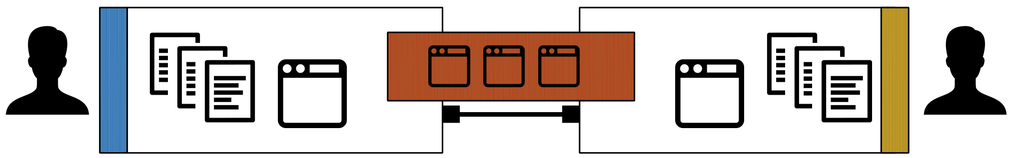


Figure 4.9: Semi-shared activity configurations are separate configurations that contain an overlapping resource container.

Semi-Shared

Next to loosely connected configurations, there are also semi-shared activity configurations. If loosely connected activity configurations have shared overlapped resources or applications, they become semi-shared configurations (Figure 4.9). Although the configurations are separate as they have different signifiers and digital configurations, they share a container for real-time distribution and replication of a delineated part of the digital configuration. This allows subjects to externalize part of their configurations and create shared overlapping information repositories. These repositories can be created ad hoc based on needs of multiple subjects that are coordinating on a work activity, but can also exist as policy induced by a company or a work team. Therefore, separate digital configurations contain an overlapping subconfiguration that contains a set of resources that are synchronized and shared with all subjects that have access. Subjects that constructed a loose connected setup can easily move back and forth between the loose and semi-shared setups. The containers of resources can thus be constructed and removed from ongoing loosely connected activity configurations. This mechanism provides subjects with the ability to easily co-create shared constructions to support emerging collaborative work.

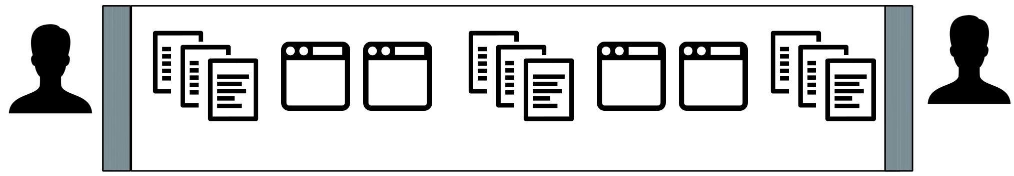


Figure 4.10: A shared activity configuration where multiple subjects access the same configuration.

Shared

If multiple subjects interact with the same activity configuration, the configuration is shared among all subjects (Figure 4.10). Because of the user multiplicity principle, all subjects have equal access limited by the description of their role. This means that all changes to the digital configuration are updated, shared and replicated across all user sessions. This type of configuration is used in more rigidly structured environments that require a high degree of collaborative work, such as hospital work. In these environments, the configuration has to be continually updated, synchronized and shared to create a shared overview of work. The activity signifier is also co-created, through work practices or dictated by policy and is a reflection of collective activity. Shared activity configurations are thus shared synchronized digital configurations of work that are co-created and mostly part of formal work practices. Note that a shared activity setup is not the same as sharing the entire digital configuration of a semi-shared setup. The central difference is that although they both share the entire digital configuration, the semi-shared setup has separate activity signifiers, i.e, within activity-centric systems the configuration is visualized, depicted and organized in a different individualized way. In a completely shared setup, there is one unique shared activity signifier, creating consistency across users and devices. Finally, in a shared activity configuration, there is no subject ownership over items in the digital configurations as all of them are co-owned by all subjects.

4.5.2 Configurations Across Devices

Activity configurations also operate under device multiplicity as digital configurations can be replicated, fragmented and shared across devices. As described above, I consider three types of devices: personal devices, ranging from smartwatches to tablets and laptops, all the way to large interactive displays; surrogates that augment non-digital artifacts so they can be included into the digital abstraction; and, configuration spaces that mediate activity-centric configuration work between other devices. Within devices, I distinguish four types of deployments of *one specific* activity configuration: (i) replication, (ii) fragmentation, (iii) surrogation and (iv) configuration space mediation.

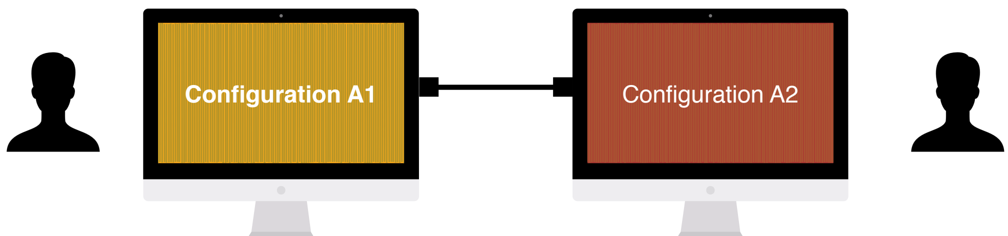


Figure 4.11: Activity configurations can be replicated between approximate homogeneous devices.

Replication

An activity configuration can be replicated across multiple devices (Figure 4.11). This means that an activity configuration is captured, packaged and sent from one device to other devices with similar capabilities. This second device has the ability to interpret, deploy and locally situate the activity configuration. This creates an exact but disconnected replica of the initial activity configuration, and depending on the type of user configuration, the device configurations are connected through communication and collaboration (loose coupled configurations), partially overlap with resource containers (semi-shared) or have a real-time synchronization between the views (shared). Replication allows subjects to grab a working context on one device and send that context to another device. Depending on the type of activity configuration, this allows for asynchronous, synchronous or hybrid collaborative setups. Replication in part captures how many other activity-centric systems currently operate. For example, systems such as the ABC system [18], ReticularSpaces [15] and Activity Explorer [125], allow the sharing of an entire activity model that is replicated across the devices of different users. In these setups, users have a copy of the entire activity that is either automatically or manually synchronized during work.

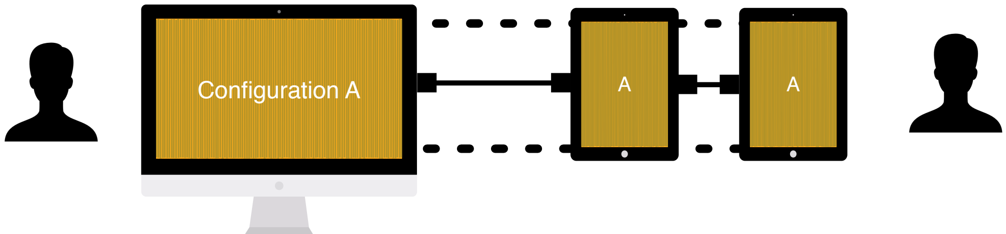


Figure 4.12: Activity configurations can be fragmented across multiple heterogeneous devices.

Fragmentation

Activity configurations can also be fragmented across different devices (Figure 4.12). Fragmented means that parts of the same activity configuration are distributed across a set of devices employed by the same or different users. The content displayed on the different devices belong to the same digital configuration. Within this setup, devices employ roles that define how digital configurations are distributed and synchronized: a master device is always identified to maintain the configurations and to push part of the activity configuration to secondary or slave devices. These devices include the same signifier but only display the specific allocated subset of the configuration. Any interaction on these secondary devices is immediately synchronized with the master device. This mechanism allows users to employ secondary devices as physical windows or resource containers that are easily connectable to the master device that holds the main activity configurations.

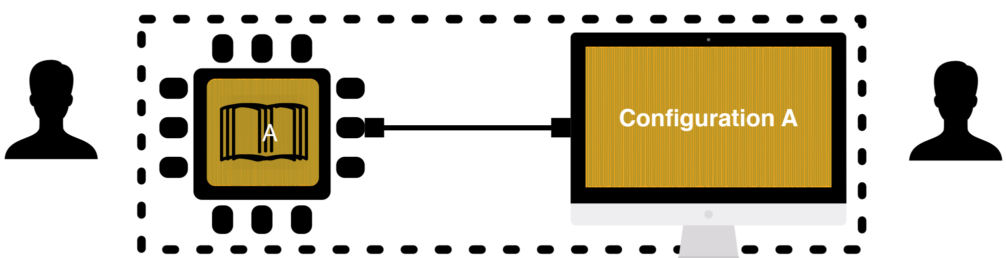


Figure 4.13: A surrogate augments a non-digital artifact so it can be included into the activity configuration.

Surrogation

Surrogates are sensors or augmentation platforms that augment a non-digital artifact with a number of interactive capabilities allowing the artifact to externalize part of the digital configuration using its properties or form factor (Figure 4.13). The surrogate is a mediator that provides duplex interaction between the non-digital artifacts and the activity configuration. The configuration can thus be visualized or externalized on the properties of the artifact, while any reconfiguration of the physical properties of the artifact changes the digital configuration. In other words, surrogates morph non-digital artifacts into TUIs that are part of the activity configuration. The surrogation process is a structured mechanism to include real physical objects into the activity configuration. Surrogates also allow subjects to create material activity signifiers, in which the sign representing the activity configuration (or *phicon* if I use Ishii and Ullmer's terminology [87]), is a property of the augmented non-digital artifact. Finally, it is important to note that depending on their capabilities, that are dictated by the sensors or augmentation, surrogated artifacts can be attached to any device configuration which potentially also results in their ability to be included in a configuration space.

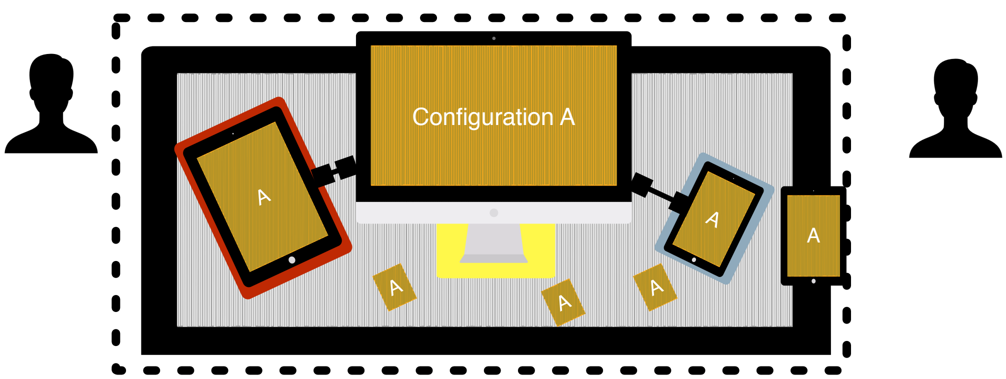


Figure 4.14: A configuration space is a special device or augmented space that mediates activity configurations across heterogeneous devices.

Configuration Space Mediation

To mitigate the challenges associated with handling activity configurations among a larger amount of devices, a special mediating device or space is employed to visually mediate cross-device activity configurations. We draw upon spatial models, such as the Situative Space Model [139] and Proxemic Interaction [8], that describe how the spatial organization of objects (and devices) as well as the action possibilities of human actors influence human perception as well as human tool ability. Because devices exist in the material world, they are intrinsically tied to spatial configurations. Subjects carry, align, position and move devices inside such physical spaces as part of their interaction with activity configurations. These spatial models suggest that knowledge about the spatial relations, such as orientation, distance, movement, identity and location, of both humans and devices, can leverage humans' spatial memory and create a better, more intelligible interaction environment that reduces configuration work.

To manage complex cross-device activity configurations, I introduce the concept of a *configuration space* to reduce configuration work across multiple devices and activity configurations (Figure 4.14). A *configuration space* is defined as a *digitally augmented physical action space [139] that visualizes the activity configuration of the user across all connected devices, using the surrounding space between the active devices*. The configuration space is created using a specialized device (such as a projector, interactive surface, or a body worn projector) that mediates the interaction between other devices and their user interfaces. A configuration space can be either public or private and has three fundamental functions:

Device Management - Dynamically and visually create and manage device ecologies by coupling or decoupling devices. Based on the changing focus of the users, the space allows users to automatically or manually change the role of the device.

Activity Configuration management - Create, copy, move, share, distribute, and fragment activity configurations across all devices in the ecology. The space allows for auto-configuration of devices using the configuration of previous activity states or other similar devices.

Interaction Space - Configure devices by manipulating resources through interaction or by the physical properties of the configuration space. By leveraging the physical dimensions of the space or interaction techniques, users can pair devices, move resources between devices and re-configure activity configurations.

A configuration space is activated by placing the master device in the space. All other devices that are added to the space become part of the same activity space. The space visualizes the content of the activity configuration as distributed among devices. The configuration spaces can support replicated, fragmented or surrogated device setups. Moreover, the space allows users to visually transform their activity configurations between the 3 different setups. Configuration spaces materialize activity-centric configuration by externalizing internal configurations in an actionable perceptible space.

4.5.3 Mapping Related Work

The framework can also be used to classify related work in multi-user and multi-device activity-centric systems. Figure 4.15 depicts all previous systems that support multi-user or multi-screen activity-centric computing in the configuration matrix. Only one previous system [26] supports a loose connected activity configuration which connects multiple users through specialized tools that allows them to collaborate and share information within those disconnected activity configurations. In this system, emails and other resources are sent through the system to other users who can organize them in activity streams. Giornata and ABC4GSD [174, 178] are currently the only approaches that allow for shared resource containers between different users in the form of a shared file region on the desktop and by allowing developers to select a subset of resources that need to be shared within one activity configuration. Most other existing systems [15, 18, 21, 125, 172] employ a shared replicated approach in which the entire activity configuration model is synchronized and distributed in its entirety. The model itself, however, ranges from systems that utilize activity configurations as basic resource containers to systems that include complex collaboration and communication tools into the activity configurations. Finally, although they technically do not support multiple devices, two previous systems [116, 144] allow for the fragmentation of one activity across a second screen, which in both cases is used to visualize the activity signifiers.

The systems that are introduced in Chapter 5 and the papers in Part II explore the open or sparse part of the matrix and contribute specifically to supporting semi-shared knowledge work setups [**C1**], fragmented multi-device setups [**C2**, C5], mediation of devices using different types of configuration spaces [C2, C3, C4] and surrogation of paper documents in clinical work [C6, C7].

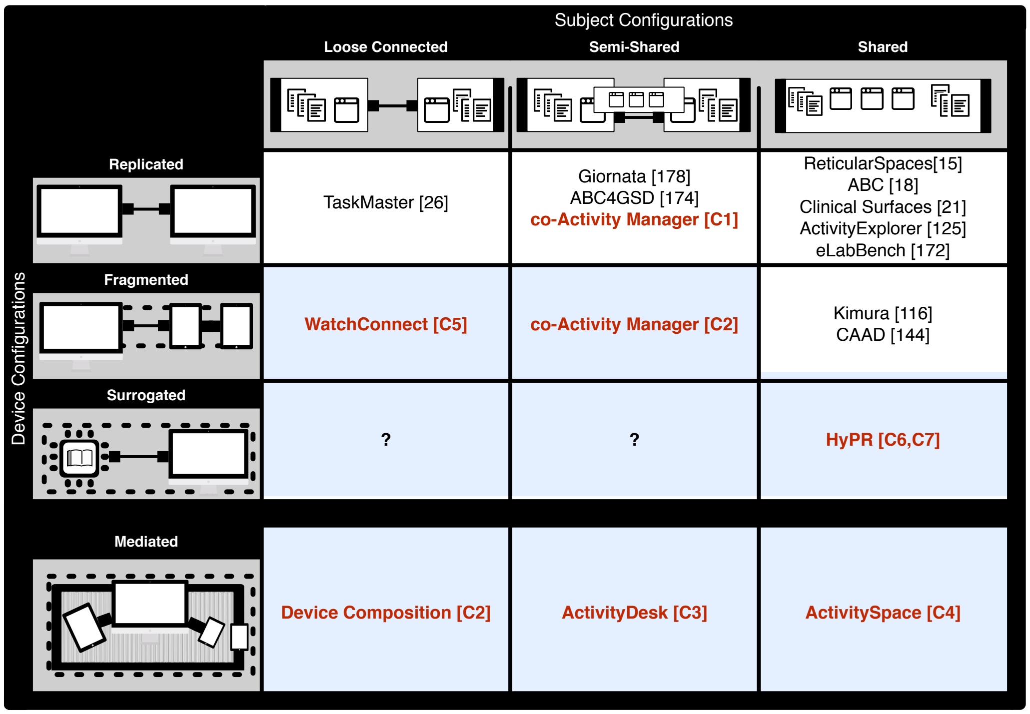


Figure 4.15: A set of previous systems that support multi-user or multi-screen activity configurations. The blue background emphasizes currently unexplored spaces in the matrix. The red colored references are systems introduced in this thesis.

4.6 Exploring Activity Systems

Using the conceptual activity configuration framework, we can model domain specific activity systems that are composed of different connected activity configurations that are being used and developed by a number of users who employ an ecology of devices and artifacts to interact with the digital information. Within this thesis, I focus on two specific domains, office and hospital work, and I here present two conceptual examples of setups that use the basic components above to describe a multi-user and multi-device activity systems.

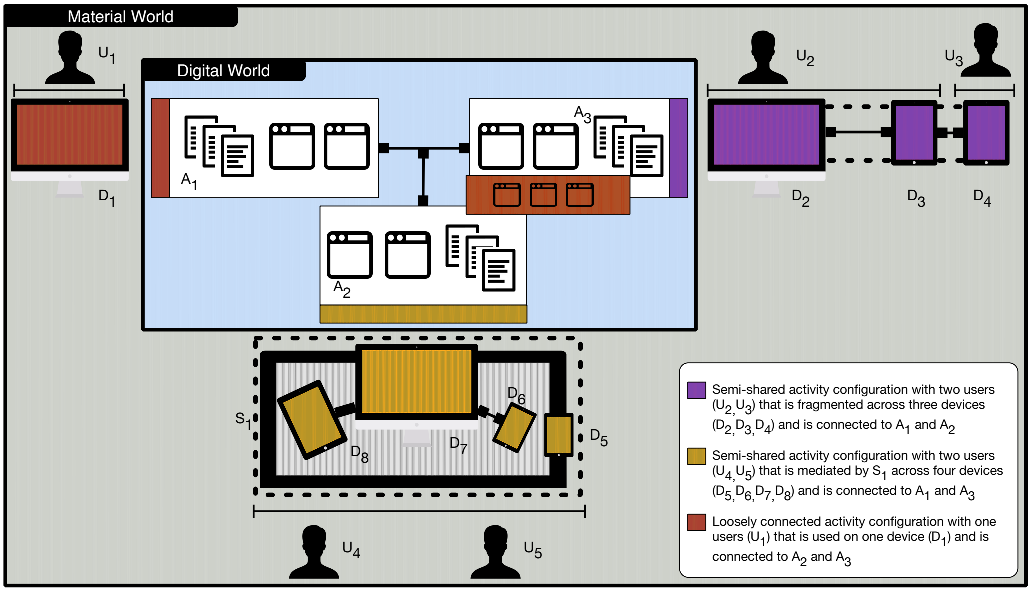


Figure 4.16: This model of office work depicts a team of knowledge workers that collaborate on a number of overlapping activities using a range of device setups.

4.6.1 Office Work

Figure 4.16 depicts the office work example that consists of three activity configurations (red, purple and yellow) which are created and used by a number of users which are part of the same team. The red configuration is being used by user U_1 who employs device D_1 to interact with a digital configuration A_1. Since the configuration is loosely connected, he can contact and communicate with the rest of the team from within the activity configuration, but cannot access the signifier or content of the other activity configurations. The purple activity configuration is in use by two remote users U_2 and U_3 who have fragmented one activity configuration across several devices in use. U_2 is employing a desktop interface and a tablet (D_2 and D_3) to interact with the digital configuration A_3. This configuration is fragmented and thus synchronized with a third device D_4 used remotely by U_3. This purple configuration is loosely connected to A_3, which means U_2 and U_3 can communicate with U_1 from within the configuration. The configuration is also semi-shared with A_2 through a shared resource repository containing information synchronized between the two configurations. Finally, the orange activity configuration is used by two collocated users U_4 and U_5 that are using a public mediating configuration space S_1 to physically spread the digital configuration across four devices in use (D_5, D_6, D_7 and D_8). Any changes to the digital configuration is automatically visualized on the configuration space and automatically synchronized across all connected devices. Similar to the other configurations, this activity configuration is also loosely connected to A_1 and semi-shared through the resource repository with A_3.

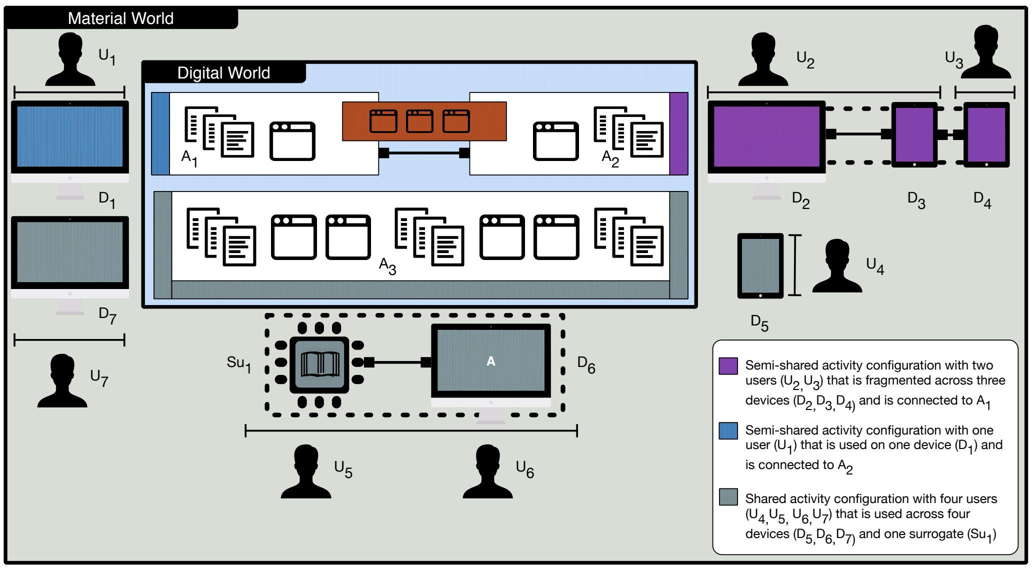


Figure 4.17: This model of clinical work depicts a team of clinicians that collaborate on a number of overlapping activities using a range of device setups.

4.6.2 Hospital Work

Figure 4.17 shows a conceptual example of three activity configurations being used by clinicians at a ward. Again, for simplicity, the example consists of three activity configurations (blue, purple and gray) created and used by a team of 7 clinicians. The blue activity configuration (A_1) is semi-shared with A_2 and is used by user U_1 through a single device D_1. Similar to the previous example, the configuration is semi-shared through a resource repository and allows for direct communication with the activity configuration A_2. Also similar to the previous example, the purple configuration is fragmented across three devices (D_2, D_3 and D_4) and used by two remote users U_2 and U_3. The third gray activity configuration is a shared configuration (e.g., a patient case) that is used by four users employing four different devices. User U_4 is using a tablet D_5 to access the configuration remotely and on the move. Similarly, U_7 is using the configuration from another remote location, through a desktop device D_7. Users U_5 and U_6 are collocated and are accessing the configuration using a large display D_6 and a paper document augmented with surrogate SU_1.

4.7 Summary

This chapter first introduced the use of theory in HCI in Section 4.1. Motivating the choice for Activity Theory as a theoretical perspective, Section 4.2 introduced the basic concepts and principles of human activity and activity systems. Section 4.3 re-conceptualized human-computer interaction into *computer-mediated activity* and Activity-Centric Computing. Extending Activity-Centric Computing with a devices layer, section 4.4 introduced the concept of *activity configuration*. Section 4.5 described a framework that explains how activity configurations are shared among subjects and devices. Finally, Section 4.6 exemplified how the framework allows for the development and design of activity systems, which are collections of interconnected activity configurations. This chapter presents the conceptual background and framing of the papers, which are summarized and linked to these concepts in Chapter 5.

5. Technology for Activity Configurations

"The major difficulty for HCI is that the object of study is not an independent natural phenomenon, as in all of the sciences. Nor is it solely the creation of new artifacts, as in the design and engineering disciplines. HCI studies the interaction between people and artificially-created-artifacts"
— Wendy E. Mackay and Anne-Laure Fayard [117]

The previous chapters described the significance of configuration work in distributed interaction and introduced *activity configurations* as a conceptual approach to this problem. Using activity configuration, devices, users and their activities can be connected and represented in a computational framework. This chapter presents an overview of tools, infrastructure and systems that operationalize the activity configuration framework and demonstrate how technology can be designed to support human activity in rich device ecologies.

Section 5.1 first provides an overview of how each system fits in the activity configuration matrix (Figure 5.1). Section 5.2 presents an infrastructure, NooSphere [**C2**], that is designed to allow for the prototyping, design and deployment of systems and technology that apply activity configurations. Using this infrastructure and the activity configuration framework as a conceptual background, Section 5.3, 5.4, 5.5 and 5.6 introduce four systems: (i) co-Activity Manager [**C1**, **C2**], (ii) ActivitySpace [**C3**, **C4**], (iii) WatchConnect [**C5**] and (iv) Hybrid Patient Record [**C6**, **C7**]. Each of these systems operationalize on cell in the activity configuration matrix discussed in Chapter 4 and demonstrate how the framework can be applied to mitigate the configuration work problems discussed in Chapter 3.

5.1 Overview

Figure 5.1 provides an overview on how each of the developed systems and papers fit into the activity configuration framework. The papers and systems

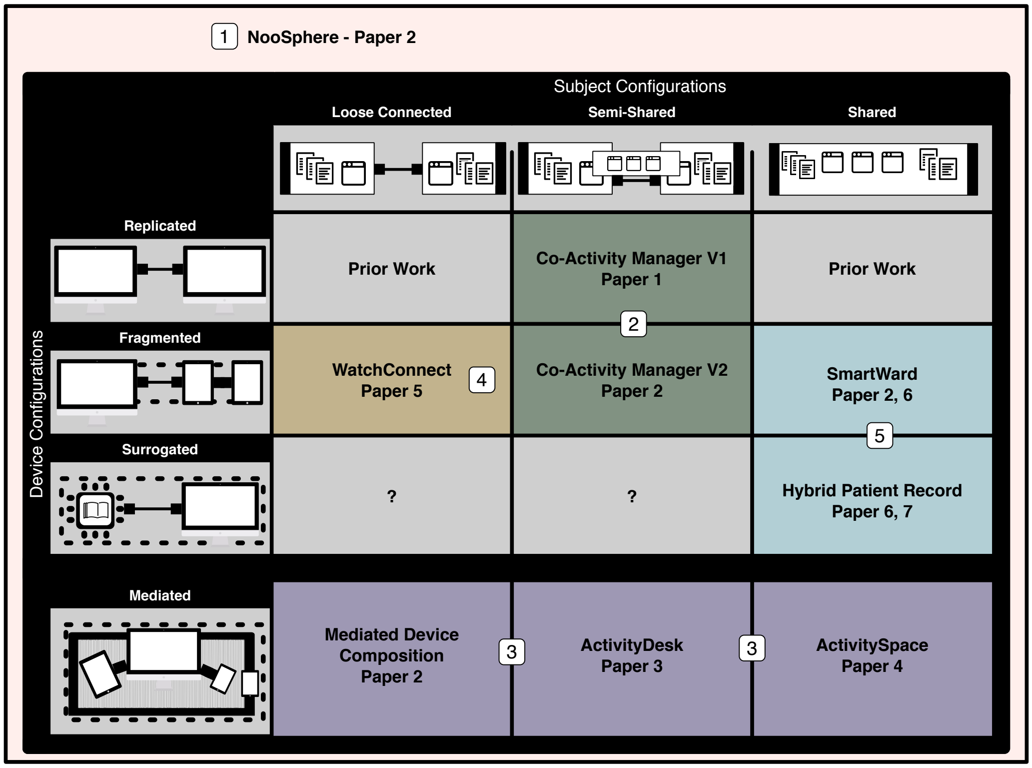


Figure 5.1: An overview of how different technological systems operationalize the dimensions of the activity configuration framework.

presented as part of this dissertation explore open quadrants of the framework and can be classified into five projects that cover different spectra of the activity configuration framework matrix:

Project 1: NooSphere - An infrastructure designed for the prototyping, development and deployment of activity-centric systems that apply the activity configuration framework [C2].

Project 2: Co-Activity Manager - An activity-centric desktop manager that explores replication and fragmentation of activity configurations in a semi-shared setup [C1, C2].

Project 3: ActivitySpace - ActivitySpace that explores device mediation for different subject configurations using an interactive table as configuration space [C3, C4].

Project 4: WatchConnect - Toolkit and applications that explore how loose connected activity configurations can be fragmented across several displays using a smartwatch [C5].

Project 5: Hybrid Patient Record - Hybrid Patient Record (HyPR) devices that explores surrogating a paper patient record and aligning that record with digital information on tablets and large interactive surfaces [C6, C7].

This chapter first discusses the infrastructure side of the *activity configuration framework*. To explore how activity configurations can be used to approach real problems in distributed interaction, we designed four systems that attempt to solve various aspects of configuration work both in an office and clinical setting. The chapter provides an overview of each of these systems, but refer to each paper for details.

5.2 Infrastructure for Activity Configurations

Many of the problems, discussed in Chapter 3, that introduce configuration work are not only conceptual but often also challenges related to technical limitations of existing technologies and paradigms. These challenges, that Edwards et al. [57] refer to as "*the infrastructure problem in Human-Computer Interaction*", point to the fact that applications, interfaces and systems are never designed in isolation, but are rather built on top of toolkits and infrastructures that provide a wide range of services and concepts that reduce engineering efforts. Distributed interaction, thus, does not only have conceptual problems, but also an infrastructure problem. Within distributed interaction, these problems first of all relate to storage, replications, synchronization and contextualization of information sources such as files, databases, web sources or even applications. Furthermore, considering device multiplicity, there also exists a wide range of challenges with the dynamic ad hoc pairing, discovering and connecting of a heterogeneous set of devices that are owned and used by multiple users in different locations. Although there are a number of solutions to many of these individual challenges, the holistic interconnected character of distributed interaction often still introduces fundamental engineering issues. A central technical problem in supporting distributed interaction is, thus, to physically connect devices, resources and applications into one seamless technical framework. The activity configuration framework connects these underlying problems and challenges into a computational abstraction, but does not explain *how* these services are implemented and designed as the framework assumes transparency to users and even developers. At its core, users interact with *activity configuration* abstractions that have intrinsic support for user and device multiplicity. This transparency, however, requires a technical framework or infrastructure that provides distributed services to users, in terms of activity configuration abstractions. We therefore designed NooSphere [C2], a framework and programming model that supports the design, development and deployment of distributed activity-centric systems.

5.2.1 Requirements for Distributed Interaction

To support the vision behind distributed activity configurations that operate using user and device multiplicity across locations, the technical implementation of that configuration should encapsulate a number of fundamental services. By including these services into the activity configuration abstraction, developers can construct distributed systems using these technical activity configuration components, without the need for direct use of the underlying technical services. Based on previous work in Distributed and Ubiquitous Computing (detailed in [C2]), we derived 7 fundamental services that are required to support distributed activity configurations:

Service 1: Persistence - to support storing of activity configurations, the infrastructures needs to provide a storage mechanism that can be used to store, access and modify resources (such as files, data or other information) from different types of devices, different locations and different users. Additionally, the infrastructure needs to handle this data persistence and modification by including, e.g., permissions and offline cashing.

Service 2: Distribution - to support distribution and fragmentation of activity configurations, the infrastructure needs to provide support for the *synchronous* and *asynchronous* distribution of activity configuration data models and attached resources. This needs to be supported both in a local environment (such as between devices) but also between different remotely connected systems.

Service 3: Discovery and Pairing - to allow for device multiplicity in activity configurations, the infrastructure needs to provide support for the automatic discovery and pairing of devices and their containing applications. The infrastructure should allow for the set up of various types of device configurations as well as ad hoc changes in the composition and aggregation of device ecologies.

Service 4: Coordination and Communication - to support user multiplicity in activity configurations, the infrastructure should allow for intrinsic coordination and communication mechanisms. This means that the infrastructure should provide mechanisms to attach sharing mechanisms, communication tools, events, workflows and other messages to the activity configuration model.

Service 5: Configuration - to capture new activity configurations, the infrastructure should allow users to create, manage, share, distribute, fragment and delete activity configurations. To seamlessly move information between different applications or devices, the infrastructure should support the configuration of information on one or multiple devices for one or multiple users.

Service 6: Context Handling - to include the material world into the activity configuration, the infrastructure should provide mechanisms to include context-aware functionality into the activity configurations. This means that the infrastructure needs to provide context processors that can handle location tracking, embedded sensors and devices as well as other real-world readings.

Service 7: Interoperability - to mitigate the integration problem of using activity configurations across existing application-centric operating systems, the infrastructure should provide an operating system agnostic platform and protocol that allows developers to use the infrastructure across different types of devices. This will allow future inclusion of new platforms.

5.2.2 Architecture

The fundamental concept behind the NooSphere infrastructure is that it implements and abstracts these seven services into *activity systems* that provide access to distributed activity configurations. The activity configuration model includes *resources* (such as files, data and other information), *coordination structures* (such as users, roles and messages) and *configuration states* (such as application descriptions, connected devices or context information) making it a well suited match for a multi-user, multi-device and multi-location environment.

NooSphere is a flexible, dynamic and reusable infrastructure and programming framework that is based on the concept of *communicating activity systems*: by using a two-layered architecture and activity configuration model, different devices, applications and even entire distributed systems can be interconnected into a distributed web of systems. NooSphere is composed of (i) *NooCloud*, a cloud-based platform for data storage, persistence, interconnection and accessibility; and (ii) *NooSystem*, a local distributed system that provides discovery, pairing and ad hoc configurations of applications and devices (Figure 5.2). Both the NooCloud and the local NooSystem abstract all distributed services into *activity managers and systems*. Data, services and applications are thus not confined within one distributed system, but can be consumed in all interconnected systems through adaption. This infrastructure architecture allows for the construction of a wide range of domain specific systems (as conceptualized in Section 4.6 of Chapter 4) that can be interconnected in various ways and allows for advanced and evolving architectural patterns.

5.2.3 NooCloud

NooCloud is a cloud-based service platform that provides support for the persistence and distribution of activity configurations and their internal components. Using a modular approach, data storage, user management and event distribution are abstracted into an Activity Cloud Controller (ACC) that provides a central entry point for the cloud part of the infrastructures. Developers and local activity systems thus interact only with the ACC and only with entire activity configurations. Internally, the ACC splits up the model into data packages, user models and events, which are internally stored in separate controllers that follow the Front Controller design pattern. Each controller is individually responsible for CRUD[1] operations for each sub part of the activity configuration model. First, the Storage Controller (SC) provides support for the storage of all infrastructure

[1] create, read, update and delete

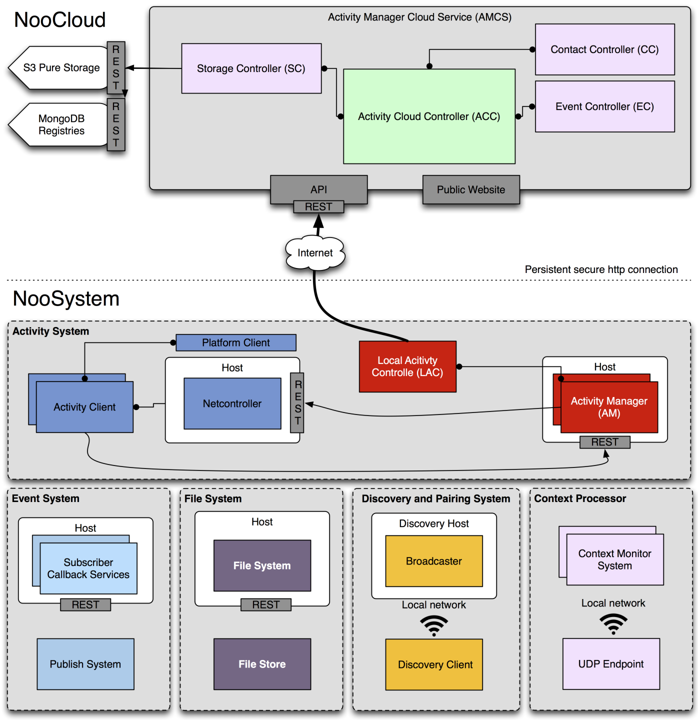


Figure 5.2: The architecture of NooSphere is composed of two components: (i) NooCloud, a cloud infrastructure that supports data storage, event distribution and activity management, and (ii) NooSystem, a dynamic distributed system that supports cross-device activity management, file and event distribution as well as a discovery and pairing mechanism.

primitives such as files, device descriptions, user models, events and also entire activity configurations. The SC stores these primitives in either a registry (implemented though a NoSQL database) or in pure storage (implemented as Amazon binary blob). The registry only stores the ID of each primitive as well as a pointer to the location of the entire primitive in the binary blob. Because activity configurations are flexible definable models, they are saved in pure storage. Second, the Event Controller (EC) handles the connections made through the ACC, and stores information about which users and devices are connected. When changes in the ACC or SC occur, the EC pushes notifications down the connections to update local devices. The EC uses a key-value store to cache and manage events and supports distribution of the models. Finally, the Contact

Controller (CC) handles user models by inspecting the role of a user in a specific configuration as well as by keeping track of connections between users. The CC thus checks and verifies which users should be able to access (a part of) the activity configurations. The ACC exposes all activity configurations as a REST service over a persistent HTTP connection that allows for real-time cloud-based publish-subscribe support. Using the API, developers can create, read, update and delete activity models and containing resources through pure REST HTTP requests.

5.2.4 NooSystem

NooSystem is a local dynamic and distributed service-based infrastructure that supports activity configurations across devices and users within one specific locally deployed activity system. This part of the infrastructure uses a flexible *just in time* service model in which *activity services* can be spawned ad hoc to create connections between devices and users. NooSystem encapsulates services such as distributed event management (over websockets), file and resource synchronization (through file system integration), discovery and automatic pairing (implementing Bonjour and WSDiscovery) and a distributed context processor (that integrates with hardware that connects over serial ports). NooSystem is internally composed of two layers: (i) the implementation of the core distributed services and (ii) the activity system, which merges all underlying services into an activity configuration abstraction layer that provides two types of *activity configuration containers*: the activity manager (AM) and activity client (AC). A master AM is typically connected to the ACC in the cloud as it functions as a local cache of a set of activity configurations that are used by the local activity system.

The AM stores activity configurations locally and acts as a mediator between the ACC and the local systems and devices. The AM can run in isolation mode when it is used on only one device, but can spawn HTTP REST services that provides distributed access to the internal functionality. The AM then broadcasts its existence over the local network and allows other slave AMs or ACs to connect. Once connected, all changes to the activity configurations are sent to all connected devices, thus keeping a synchronized model. Activity Clients are lightweight

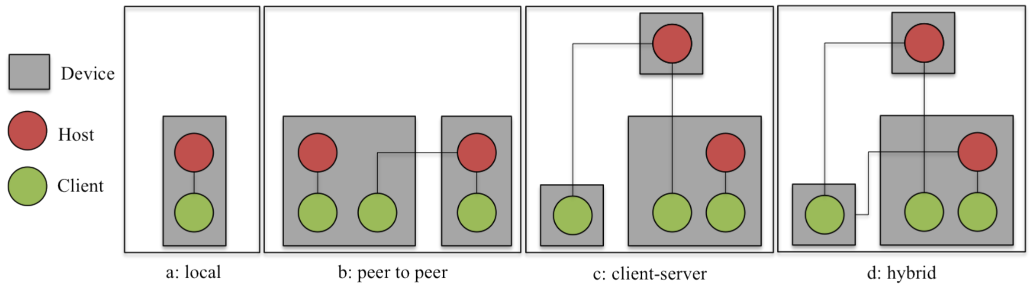


Figure 5.3: The local infrastructure NooSystem, can be deployed in different configurations: (a) manager and client on the same devices, (b) peer to peer connection between manager on different devices, (c) traditional client server approach with a manager on a dedicated device or (d) hybrid setup composed of both dedicated and local managers.

activity configuration containers that do not cache any local data, but rather use an AM to access (part of) the activity configurations. Using a combination of ACs and AMs, local activity systems can be created, modified and changed ad hoc. As depicted in Figure 5.3, the manager and clients can be connected in a wide range of architectures allowing for a flexible mechanism to distribute activity configurations within or across devices. Moreover, the infrastructure allows for runtime changes in the type of activity container (manager or client) and runtime connection with other nodes in the activity system. This allows developers to design activity-centric systems that can allow for ad hoc creation of device configurations that can change over time and be reconfigured by the end-user.

5.2.5 Contributions

Paper **C2** provides an in depth evaluation of the infrastructure by describing how three distributed systems (Figure 5.4) were built using the framework. This approach of building example applications on top of a programming framework and infrastructure has been proposed as a robust method to demonstrate the *stability*, *performance* and *feasibility* of the infrastructure [56]. The main contribution of NooSphere is that it dramatically reduces the development effort of distributed Activity-Centric systems that allow for the ad hoc composition of device- and user configurations over different locations. It uses a combination of a modern cloud platform for scalability and connectivity and merges that platform with a local dynamic infrastructure that can be run on a single machine, thus, eliminating the need for any type of special servers. NooSphere encapsulates complex network code, discovery mechanisms, file and model replication into a transparent activity-centric platform that can be accessed over HTTP REST. The architecture of the system allows for a broad range of setups ranging from local, client-server to peer-to-peer or even hybrid setups. Noosphere provides a robust technical framework for the design, development and deployment of activity-centric systems that apply concepts and ideas from the activity configuration framework. Various parts of NooSphere were used for different stages in research projects discussed in the remainder of this chapter and the infrastructure is currently still being used for various research projects in our lab that require synchronization of complex data structures. The infrastructure is released as open source software at https://github.com/StevenHouben/NooSphere.

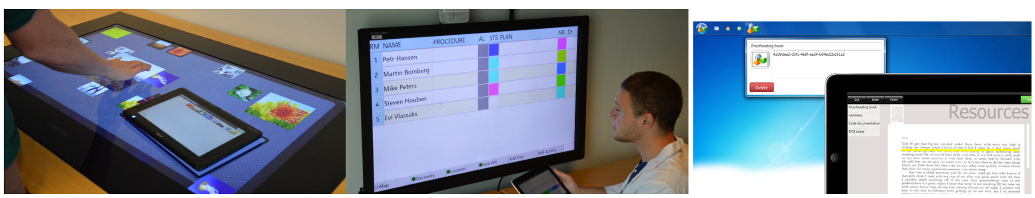


Figure 5.4: NooSphere was evaluated by designing three distributed systems that employ different aspects and services of the infrastructure [C2].

5.3 Co-Activity Manager

Modern knowledge work consists of individual work that is often part of larger collaborative activities that spread across a number of different users across multiple locations. Knowledge workers, thus, employ a range of personal computing devices, such as a desktop or laptop, tablets and phones, to perform individual tasks that are dependent and connected to the work of other team members. Despite the fact that there is a high degree of collaboration between team members, users still prefer to organize and tailor their part of the collaborative work according to their personal preferences. Going back to our knowledge worker team, introduced in Chapter 3, and depicted again in Figure 5.5, we observed that these types of multidisciplinary teams continuously collaborate on a number of partially overlapping activities that require some form of synchronization and articulation. There are a number of articulation, collaboration and configurations problems related to handling these parallel semi-shared activities. Current desktop interfaces provide very little support for these problems and essentially force users to employ and appropriate a range of external tools and methods.

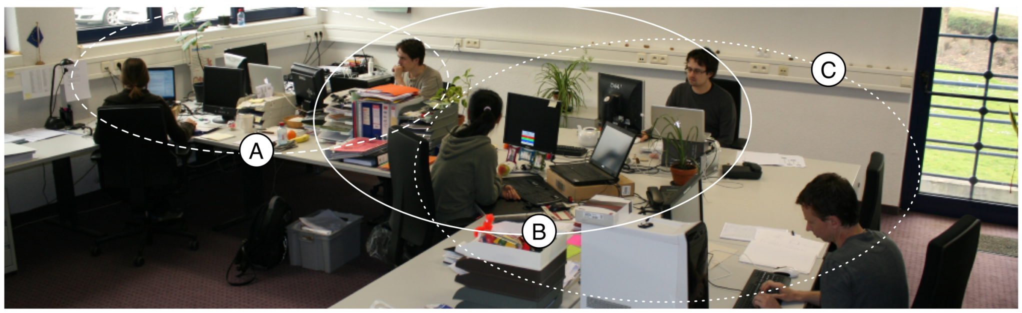


Figure 5.5: A group of knowledge workers that use a range of devices to collaborate within several sub-activities.

To mitigate these configuration, articulation and coordination problems in ad hoc knowledge work, we designed co-Activity Manager [C1], a desktop interface that extends the standard Windows 7 shell with an activity-centric layer that provides support for:

Activity-Oriented Workspace - The workspace allows users to organize applications, files and resources into activity configurations that are visualized and captured into virtual desktops that can be given an activity signifier in the form of a name, icon and color.

Activity Configuration Sharing - Activity configurations can be captured and sent to other users who can deploy the system on a different machine. This sharing mechanism allows users to 'grab' their work context and share that context with another user or device.

Activity-Centric Communication and Collaboration - co-Activity Manager includes collaboration processes, such as instant messaging and file sharing into the activity configuration abstraction, allowing users to control their collaboration lists for each specific activity configuration.

5.3.1 System Description

Co-Activity Manager (Figure 5.6) allows users to organize files, folders and applications into activity configurations, which are implemented as virtual desktops that are called *activity workspaces* (Figure 5.6A). All interactions with windows and files are confined within the workspace, and switching between different workspaces causes the system to repopulate the desktop with the windows and files that are associated with the activity configuration. This means that within each activity configuration, users can simply drop their files on the desktop. Inspired by earlier work [13, 165], activity configurations are represented to the user on an activity taskbar (Figure 5.6B) and activity start menu. Each open activity configuration is visualized through an activity signifier (a button with the name and icon). Clicking this signifier will cause the system to suspend the ongoing activity configuration and to resume the activity configuration represented by the signifier. The bar also contains a 'quick launch area' that can be used to auto-hide the bar, copy windows between different activity configurations or quickly create a new empty activity workspace. The start menu (Figure 5.6C) allows users to create new configurations, load existing configurations that are stored online or on the file system and manage the options of the application. Furthermore, through the start menu, the user can change the workbench, which is the set of all open activity configurations on the taskbar. This allows users to swap the entire activity bar based on changing context, such as switching between home and work activity configurations. The system also includes a context processor that uses the SSID of the Wifi to automatically switch between existing workbenches.

co-Activity Manager includes collaboration and communication processes into the activity configuration abstraction. It provides a built-in collaboration manager (Figure 5.6D) that allows users to select contacts that are relevant for that specific activity. By clicking the light-bulb next to the contact name, users are added as collaborators to the activity configuration. This implies that these selected collaborators will see the user as 'online', while other collaborators will see the users as 'offline'. Switching between activity configurations causes the system to repopulate the collaboration manager and to update the availability based on each activity configuration. This means that collaborators will see the user

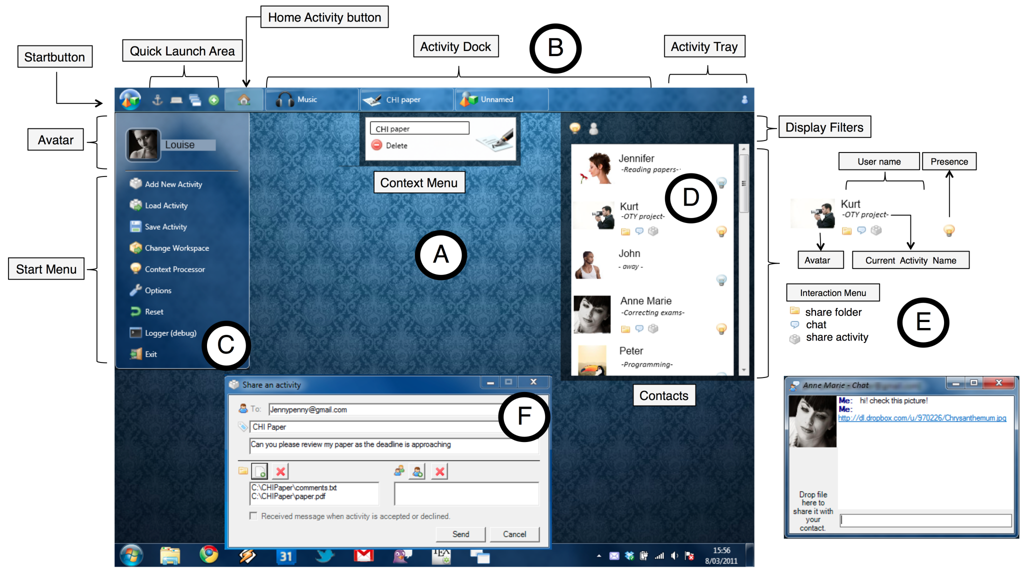


Figure 5.6: The interface of co-Activity Manager consists of (A) a per-activity workspace, (B) an activity task bar to visualize activities to the user, (C) an activity start menu to manage activity configurations and applications, and (D) a collaboration manager to interact and share with contacts. Each contact is visualized with an avatar, name and status field. The interaction menu (E) can be used to share a folder, chat or share an activity configuration (F).

appear 'online' or 'offline' depending on whether they are added as collaborator to the selected activity configuration. The collaboration manager provides filters that remove all non-related collaborators, thus providing users with a mechanism to contextually construct a collaboration list of each activity configuration. From within the collaboration manager, users can create shared file containers, start chat messages or share an entire activity configuration (Figure 5.6E). Sharing an activity configuration launches a special window (Figure 5.6F) which allows users to select which part of the ongoing digital configuration should be sent to the collaborator. The receiving user is notified that a new activity configuration was received and can select to accept or queue the configuration. Accepting the activity configuration causes the system to create a new activity workspace with the digital configuration provided by the user. This 'template' can then be appropriated and changed by the user.

Users can also attach a tablet or phone to co-Activity Manager. Tablets can run a discovery service and if the correct key is provided by the user, the device is attached to co-Activity Manager and visualized in the activity tray (as seen in Figure 5.4, outer right). If the tablet is successfully connected, users can drag and drop resources, such as files or websites, to the activity signifier (button), which will automatically send the resources to a viewer on the tablet. This mechanism allows users to easily fragment the resources of one activity configuration across multiple devices in use. The application on the tablet allows users to annotate, browse and organize the attached files; by clicking a button on the tablet, the resource is synchronized back to the desktop interface and visualized on the desktop.

5.3.2 Field Deployment

co-Activity Manager was deployed for a period of two weeks in a multidisciplinary software development team that consisted of five people (3 male, 2 female, mean age = 31). The team included users with different backgrounds including a historian, designer and three software engineers. All participants were instructed to use the system on their main machine as part of their daily work. During the two week period, participants were observed and interviewed regularly throughout the deployment. After the deployment was completed, participants were asked to complete a Likert-scale survey which was used as input for a semi-structured interview. Table 5.1 highlights the results of the 5-point Likert scale survey that ranged from "not at all useful" (1) to "very useful" (5).

Questions	*Min*	*Q1*	$\tilde{x}$	*Q3*	*Max*	*Iqr*
Usefulness cAM	3	3.75	4	4	4	0.25
Activity-centric concept	1	3.25	4	4.25	5	1
Sharing Activity Workspace	1	3.25	4	4	4	0,75
Activity-centric collaboration	3	3.75	4	4	4	0.25
Activity Cloud support	3	3.75	4	4	4	0.25

Table 5.1: The result of the 5-point Likert scale survey that was used as a basis for the interview. The table shows an overview of the minimum, maximum, median ($\tilde{x}$) and the interquartile range (iqr).

In summary, the study showed three main results. First, during the deployment we observed how the lightweight activity configuration mechanism allowed for a flexible and diverse organization of work. Some users would employ the activity bar for 'project-based' configurations, while others used it as to-do's, in which each activity configuration was created up front to represent the work they planned to do during the day. More general, we observed how users appropriated the activity configuration mechanism into a distributed life cycle (Figure 5.7) in which activity configurations were created, shared and used for different purposes. This led to three types of activity configurations: (i) long term configurations that were high level representations of work, (ii) short term or transitional configurations, that consisted of more focused units of work and (iii) ad hoc activities, which were used for quick operations that required a clean workspace. These types of configurations do not necessarily point to their temporal dimension but rather to their intention. Within the activity cycle, users would move between these different types of activities and shared these across the team.

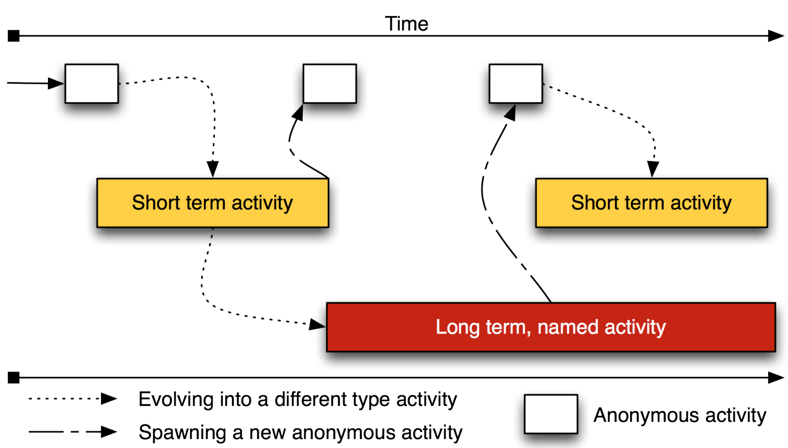


Figure 5.7: Activity Lifecycle - activity configurations are created as anonymous ad hoc configurations before disappearing or evolving into short or long-term defined configurations. From within these defined configurations, new ad hoc activity configurations would emerge, that in turn would disappear or evolve.

Second, Activity-Centric collaboration emerged in two distinct phases. Users would first share an entire activity configuration template with another user, who would be able to set up a similar working context. In this sharing phase, users would align the semi-shared activity configuration. In the second coordination phase, users would employ the built-in manager to collaborate from within the semi-shared configuration. This mechanism essentially allows users to create an individual activity configuration template that forms the basis of a semi-shared activity configuration that provides the team with a consistent distributed project. Participants described this mechanism as a 'workflow' that helped them bundle and organize temporal steps in the projects.

Third, the contextual contact list was perceived as a useful feature but also had a number of drawbacks. Frequent switching between configurations would cut of collaborators mid-sentence as the main user would suddenly appear 'offline'. Our study also pointed to a number of privacy and confidentiality issues. Because the collaboration system would automatically distribute the signifier of the ongoing activity configuration to all collaborators, potentially sensitive information could be leaked. In one instance, e.g, the name of a still undisclosed project that was used as name for one of the activity configurations, was unintentionally distributed among all contacts. This was especially a problem considering the fact that users often mixed both work and personal contacts into the collaboration manager (e.g., because they wanted to be able to chat with their wife or husband) meaning that information would leak outside of the organization.

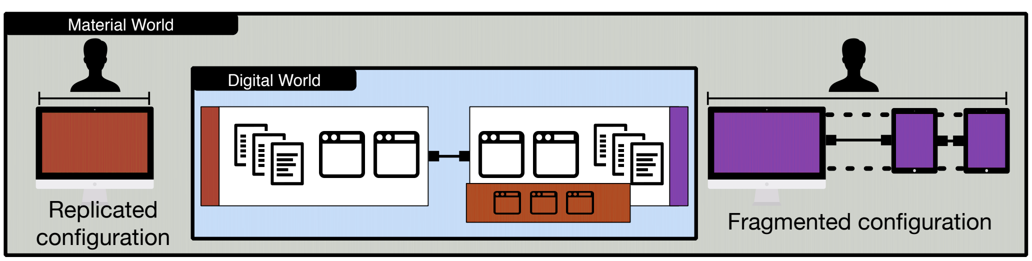


Figure 5.8: co-Activity Manager supports semi-shared replicated and fragmented activity configurations.

5.3.3 Contributions

As indicated in the overview in Figure 5.1 and 5.8, co-Activity Manager explored how activity configurations could be used in semi-shared replicated configurations that support the ad hoc nature of collaborative knowledge work. The main contribution of this work is the design of an activity-centric desktop manager that supports both personal and collaborative activity-centric workflows by integrating activity-centric collaboration and interruption management tools into one activity configuration representation. The novelty of this work is how *configuration*, *collaboration* and *articulation* tools are included in one central desktop interface. Fundamental to this approach is that it allows these tools to become an intrinsic part of the entire activity-centric workflow, rather than using external tools that need constant (re-)configuration. Furthermore, users can attach tablets and phones to these desktop configurations to support cross-device resource sharing. Our study shows how the fragmentation and replication of activity configurations create semi-shared setups in which individual users can maintain their individual workflow but are provided with tools that allow them to easily externalize these workflows to other team members. We identified a distributed life cycle of activity configurations in which the basic configuration is appropriated in a wide range of different practices that can be connected to one central collaborative workflow.

5.4 ActivitySpace

One of the core observations in the co-Activity Manager study [**C1**] but also other previous work [139, 8], is that space often plays an important role in ad hoc collaboration between people. Although collaborative mechanisms, such as the one presented in the co-Activity Manager system, are valuable for structured collaborative work, they fail to capture the high degree of physicality of human-computer interaction. Especially with an increasing rise of the amount of devices people own and use, the physical form factor but also the spatial relation between users and devices plays an important role in how people share information. Going back to the desk pictures (Figure 3.1 in Chapter 3) we observed how people spread their devices and physical artifacts, such as paper and sticky notes, over a physical space using highly organized strategies. Analyzing these strategies resulted in a 'meta desk model' (Figure 5.9) that is discussed in depth in paper **C3**. This model, based on an analysis of 15 desks, shows how the space is organized into specific zones that are allocated to (i) active work and interaction

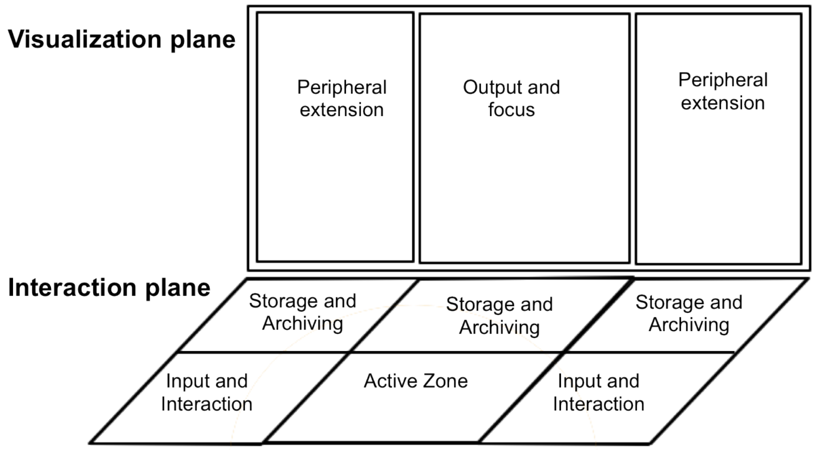


Figure 5.9: The zones defined in this meta-desk were found in all analyzed desks.

with artifacts, (ii) storage and archiving and (iii) focused and peripheral output of information. This analysis is in line with spatial models, such as the Situative Space Model [139] and the more generic Proxemic Interaction framework [8], suggesting that the function of these zones can be extended to support digital exchange of information or even to create device configurations.

5.4.1 System Description

ActivitySpace [C4] is an implementation of the *configuration space* concept discussed in Chapter 4 and is designed to reduce the amount of configuration work when interacting with a set of resources that are distributed over a number of different people and devices in an ad hoc context (Figure 5.10). The system allows for activity-centric resource and device management. Using an interactive surface as a spatially stable configuration space, users can connect different heterogeneous devices to the ongoing activity configuration sessions by simply placing them into the space. Using computer vision, devices are detected and added or removed to the distributed activity configurations. The first device that is placed in the public configuration space becomes the master device and takes ownership over the space and allows the space to access the activity configurations on the device. When the configuration space detects a new device, it adds a visualization to the space to indicate a physical connection. If there is already a master device, the newly detected devices are automatically configured into slave mode, and attached to the activity configuration of the master device. Removing slave devices disconnects them automatically from the master device and removes access to the activity configurations. If the master is removed, the first attached slave device becomes the master. If no slaves are in the space, the session is terminated. The device visualization provides two icons that (i) allows users to pin the device to the space, meaning that it can be removed and carried around without losing connection (Figure 5.11D), (ii) indicates whether the underlying connection is successfully established. Within the configuration space, activity configurations are fragmented across all connected devices. These devices can be from one or multiple people, thus, supporting both individual and collaborative workflows.

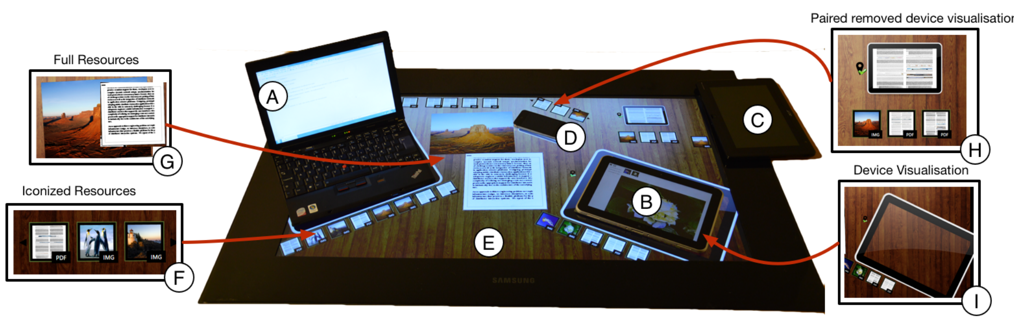


Figure 5.10: *ActivitySpace* supports activity-centric resource management spanning across (A) laptops, (B-C) tablets and (D) phones using an (E) interactive desk as mediating configuration space. The configuration space visualizes all devices (H-I) and their allocated resources (F-G), that are part of the current activity. Additional devices can be added by placing them in the configuration space. Moving resources to a device is done by drag and drop. When the user suspends an activity, the entire space configuration on all linked devices is stored. When the activity is resumed, the entire configuration is reestablished.

The system supports three types of devices: (i) laptops, running a modified and newer version of co-activity Manager (Figure 5.10A), (ii) tablets, with a resource viewer that allows for active reading (Figure 5.10B-C), and (iii) phones, that run a special notification system (Figure 5.10D). Each device can be used as master, in which all open activity configurations can be inspected, created and used, or in slave mode, in which it can only access resources that are allocated by a master device. If more than one device is located in the space, the resources that are allocated to each device are visualized on the space as icons that are attached to the device frame (Figure 5.10H). These icons are touch-enabled and can be detached from the allocated device and be used inside the configuration space using a range of interaction techniques. The techniques include: double tapping to create carbon copies of the resources (Figure 5.11A); inspecting the content of the iconized resource by semantically zooming (Figure 5.11B); docking the resources to the side of the space in case they are not used by any device but simply need to be stored in the space (Figure 5.11C), and drag and drop iconized resource to attach it to a device (Figure 5.11E). Attaching a resource to a device sends the resource automatically to that device and shows it on the screen. Resources can be either attached to a device or simply be used or spatially organized and stored in the configuration space.

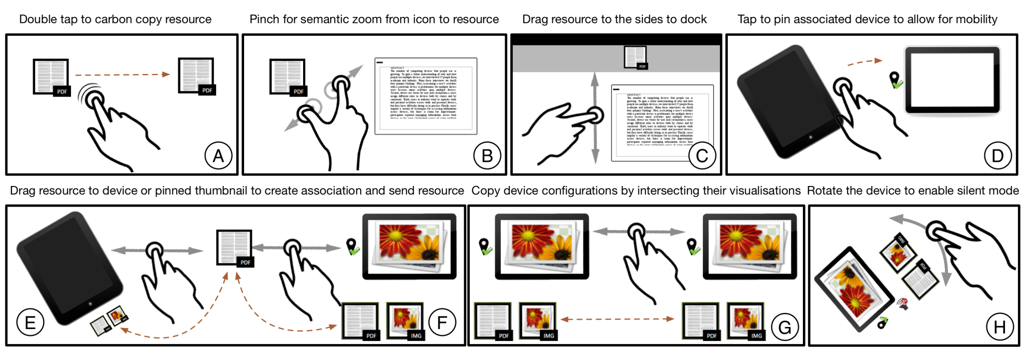


Figure 5.11: The configuration space provides a number of basic techniques to interact with resources and devices. User can copy (A), scale (B) or dock (C) resources on the space or drag them to another device (E). The space also allows users to pin their device (D), copy entire device configurations (G) and use spatial orientation to configure device properties (H).

Switching between activity configurations on one of the devices (e.g., the activity bar of the laptop) will update the entire configuration space, including the physical layout of all icons and device visualizations. This means that activity configurations store the entire cross-device allocation of resources, and users can easily switch between these spatial configurations. Creating a new activity configuration clears all resources on the configuration space and creates an empty blank canvas, similar to a new physically distributed spatial virtual desktop. To allow for quick multi-user fragmentation of activities, entire device configurations can be copied by bumping physical devices together, or intersecting their iconized visualizations. Finally, the space also uses spatial orientations to update functionality on the devices: the phone, e.g., can be put in silent mode by simply rotating the screen upside-down.

5.4.2 Lab Study

ActivitySpace was deployed and evaluated in a scenario-based lab evaluation with 9 users (two female, seven male, mean age = 30) from different backgrounds including clinical work, software development, business, and research. After introducing participants to the system and concept, they were asked to conduct a scenario which revolved around collecting, comparing, selecting and sharing resources as part of the design of a new website. The scenario (that is detailed in **C4**) was created to explore (i) device coupling and decoupling, (ii) cross-device resource allocation, (iii) activity switching, (iv) multi-user interaction, (v) interruption management, and (vi) local mobility. After completing the scenario, participants were asked to complete a Likert scale survey which was used as input for a semi-structured interview. The entire experiment was videotaped and detailed interaction logs were captured during interaction with the system. Figure 5.12 highlights the results of the 5-point Likert scale survey on the general usefulness of the different aspects of the system.

The study first of all showed that users were very quickly able to perform complex cross-device information curation operations that in current systems are hard or even impossible to perform. Users valued the idea of using the open physical space between devices for (i) externalization of resources in use and (ii) visually moving information between devices. One participant mentioned:

> *– "I really like the idea of using the empty space between devices to show what's on them. Most of my devices are currently already on or around my desk, so why not use this space." – P4*

Already during this short evaluation, we observed how users quickly took ownership over the desk. The concept behind a configuration space is that it is a public mediating device that needs explicit permission to access information on

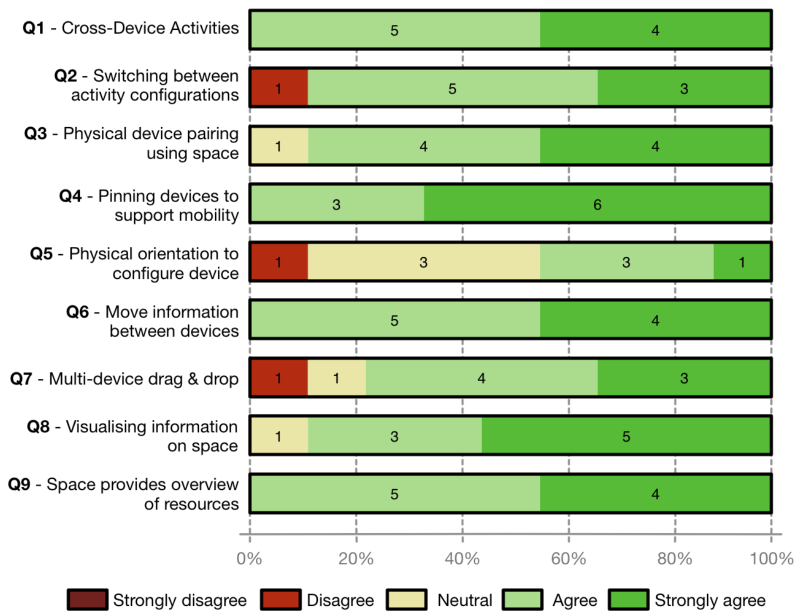


Figure 5.12: The results of a 5-point Likert scale questionnaire on the usefulness of the different parts of the system. The numbers in the bar represent the amount of participants.

the user's device. However, participants argued that, similar to normal desks, people should be able to 'take ownership' over the desk and customize and claim the configuration space for personal work. During the scenario, we observed a number of recurring patterns of use. Users actively pushed and pulled devices around to optimize the space for using resources on both the devices and the space. Unused devices were frequently placed on the edge of the configuration space, outside of the tracked zone. We did, however, notice that users would 'pin' these devices, just in case they wanted to use them anyway. Interestingly, some users even appropriated inactive tracked devices as physical folders: they would allocate a set of related resources to one device, simply to be able to move this physical resource container around and not deal with individual resources. As predicted by our 'meta-desk' and seen in the plot of touch interactions in Figure 5.13, users often moved devices to the front of the desk into the 'active zone', while pushing secondary devices to the back. Furthermore, we observed circular movements of devices indicating that users maneuvered devices around resources in focus.

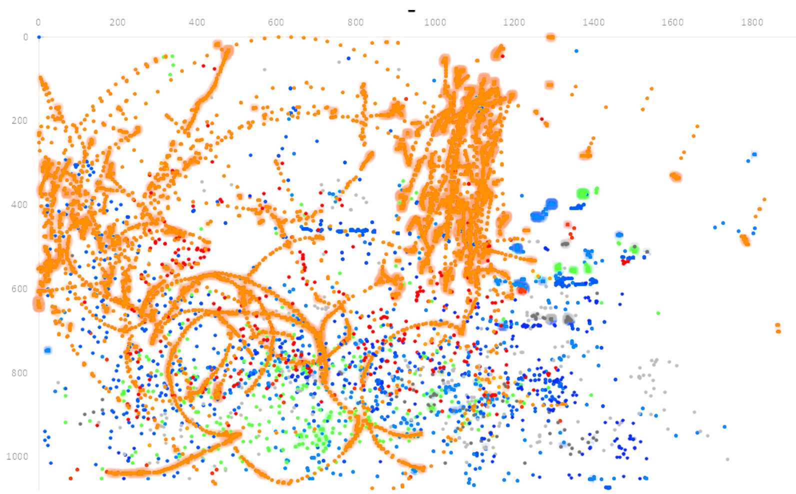


Figure 5.13: Touch interaction data of 9 users plotted on the desk space. The data shows circular movements of resources around visualizations (orange color). Other colors represent the touch input of participants which is primarily focused in the front middle of the desk.

Fragmenting activity configurations between users is done using a social sharing model in which users allow other users to place their device into the space. Once connected, both users can agree and physically and visually share resources within the shared activity configuration. This approach mitigates a number of technical and disembodiment challenges in ad hoc collaboration, but also introduces a social public—private tension. We observed how almost all participants created implicit zones on the space that were allocated for personal use and zones allocated for devices from 'visitors'. These visitor zones were created using the spatial positioning and proximity of the devices of the owner. Although this sharing approach creates a very stable social mechanism for information exchange, it also opened a discussion about privacy and accessibility, pointing to situations when, e.g., the owner was not physically at their own desk.

ActivitySpace currently only provides support for the visualization and fragmentation of one activity configuration, the one that it is mediating. Participants, however, almost unanimously argued that this should be extended to multiple activity configurations and even to more advanced 'meta' functions, such as a cross-device task bar that would allow switching applications across devices, or a cross-device sound and display property manager, and even a centralized notification system that bundles interruptions from all devices and visualizes them in the space. It was clear that participants suggested offloading configuration work of all devices to one mediating device to keep the interaction with all other devices focused on the activity at hand.

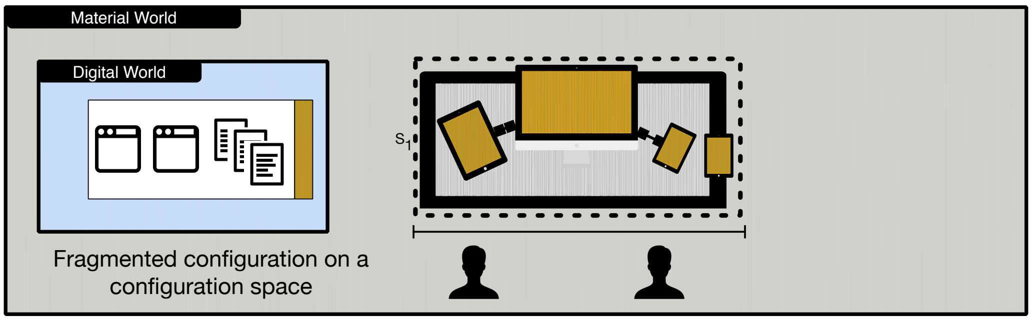


Figure 5.14: ActivitySpace uses a configuration space to mediate fragmented activity configurations.

5.4.3 Contributions

As seen in Figure 5.1 and 5.14, the central purpose of this work is the operationalization of the *configuration space* concept. The contribution is a novel activity-centric cross-device information management system that uses an interactive surface to mediate and visualize interactions within an ecology of heterogeneous devices. This approach proposes to use a stable physical space, augmented with an active tracking zone, to allow users to create device configurations, perform complex cross-device information curation activities and set up shared activity configurations using a social sharing model. Our study demonstrates that users are able to perform complex cross-device information management tasks using activity configurations as a computational construction. We highlight a number of interaction patterns that show how users appropriate the space and interact with an ecology of devices. Finally, our study indicates that the social sharing model is conceptual very easy for users, but holds a number of potential limitations, related to privacy and accessibility.

5.5 WatchConnect

After the widespread success of *mobile devices*, such as smartphones and tablets, a new generation of *wearable devices*, such as smartwatches and interactive glasses, are increasingly finding their ways into the consumer market. Smartwatches provide people with a lightweight and immediate access point to their digital information, such as messages, notifications and other resources. As standalone device, smartwatches provide users with easy access *on the go*, but combined with other devices, they open up an interesting design space for novel interactions that allow for seamless configuration of resources across devices. The current generation of smartwatch technology, however, only provides limited support for exploring cross-device applications and systems. Because of this, there are only a few previous explorations into these types of applications and techniques [46, 120]. Smartwatches have the potential to become a central mediating access point to users' digital information cloud. We see smartwatches as a wearable portal or key that can be leveraged to configure and move information within device ecologies, using physical interaction with the watch hand. By leveraging the sensors of the smartwatch, the watch hand can be elevated to a reconfigurable instrument [24] that can modify digital information and devices.

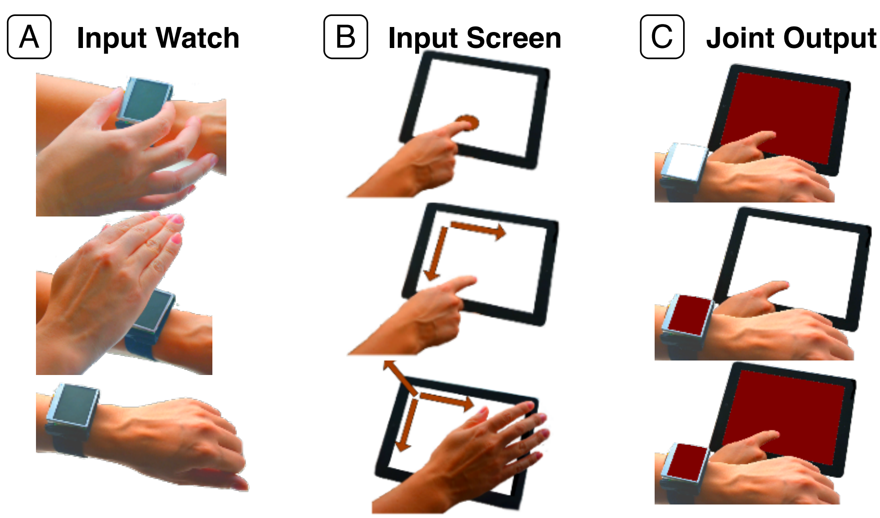


Figure 5.15: The interaction and configuration space of a watch-screen setup consists of (A) watch input, (B) screen input and (C) a joint output space.

The interaction or configuration space between the watch and any interactive display is created by blending the input and output spaces of both devices (Figure 5.15). First, the watch allows for three types of input: (W1) on the watch using a touch screen, interactive bevel or watchstrap, (W2) above the watch by sensing the proximity or distance and (W3) using the relative three dimensional position of the watch using a built-in motion sensor. Second, the interactive display also supports three types of input: (S1) identification of the touch point in two dimensional space, (S2) physical multi-touch input in two dimensional space and (S3) physical movement in the three dimensional space around the screen. Finally, the combination of screens on both the watch and display allows for three types of joint output space: (O1) output of the distributed interaction on the display, (O2) output of the distributed interaction on the watch and (O3) output of the interaction distributed over the two displays. Using a temporally synchronized combination of these input and output spaces provide users with a temporal interaction framework that allows them to easily configure and move resources between different devices.

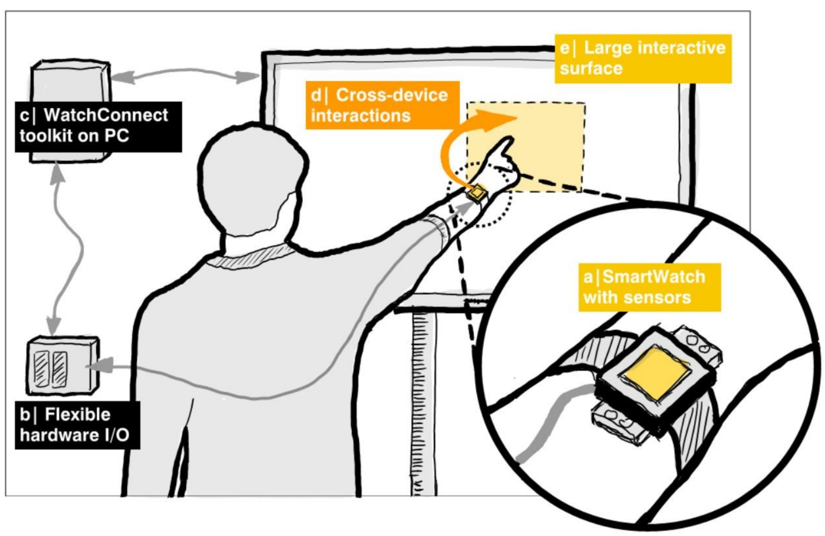


Figure 5.16: The different components of the WatchConnect toolkit.

5.5.1 System Description

To build and deploy watch-centric cross-device applications and systems that allows for the fragmentation and distribution of activity configurations across devices, we designed and implemented the *WatchConnect* [**C5**] toolkit (Figure 5.16). The central purpose of the toolkit is to provide an event-driven and extendable platform for the rapid prototyping of watch-centric cross-device systems and interaction techniques. The toolkit consists of wired prototyping smartwatches that are equipped with sensors and a programmable microprocessor (Figure 5.16A) that can be accessed through a flexible and extendable hardware layer (Figure 5.16B). WatchConnect also includes a software development platform providing user interface components and a rich set of input and output events and gestures (Figure 5.16C), that facilitate the rapid prototyping of cross-device interactions (Figure 5.16D) with other interactive surfaces (Figure 5.16D). The toolkit is designed to support the rapid prototyping of systems that fit into the aforementioned distributed interaction space between a smartwatch and an interactive screen. Its flexible hardware platform allows developers to easily add new sensors both on the physical device and in the software. By abstracting all the complex hardware code into high level objects and event, the toolkit provides a low threshold for developers to use hardware, gestures and postures through machine learning and complex distributed software designs. Moreover, the toolkit integrates with the NooSphere infrastructure, thus allowing for distribution of configurations across multiple users and devices.

The toolkit is built around a wired prototyping watch emulator that resembles a real watch but is composed of a miniature screen, a number of touch and motion sensors and an integrated programmable microprocessor. The default watch probe included in the toolkit (seen in Figure 5.17) is built around an Arduino microprocessor and contains a light sensor, two infrared proximity sensors, an 8 channel capacitive touch sensor, a six-axis MEMS motion tracker (gyro + accelerometer), a RGB led, a flexible force sensing potentiometer and a 2 inch TFT display. The hardware components are soldered on a PCB which slides into a 3D printed enclosure that can be worn on the wrist. Watchprobes are connected to the main devices using a special USB/VGA connection.

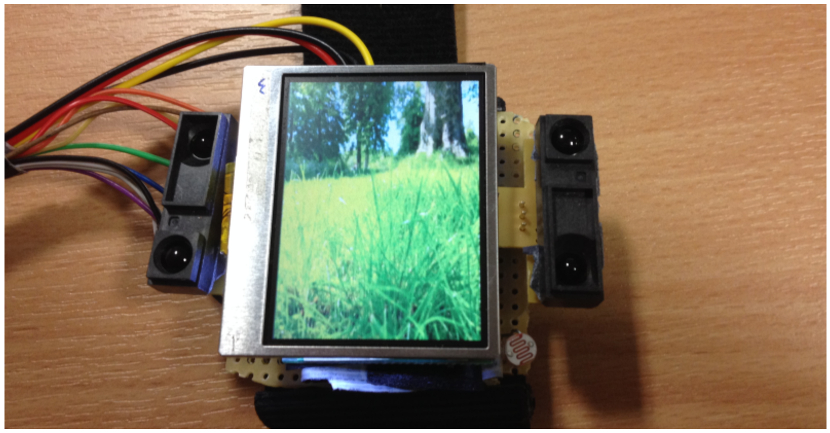


Figure 5.17: The standard watchprobe used in the toolkit.

To use the flexible prototyping platform, *WatchConnect* includes a software development platform and a watch runtime environment. The software platform integrates all sensors and serial input into a number of high level objects and events that can be used in a Visual Studio C# environment. The toolkit integrates with the standard Windows Presentation Foundation (WPF) tools and allows users to create drag and drop cross-device applications using the Xaml design language. The watch runtime is a configurable runtime environment that can be launched on the actual watch or in a native Windows window to simulate the watch output in case no physical device is connected. The software stack includes modules for machine learning to support three dimensional gestures and postures, a serial processor that provides event-based access to low level sensor output and an extensible input management system. By default, the toolkit includes numerous events, such as mid-air swipes or hover events above the watch, touch events on the bevel, screen and strap of the watch, and gesture motions with the watch hand itself. The toolkit provides an easy to use API to access and extend all objects and events in the software framework. Finally, the toolkit includes visual editors and tools to monitor, debug and capture sensors, touch events and machine learning detections.

Technique	Application	Watch Input: W1: Touch	W2: Above	W3: 3D Move	Screen Input: S1: Identify user	S2: Multi touch	S3: 3D Space	Output: O1: W feedback	O2: S feedback	O3: Distr. UI
Touch & Swipe	Data transfer	✓	✓	☐	✓	✓	☐	✓	✓	✓
Touch &Twist	Authentication	☐	☐	✓	✓	☐	☐	☐	✓	☐
Pose & Touch	Privacy	☐	☐	✓	✓	☐	☐	✓	✓	✓
Touch & Push	Navigation	✓	✓	☐	✓	☐	☐	✓	✓	✓
Connect & Gesture	Game	☐	☐	✓	☐	☐	✓	☐	☐	✓
Gesture & Touch	Reading	✓	☐	✓	✓	✓	☐	☐	✓	✓
Touch & Beam	UI composition	☐	☐	☐	✓	☐	☐	☐	✓	☐

Figure 5.18: An overview of 7 systems built on top of the standard toolkit.

5.5.2 Applications

To explore how the toolkit supports instrumental fragmentation of activity configurations using a smartwatch, we designed 7 interactive applications and interaction techniques that explore different realistic scenarios for the use of a smartwatch as mediating device while interacting with a standard laptop with touch screen. Figure 5.18 provides an overview on how each of the applications uses the toolkit. Paper **C5** provides an in depth overview and discussion of each application, but I will here detail two examples of how activity configurations can be fragmented across devices. All applications use only the default software and hardware stack of the toolkit and do not implement any custom functionality.

Resource Management

As described in Chapter 3, one of the core problems in distributed interaction is providing fast, intuitive and flexible ways to support the transfer of files and resources between different devices in use. A large body of research, including [118, 146, 159], have explored different ways to support seamless cross-device data transfer. These techniques can be extended using a smartwatch as a *wearable mediating storage device* that allows users to easily fragment the resources of an activity configuration across devices by physical and instrumental interaction. The *touch and swipe* technique presented in this approach allows users to physically touch the display with the watch hand to create a data connection. To distinguish between touches done with the watch hand and the other hand, the system shows a colored rectangle on the screen only if a watch hand touch is detected. The rectangle fills up (like a progress bar) over a period of two seconds to give users a time-window in which the activity configuration on the watch can be fragmented and shared with the display.

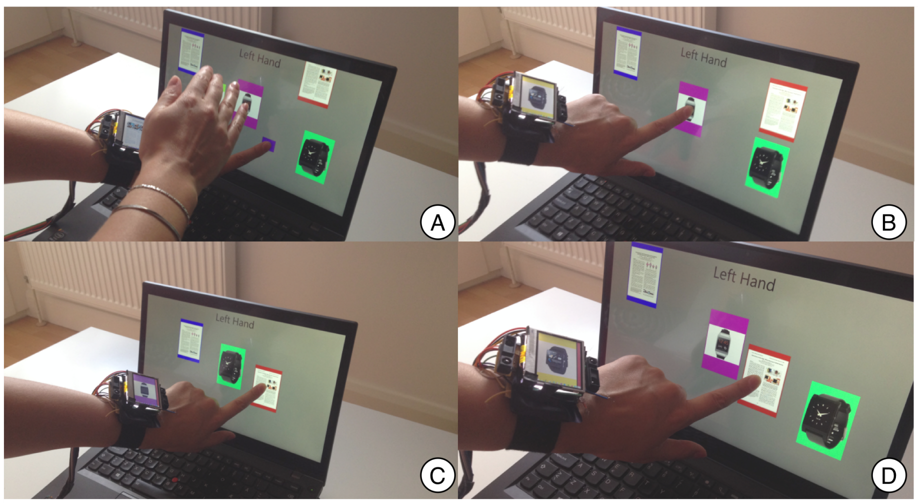


Figure 5.19: Application that detects watch hand touch input and allows users to move resources to the display using mid-air gestures.

After the time window is passed, the touch is degraded to a normal touch input. The color of the rectangle is the average color of the resources on the watch to represent the possible connection to the user. During the two second time window, users can perform a left-to-right swipe above the watch face. This swipe will send the resources from the watch to the touchpoint on the screen (Figure 5.19A - B). The resources can then be manipulated and moved on the touchscreen of the display. Users can toggle between different resources of the activity configuration on the watch by tapping the wristband of the watch. When users touch existing resources on the display with the watch hand, the system visualizes a signifier on the watch (Figure 5.19C - D) to reveal that the resources can be used and sent back into the watchface. During this visual connection, users can perform a right-to-left swipe above the watch to send the resource back to the watch, hence removing it from the display. During general interaction, the display can thus distinguish between watch hand and normal hand input to allow for a rich set of interactions.

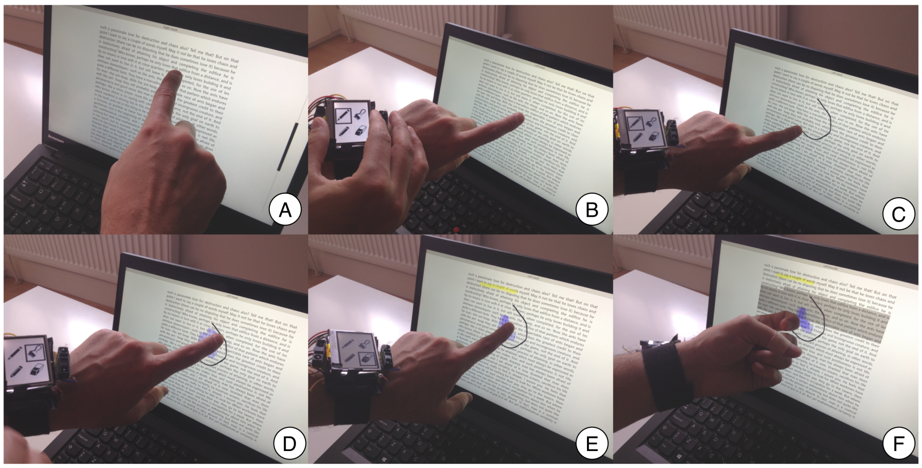


Figure 5.20: System that utilizes the watch hand as an instrument to interact with a menu-less active reading application.

Active Reading

Previous work and our analysis in Chapter 3, demonstrate that second screen devices are often used for active reading or browsing of news and documents. Most of these applications are often hard to use and heavily based around menus that allow users to switch between browsing, annotating or sketching. We implemented the *gesture and touch* interaction technique to provide a fluid menu-less interaction with an active reading application. The core concept behind this application and technique is that the watch is used to *configure the hand* into a specific *instrument* [24] that can be used to interact and modify the content. The non watch hand is used for passive browsing, while the watch hand is used for active input or modification of the documents. Using the non watch hand, users can scroll through the document using default touch and swipe gestures (Figure 5.20A). The watch hand can be configured into four different active modes by touching the capacitive bevel of the watch (Figure 5.20B). These modes include: (i) black marker for sketching (Figure 5.20 C), (ii) a translucent brush to mark entire regions or paragraphs (Figure 5.20D), (iii) a yellow marker to mark sentences or words (Figure 5.20E) and (iv) an eraser. Furthermore, by leveraging the sensors and machine learning techniques built into the toolkit, the system can distinguish between different finger input or hand posture of the watch hand. This means the display can recognize the difference between touching the display with the index finger, pinky, thumb or knuckle. In this application, e.g., touching the screen with the knuckle of the watch hand allows users to select and copy text (Figure 5.20F). The activity configuration representation is, thus, fragmented between the watch face, that provides the tools to reconfigure the hand into an instrument, and the screen, which shows the resources in use.

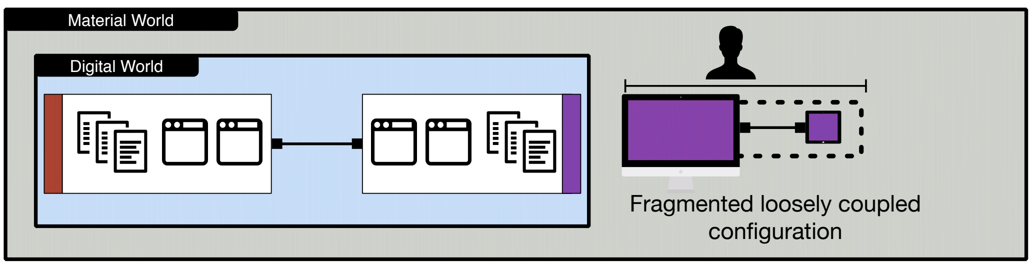


Figure 5.21: WatchConnect supports fragmented activity configurations using a watch.

5.5.3 Contributions

As indicated in Figure 5.1 and 5.21, WatchConnect and the example applications demonstrate how a smartwatch can be leveraged to become a mediating device that elevates the users' hand into a reconfigurable instrument that can be used to fragment activity configurations and their content across screens through direct manipulation. The contribution of this work is twofold. First of all, because there is currently no standard technical solution to prototype these types of interactions, we presented a novel approach for rapidly prototyping and designing smartwatch-based cross-device applications and interaction techniques, using simulated hardware and software. The WatchConnect toolkit abstracts complex hardware management, sensor fusion, machine learning algorithms and input and output management in high level objects and events that are easy to use and integrate with an existing visual design language for interface design. The toolkit provides developers access on all layers of the abstractions, thus, also providing access to the raw sensor data or allowing for the design of new machine learning approaches. Second, using this toolkit and the NooSphere infrastructure, we designed and built 7 applications and interaction techniques that are focused on exploring different approaches to fragment resources from one activity configuration across a watch and interactive display.

5.6 Hybrid Patient Record Device

As described in Chapter 3, the medical record is the most important artifact in medical work as it used to organize patient information, communicate relevant information with other wards and clinicians, and coordinate complex treatment procedures. In recent years an increasing amount of effort has been put into designing Electronic Health Records (EHRs) as a replacement of the classic Paper Medical Record (PMR), as they increase quality of care, accessibility and standardization across wards and hospitals. However, by attempting to replace the classic paper records, the intrinsic advantages and affordances (such as handleability, manipulability and portability [114]) of paper-based interaction, that fit very well with nomadic work in hospitals, are also removed. Furthermore, EHRs are often complex applications that force standardized workflows and mechanisms. Because of this, the EHR often operates only as a passive information repository that is supplemented with a PMR that holds informal documentation, as part of a *working record* [65] or *transitional artifact* [47]. The reality of today is that many hospitals still operate using a dual record that consists of both an EHR and a PMR. This dual record, however, introduces extra configuration work related to synchronizing the content of the digital and paper record and finding and handling the dual record in a nomadic workflow.

Prior research on medical work (e.g., [47, 137]) and our field study [**C7**] point to a range of challenges associated with handling medical records. At its core, these challenges are tied to clinicians' need to handle, align and coordinate physical and digital information simultaneously. Rather than designing for the 'paperless hospital', there is a need to design for the parallel management of both paper and electronic medical records, thereby supporting a *hybrid* medical record. Inspired by our field study [**C7**] and prior work, we propose the following three design principles to support a dual or hybrid patient record:

D1 Dual Use - Because the paper and electronic version of the record are almost always used simultaneously, setting up and removing the connection between the paper record and a device representing the electronic record should be instant and easy. Both representations should be usable separately, without any changes to their original purpose or use. Since the paper record is used to identify the patient case, the hybrid record should use this patient context to load and visualize the correct data. To facilitate the usage of the double record, it should be integrated with existing practices, devices and technology.

D2 Recognizability - To support easy identification and recognition of a patient record (e.g., in a cluttered office space) the patient record should be able to relay and display various kinds of status and awareness information. Temporal visual and auditory cues (similar to the analogue affordance of, e.g., sticky notes) should be supported to provide clinicians with an easy and fast configuration mechanism for self-reflection or coordination with other clinicians.

D3 Mobility - The patient record should support the nomadic workflow in hospitals, meaning that both the electronic and paper representation of patient data should be available in a portable and traceable form factor. To support clinicians in finding and managing the location of the record, the supporting infrastructure should support location tracking and remote access to the state of the paper record. Additionally, the location should be used to ease information retrieval.

5.6.1 System Description

To support this dual record in nomadic work, we introduce the notion of a Hybrid Patient Record (HyPR) that consists of (i) a classic paper record, (ii) a tablet, phone or other mobile devices that provides access to the EHR and (iii) a HyPR device [**C6**], a mediating sensor platform that augments the paper record with a notification system (color and sound), location tracking and ad hoc integration to a tablet that provides access to contextual relevant electronic patient data (Figure 5.22). The dual record is modeled as an activity configuration that is distributed over a device and a surrogate that connects the paper record to the digital workflow. The HyPR device, the surrogate, is a rectangular plastic

Figure 5.22: A Hybrid Patient Record (HyPR) device augments the paper patient record with color configuration and location tracking, while allowing clinicians to pair a tablet which shows the digital information associated with the paper record.

plate with the same width and height as the paper record. The PMR is attached to the plastic plate using metal clips to create a permanent fusion to the device. Each paper record has one unique attached HyPR device. Clinicians can interact with the HyPR by placing a tablet or phone on top of the augmented paper record. This causes the underlying infrastructure to pair the mobile device to the record and push the correct activity configuration, in the form of an EHR, to that mobile device. Any changes made to the record, are updated and synchronized in real time with all other devices. The HyPR device is thus a small configuration space that provides clinicians with quick and easy access to patient data by physical connection between the mobile device and augmented paper record. This mechanism allows clinicians to very quickly reconfigure their mobile device to the activity configuration of the rigid surrogated artifact, the paper record. This dramatically reduces configuration work during, e.g., ward rounds or emergency situations in which manually finding the activity configuration of the patient would be too time consuming or inappropriate. To carry both the augmented paper record and mobile device comfortably, microsuction tape is attached to the paper record to keep the tablet or phone attached to the record. Once the mobile device is paired to the record it can be moved and used separately without losing the connection. Devices can also be remotely connected to the HyPR device by simply looking up the patient case in the application. Once the devices are paired, the clinician can modify the color scheme of the led matrix that is built into the HyPR device (as seen in Figure 5.22), to signal status information. The colored lights can be used as a physical activity signifier that externalizes the workflow through physical changes of the surrogate. This information can be a patient status (e.g., Early Warning Score) or the color can be associated to the nurse or doctor who is responsible for the patient case. The lights can also be set to 'blink' in different patterns to signal more complex workflows such as the arrival of new blood tests. Clinicians can also turn on a sound signal on the device, which can be used to help locate records, which may be scattered over the ward or located in a cupboard. The HyPR device is also equipped with an ultrasound location tag that can be used to look up where the paper record is located.

The HyPR device operates within a large infrastructure, called Smartward, which is based on NooSphere and provides activity-aware patient management and information access designed to support multi-device location-aware collaborative workflows in patient wards. The infrastructure supports (i) large interactive screens for shared collaborative workspaces, (ii) tablet applications for mobile personalized activity configurations in the form of detailed patient information, and (iii) desktop systems for activity-centric integration with existing applications and services. The HyPR application running on the tablet is a web-based stripped down electronic patient record that consists of a patient overview screen (Figure 5.23A) and a detailed patient record (Figure 5.23B). In the overview screen, all patients that are currently at the ward are listed with basic information including their name, medical procedure, assigned color and room number. Using this patient overview, clinicians can set the colored lights of a specific patient record to 'blinking', thus asking for attention. Clinicians can also turn on the buzzing sound (which automatically stops after 15 seconds) of the record to quickly locate it when it is in a drawer or on a stack of other records.

The tablet is synchronized with the paper record through physical proximity. Placing the tablet on top of the paper record, automatically opens the detailed patient information of that patient to the tablet (Figure 5.23B). This view lists all detailed medical information and allows clinicians to add new medical data or messages. It can also be used to change the colored representation of the patient state. Changing this color in the details view updates the color on the HyPR

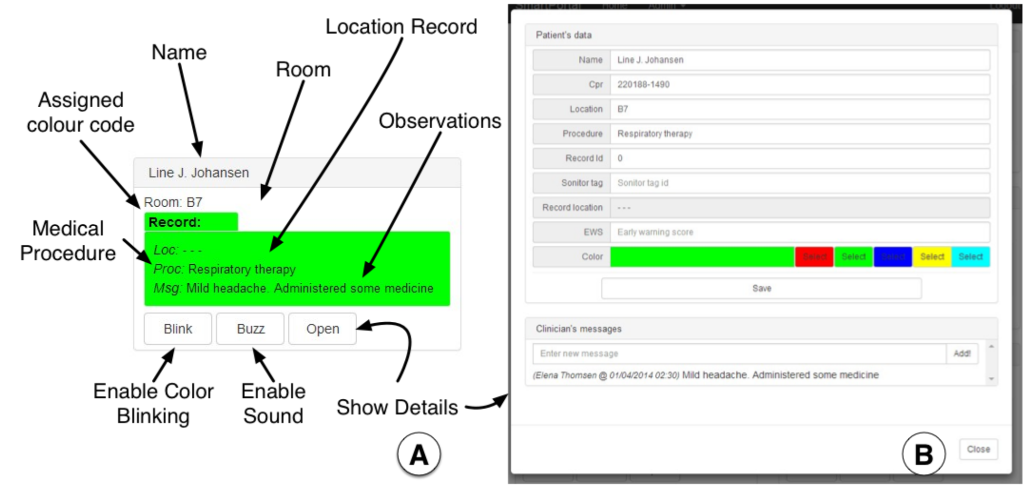


Figure 5.23: The details of the patient record.

device. When medical information is added remotely, through another tablet or computing device that is not physically paired to the paper record, the device's colored lights start blinking to signify an update. Once a clinician pairs the tablet, the new data is shown and the record stops blinking. All patients were modeled as shared activity configurations in which all information, collaborative flows and interaction possibilities with the HyPR device are centralized. The color and name of patients are used as activity signifiers that are synchronized with the color of the HyPR device.

5.6.2 Clinical Simulation

Because doing a field study with this type of pervasive technology in a real hospital is not safe and feasible, we conducted a clinical simulation in a separate 1:1 simulation environment. In the last decades, simulations have been increasingly used to train clinical personnel in new procedures, surgery or medication prescription. Recently, this approach has also been used to test, verify and evaluate systems and software that are used in a clinical environment [89]. The approach allows for the deployment of experimental software and hardware with representative users doing representative tasks, in an ecologically valid setting. During the two-day simulation, 8 senior clinicians (5 female, 3 male, mean age = 46) with different specialties (such as surgery, psychiatry and intensive care) participated in the experiment. Participants included 5 doctors, 2 nurses and a psychologist. The entire simulation was recorded using video and audio as well as extensive note taking and observations from inside the observation room through a one-way mirror. This simulation facility supports the simulation of different hospital departments ranging from patient wards to surgical departments and emergency departments. For our study, we set up the facility to be identical to a fully equipped patient ward with two patient bed rooms. Figure 5.24 shows the layout of the setup consisting of five zones: two patient rooms, a nurse station, a coffee room and the hallway. One human actor performed as a patient in a bed in room 2 (Figure 5.24, green dot). The other patient beds were equipped with simulation dolls, each connected to a monitor displaying the vital signs of the 'patient' (such as heart rate, saturation, blood pressure, temperature, etc). The setup included artifacts such as a traditional whiteboard with patients' data, desks in the nursing station with a stationary computer and nursing carts with medical equipment.

The study applied a scenario-based evaluation with scenarios that were drawn directly from the empirical field study (discussed in depth in paper **C7**). The scenarios revolved around interacting with the patients (both the actor and simulation dolls) to assess the patient case, update the status in the EHR and add or remove all necessary documents to the PMR. Scenarios included:

- S1 *Ward Round* – Clinicians were asked to perform a ward round to assess the situation of four patients. By examining the patients and monitoring vitals signs on the monitor, they had to calculate an Early Warning Score (EWS) to describe their current status.
- S2 *Blood Result* – Clinicians were asked to order a blood test result while working on the case of patient P1. After receiving the results they had to visit the patient, re-calculate the EWS and discuss the situation with the patient.
- S3 *Lost Record* – Clinicians were asked to find a number of PMRs, which, after a shift change, were not at their usual place. For this scenario, they could employ information on the patient's location, the last treating doctor, the current treatment procedure and the location of the HyPR.

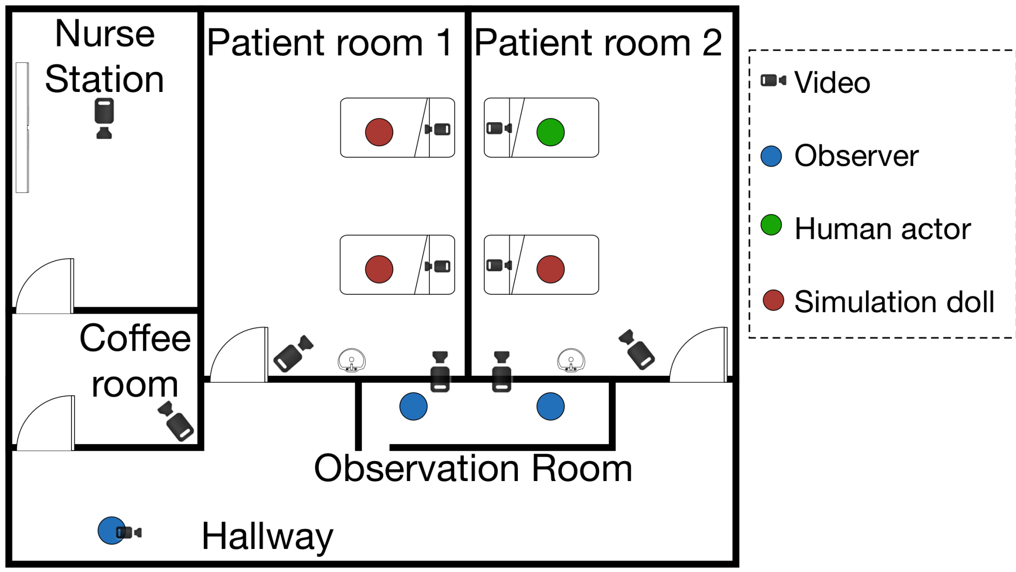


Figure 5.24: The simulation setup consisted of a medical ward with five zones including two patient rooms, a nurse station, a coffee room and a hallway. The simulation facility is equipped with hidden cameras and an observation room behind a one-way mirror. The simulation included three simulation dolls (patients) and one human acting as a real patient.

We did not provide any detailed instructions on how to perform the scenarios, which patients to look at first or how to use the system. Because we were interested in how clinicians would leverage their existing practices while using the HyPR setup, the scenarios were deliberately open ended: no explicit instructions or training on the system was given to them and the facilitator only intervened to solve technical issues. Because the initial field study showed that most medical work is highly collaborative, involving both doctors and nurses, the scenarios were conducted in pairs of two clinicians from the same department. Table 5.2 highlights the results of the 5-point Likert scale survey that ranged from "not at all useful" (1) to "very useful" (5).

Summarized, our study showed two main results. First, as described in paper **C6**, clinicians were able to very quickly use the HyPR as part of typical clinical work. Throughout the scenarios, we observed very similar use patterns among clinicians. During the ward round, most clinicians would almost immediately pair the mobile device with the HyPR device, before checking the vital signs

HyPR Usefulness (N=8)	*Min*	*Q1*	$\tilde{x}$	*Q3*	*Max*	*Iqr*
In general, the HyPR is useful	3	4	**4,5**	5	5	1
Pairing the PR and ER is useful	3	4	**4**	5	5	1
Color feature is useful	3	3,75	**4**	4,25	5	0,5
Tracking the PR is useful	4	4	**5**	5	5	1

Table 5.2: The results of the 5-point Likert scale questionnaire on the usefulness of the basic functions of a HyPR device. The table shows the minimum, maximum, median ($\tilde{x}$) and the interquartile range (iqr) of the scores. PR: paper record; ER: electronic record.

of the patients on the monitor or reading the content of paper forms. This configuration or artifact alignment process was thus the starting point for the interaction with the patient. During the patient assessment, clinicians would detach the mobile device from the HyPR and distribute it among both clinicians. One clinician would typically hold the tablet to enter information about the assessment while the other clinician browsed through the paper documentation to provide an overview on the patient case. Although the feedback on the design of the device was very positive, clinicians generally argued that the device was too heavy and thick and should be reduced to flexible paper-like constructions. Clinicians quickly appropriated the colored lights to structure their ward round or communicate workflows. During the scenarios, clinicians would organize the ward round based on the assigned color of each HyPR device. The colored lights were used as a shared workflow planning mechanism used to prioritized patients and externalize work process. However, the colored lights also opened a discussion related to effects on patients. Some clinicians argued that although the colored lights were a very useful workflow mechanism for clinicians, the effect on patients could be quite significant, as they might not understand why the color is changing,

Second, as detailed in paper **C7**, when comparing the interactions and workflows done with the PMR to those of the HyPR, we observed a range of existing collaborative affordances that were translated from the PMR to the HyPR. These collaborative affordances include *mobility and portability*, *collocated access*, *shared overview* and *mutual awareness*. We observed that the support for the mobility and portability in HyPR is very close to that of the PMR. The HyPR essentially acts as a portable place [134] as it can move across space and time but retain the indexical structure which points out relevant participants, places and times. Using the HyPR provides clinicians also with a high degree of plasticity to import a digital device into standard operation configurations, e.g., at the bedside of the patient, thus providing collocated access. The HyPR supports ad hoc and easy configuration of a distributed overview, as we observed how clinicians would break open the record during bedside patient interaction. Furthermore, when placing the HyPR device in the ward, it would be positioned in such a way that the colored lights where clearly visible during ward rounds or in the nurse station. The HyPR device was used to signal important information between clinicians, thus providing mutual awareness.

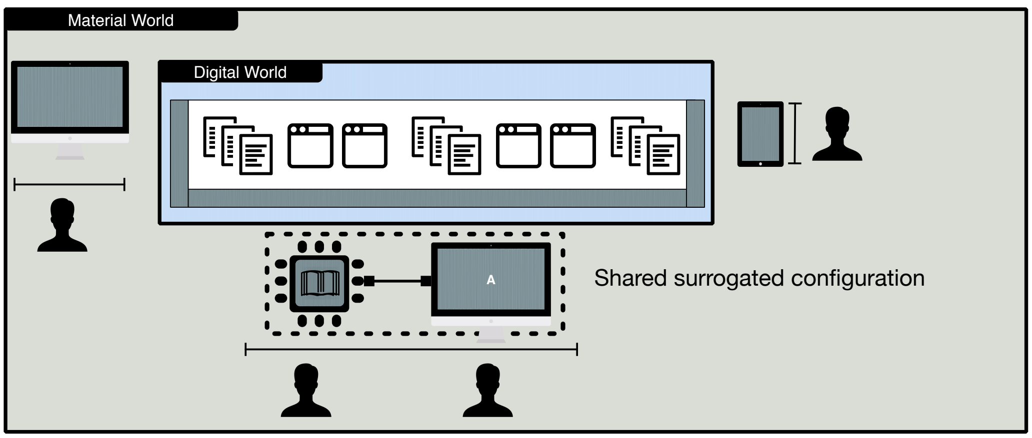


Figure 5.25: The HyPR supports surrogation of paper patient records to include it into shared activity configurations.

5.6.3 Contributions

Figure 5.1 and 5.25 situate the HyPR device into the activity configuration framework as an exploration into surrogates that augment non-digital artifacts to connect them to activity configurations. The central contribution of this work is twofold. First, we introduced the HyPR device and underlying infrastructure as a mechanism to reduce configuration work while using both paper and electronic records in nomadic clinical work. The HyPR approach supports flexible and dynamic configuration of paper and digital information, which allows for a gradual transition between a paper and digital workflow. Second, we conducted a field study to capture how the dual record is currently being used in hospitals and used those observations to conduct a medical simulation in which the system was deployed and tested in an ecologically valid environment. Based on these two studies we introduced the concept of collaborative affordances, which denotes a set of properties of physical devices and artifacts that support collaboration. These collaborative affordances include: mobility and portability, collocated access, shared overview and mutual awareness. The concept of collaborative affordances can be used in the analysis and design of collaborative technologies.

5.7 Summary

This chapter presented an overview of the tools, infrastructure and systems that operationalize the activity configuration framework. Through these systems, I demonstrated how technology can be designed to support human activity in rich device ecologies in real world scenarios and domains. As described in Section 5.1, each of these systems operationalize one cell in the activity configuration matrix and demonstrate how the framework can be applied to mitigate the configuration work problems discussed in Chapter 3. Section 5.2 first presented an infrastructure, NooSphere, that is designed to allow for the prototyping, design and deployment of systems and technology that apply activity configurations. I then presented co-activity Manager (Section 5.3), ActivitySpace (Section 5.4), WatchConnect (Section 5.5) and Hybrid Patient Record (Section 5.6) as examples of how users interact with activity configurations in both complex and real office- and hospital work settings.

6. Conclusions

This chapter concludes Part I of this dissertation. In this dissertation, I explored how activity configurations can be leveraged to reduce configuration work in distributed interaction. Using a triangulation approach that takes a theoretical, empirical and design perspective, this dissertation approached multi-device configuration work in both clinical and office work. Chapter 3 took an empirical perspective and analyzed configuration problems, that we observed during field studies, related to resource, device and activity management. Based on this analysis, I introduced a definition and description of configuration work. From a theoretical perspective, Chapter 4 explored how insights from tool mediation in Activity Theory could be used to conceptually extend Activity-Centric Computing with a device layer, opening up a design space for cross-device and artifacts interactions and systems. In Chapter 5, this design space was translated into an infrastructure and four systems that attempt to solve various real configuration work problems in both office and clinical work. These systems were developed, evaluated and deployed with real users through an iterative design process. This dissertation demonstrated how activity configurations can be used as a concept to reduce configuration work when interacting within complex workflows that are distributed over multiple devices and users.

6.1 Addressing the Research Questions

The overall research question that was addressed in this dissertation is: "*how can activity-centric computing be used to reduce configuration work in distributed interaction?*". Summarized, this question can be subdivided into three core research questions:

Q1 Configuration Work - How do users *set up*, *manage*, *communicate*, *understand* and *use* information, applications and services that are distributed over all devices in use and people involved? What constitutes configuration work? What are the properties and processes of configuration work?

To address **Q1**, Chapter 3 presented an overview of empirical work from the co-Activity Manager [**C1**], ActivitySpace [**C3**, **C4**] and Hybrid Patient Record [**C6**, **C7**] papers to provide an analysis of the problems and challenges in distributed setups. The chapter concluded that employing multiple devices to perform a particular task requires users to put a significant effort in what I call *configuration work*, the effort required to *set up*, *manage*, *communicate*, *understand* and *use* information, applications and services, which are distributed across several devices and people. Configuration work in distributed interaction is composed of several subprocesses that can occur in any order, duration or frequency. These subprocesses are curation work, task resumption lag, mobility work, physical handling of devices and articulation work. A number of multi-device use patterns can be derived that crystallize how users employ multiple devices. The patterns exemplify where configuration work is induced during parallel use of devices, alignment of information across devices, fragmentation of data between devices and replication or hand off of information between different devices.

Q2 **Activity Configuration** - How can human activities be captured in a digital system configuration? How is such a system activity configuration defined? What are the properties and benefits of using activity configuration as an approach to reduce configuration work? How is activity configuration work performed? How are configurations constructed and shared between users and devices?

As an approach to reduce and mitigate configuration work in distributed interaction, Chapter 4 addresses **Q2** by proposing to use Activity-Centric Computing. By emphasizing the tool mediation aspect of an activity theoretical analysis of human-machine interaction, the original concept of Activity-Centric Computing was extended with an explicit device and artifact layer, leading to the *activity configuration* concept. To make a clear delineation between real human activities and their computational representation, I defined activity configuration as "*a description of an interaction and social context (including files, devices, applications, material artifacts, other meta information, and coordination and communication tools) that is a reflection of the real ongoing activity*". Activity configurations are computational representations of real activities and are used to reduce the amount of configuration work when working in a distributed interactive setup, as they can be shared, distributed, fragmented and used on multiple devices that are used by multiple users. By connecting the real activity of users explicitly to their devices, users are presented with a cross-device representation that moves away from the predominant application- and document-centric paradigm. Activity configurations can be used on three types of devices: (i) personal computing devices, that are used by people for direct interaction with information, (ii) surrogates, that augment a non-digital artifact with sensors to connect it to digital information, and, (iii) configuration spaces, which are active spaces that visualize and mediate the management of information across device ecologies.

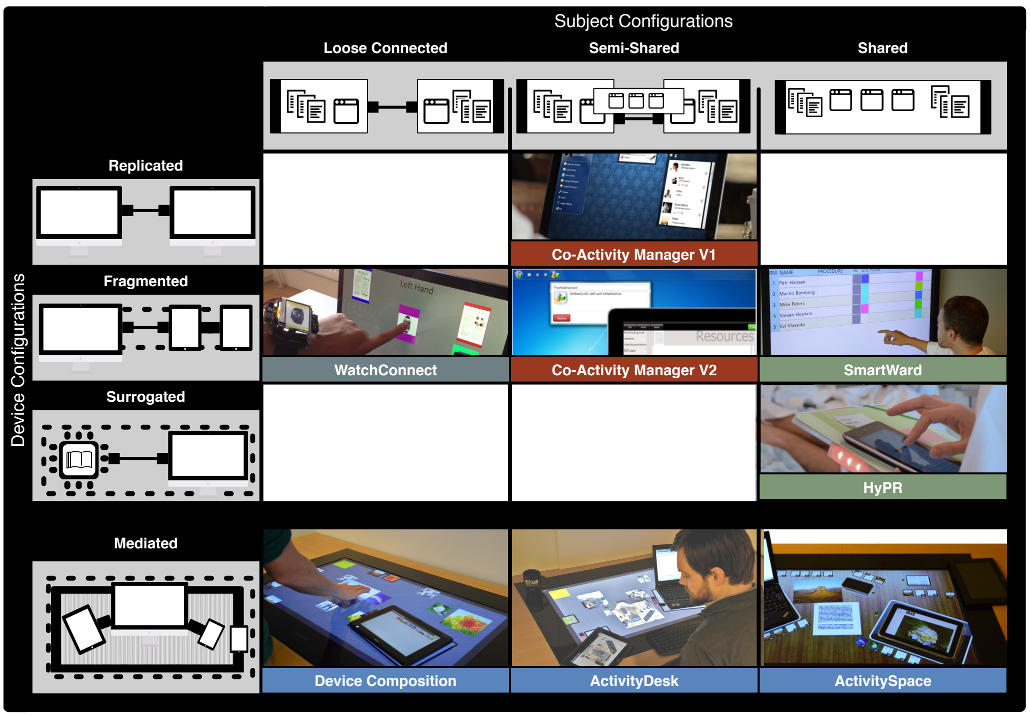


Figure 6.1: An overview of all the systems introduced in this thesis.

Using activity configuration as a basic unit of analysis for distributed interaction, I conceptualize how activity configurations are shared across users and devices. This resulted in the configuration matrix (Figure 6.1) that provides us with an analytical framework for the design and implementation of multi-device and multi-user activity-centric systems. The framework considers three types of user setups: (i) loose connected configuration, in which separate activity configurations are used that are connected through communication and coordination tools,

(ii) semi-shared configurations, in which a subset of the activity configuration is shared through an overlapping resource container, and, (iii) shared configurations, in which the entire activity configuration is synchronized and distributed across users. Grounded in the patterns identified in Chapter 3, the three types of user configurations can be setup and used in four types of device configurations: (i) replicated configuration, in which the activity configuration is copied across devices, (ii) fragmented configuration, in which the components of one activity configuration are fragmented across multiple devices, (iii) surrogated configuration, in which a non-digital artifact is augmented with sensors to connect it to a personal device, and, (iv) a special configuration space setup, in which the prior three device setups can be visualized and mediated inside an active space.

Q3 **Technology** - What are the technical and user requirements for technology that supports activity configurations? How can activity-centric technology be used to reduce configuration work in distributed interaction?

To address **Q3**, Chapter 5 provides an overview of the systems and technology described in the 7 papers of Part II that aim to solve real world configuration problems in distributed scenarios. To address the infrastructure problems in distributed interaction, the chapter first introduced NooSphere [**C2**], an infrastructure to prototype, build and deploy distributed activity-centric systems and applications. Using a two-layer architecture and a computational representation of activity configuration, the infrastructure implements and supports seven core distributed computing services including data persistence and distribution, discovery and pairing of devices, support for integrated coordination and communication, application and activity configuration management, context handling and interoperability. NooSphere abstracts these services into distributed activity nodes that can dynamically be created and used in various architectural patterns to allow for a wide range of different device configurations.

As summarized in Figure 6.1, we constructed four systems that explored different cells in the configuration matrix. First, co-Activity Manager [**C1**, **C2**] was designed to explore multi-device and multi-user activity configurations that reduce the overhead of manually sharing and organizing team work. The deployment of this system showed that users appropriated the activity configuration mechanism to support a wide range of different work practices, and that configurations are used in complex distributed activity lifecycles. Second, building on the idea of supporting fluid cross-device information spaces that allowed for the easy configuration of devices, ActivityDesk [C3] and ActivitySpace [C4] were designed as examples of configuration spaces that mediate and facilitate activity-centric configuration work across devices. Our lab deployment showed that, similar to the co-Activity Manager [C1] study, users appropriated the cross-device configuration mechanism in several ways, but also that users employ a number of different strategies to spatially organize their devices and resources. Third, to explore in more depth how activity configurations can be fragmented across devices using our hands as reconfigurable instruments, WatchConnect [C5] introduced a toolkit and set of interaction techniques and applications that explored how the input and output space of smartwatches can be leveraged to allow for fluid cross-device activity configurations. Using a watch probe, we demonstrated, through 7 interaction techniques and applications, how the characteristics of a watch can be used to easily move information or input and output between devices. Finally, to explore how non-digital artifacts can be included into complex distributed workflows, the Hybrid Patient Record (HyPR) device [C6, C7] augments paper patient records with (i) near-field communication, to allow for the configuration of paper and digital patient data, (ii) a notification system, that equips the record with reconfigurable colors and sound that externalizes distributed workflows, and, (iii) an ultrasound location tracker, that allows clinicians to find the record in the hospital. Our clinical simulation demonstrated that the HyPR is a stable concept that supports existing work practices of clinicians but also dramatically reduced configuration work.

6.2 Future Work

Although this dissertation demonstrates how activity configurations can be used as a flexible and generic approach to build systems that reduce configuration work in distributed interactions, this work spawns a number of new research questions and topics that require further exploration.

6.2.1 Distributed Activity Configuration Lifecycle

A fundamental open question is how the entire multi-device and multi-user activity lifecycle can be supported and visualized. Although we have done early attempts at understanding and visualizing this lifecycle, the Activity-Centric Computing paradigm currently lacks conceptual tools to describe the complex interdependencies and lifecycles of activity configurations. It is currently unclear which processes are part of distributed activity configuration lifecycles, and how users would leverage these processes to construct temporally stable information structures that support distributed work over long periods of times. A second question about the lifecycle is on the tension and differences between highly formal workflows, such as those in hospitals, and more loose workflows, such as office work. These tensions are currently not embedded into the concept of an activity configuration but left in the hands of the designer.

6.2.2 Spatial Mediation

The configuration space concept explored the use of a physical action space to mediate devices and visualize and externalize activity configurations across connected devices. Device configurations are not static since device roles change during interaction with the world. Moreover, some devices have the expressive power to mediate interaction between other devices from the same or different users. Future work could explore how this class of mediating devices (such as interactive surfaces in stationary situations, or projection-based interfaces in mobile scenarios) can reduce intelligibility problems and support users in interacting with complex distributed device ecologies. The configuration space concept could be extended to a configuration room, in which the user is tracked using proxemics [8], that allows for larger scale and fine grained control over spatially organized devices and their containing resources. Furthermore, the social impact of this type of implicit body-centric configuration work could be explored to create a better understanding of how users would utilize these spaces.

6.2.3 Physical Configuration

With the increasing amount of personal devices, the role of those devices is becoming more important. Extending on our WatchConnect toolkit, future work could explore how wearable devices, such as head-mounted displays or smartwatches, can be leveraged as mediating devices that transform the human body into instruments that can be used to configure other devices, artifacts and other objects in the world. This approach opens up an interesting design space to explore how interactions with activity configurations can be done in a more physical way. Users could employ their personal wearable device to enable interaction with any public device or surface by simply looking at a screen or grabbing the device. These *look and use* or *grab and use* approaches can disconnect users from personal devices, and place a focus on *one* mediating device that can enable and control other public devices in range.

Bibliography

[1] Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R., and Pinkerton, M. Cyberguide: A mobile context-aware tour guide. *Wireless networks 3*, 5 (1997), 421–433.

[2] Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., and Steggles, P. Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing* (1999), Springer, pp. 304–307.

[3] Abowd, G. D., and Mynatt, E. D. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction (TOCHI) 7*, 1 (2000), 29–58.

[4] Arthur, R., and Olsen Jr, D. R. Xice windowing toolkit: Seamless display annexation. *ACM Transactions on Computer-Human Interaction (TOCHI) 18*, 3 (2011), 14.

[5] Arthur, R., and Olsen Jr, D. R. Window brokers: Collaborative display space control. *ACM Transactions on Computer-Human Interaction (TOCHI) 19*, 3 (2012), 17.

[6] Bærentsen, K. B., and Trettvik, J. An activity theory approach to affordance. In *Proceedings of the second Nordic conference on Human-computer interaction* (2002), ACM, pp. 51–60.

[7] Balakrishnan, A. D., Matthews, T., and Moran, T. P. Fitting an activity-centric system into an ecology of workplace tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2010), ACM, pp. 787–790.

[8] Ballendat, T., Marquardt, N., and Greenberg, S. Proxemic interaction: designing for a proximity and orientation-aware environment. In *ACM International Conference on Interactive Tabletops and Surfaces* (2010), ACM, pp. 121–130.

[9] Bannon, L., and Bødker, S. Constructing common information spaces. In *Proceedings of the Fifth European Conference on Computer Supported Cooperative Work* (1997), Springer, pp. 81–96.

[10] Bannon, L., Cypher, A., Greenspan, S., and Monty, M. L. Evaluation and analysis of users' activity organization. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (1983), ACM, pp. 54–57.

[11] Bannon, L. J., and Bødker, S. Beyond the interface: Encountering artifacts in use. *DAIMI Report Series 18*, 288 (1989).

[12] Bardram, J. Designing for the dynamics of cooperative work activities. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work* (1998), ACM, pp. 89–98.

[13] Bardram, J., Bunde-Pedersen, J., and Soegaard, M. Support for activity-based computing in a personal computing operating system. In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (2006), ACM, pp. 211–220.

[14] Bardram, J., and Doryab, A. Activity analysis: applying activity theory to analyze complex work in hospitals. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work* (2011), ACM, pp. 455–464.

[15] Bardram, J., Gueddana, S., Houben, S., and Nielsen, S. Reticularspaces: activity-based computing support for physically distributed and collaborative smart spaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), ACM, pp. 2845–2854.

[16] Bardram, J. E. Plans as situated action: an activity theory approach to workflow systems. In *Proceedings of the Fifth European Conference on Computer Supported Cooperative Work* (1997), Springer, pp. 17–32.

[17] Bardram, J. E. Temporal coordination–on time and coordination of collaborativeactivities at a surgical department. *Computer Supported Cooperative Work (CSCW) 9*, 2 (2000), 157–187.

[18] Bardram, J. E. Activity-based computing for medical work in hospitals. *ACM Transactions on Computer-Human Interaction (TOCHI) 16*, 2 (2009), 10.

[19] Bardram, J. E., and Bossen, C. Mobility work: The spatial dimension of collaboration at a hospital. *Computer Supported Cooperative Work (CSCW) 14*, 2 (2005), 131–160.

[20] Bardram, J. E., and Bossen, C. A web of coordinative artifacts: collaborative work at a hospital ward. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work* (2005), ACM, pp. 168–176.

[21] Bardram, J. E., Bunde-Pedersen, J., Doryab, A., and Sørensen, S. Clinical surfaces–activity-based computing for distributed multi-display environments in hospitals. In *Human-Computer Interaction–INTERACT 2009*. Springer, 2009, pp. 704–717.

[22] Bardram, J. E., and Hansen, T. R. Peri-operative coordination and communication systems: A case of cscw in medical informatics. In *Proceedings of the CSCW 2010 workshop on CSCW Research in Healthcare: Past, Present and Future* (2010).

[23] Bardram, J. E., Hansen, T. R., and Soegaard, M. Awaremedia: a shared interactive display supporting social, temporal, and spatial awareness in surgery. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work* (2006), ACM, pp. 109–118.

[24] Beaudouin-Lafon, M. Instrumental interaction: an interaction model for designing post-wimp user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2000), ACM, pp. 446–453.

[25] Beaudouin-Lafon, M., Huot, S., Nancel, M., Mackay, W., Pietriga, E., Primet, R., Wagner, J., Chapuis, O., Pillias, C., Eagan, J. R., et al. Multisurface interaction in the wild room. *Computer 45*, 4 (2012), 48–56.

[26] Bellotti, V., Ducheneaut, N., Howard, M., and Smith, I. Taskmaster: recasting email as task management. *PARC, CSCW 2* (2002).

[27] Berg, M. Accumulating and coordinating: occasions for information technologies in medical work. *Computer Supported Cooperative Work (CSCW) 8*, 4 (1999), 373–401.

[28] Bergman, O., Beyth-Marom, R., and Nachmias, R. The project fragmentation problem in personal information management. In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (2006), ACM, pp. 271–274.

[29] Bernstein, A. How can cooperative work tools support dynamic group process? bridging the specificity frontier. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work* (2000), ACM, pp. 279–288.

[30] Biehl, J. T., and Bailey, B. P. Aris: an interface for application relocation in an interactive space. In *Proceedings of the 2004 Graphics Interface Conference* (2004), Canadian Human-Computer Communications Society, pp. 107–116.

[31] Biehl, J. T., Baker, W. T., Bailey, B. P., Tan, D. S., Inkpen, K. M., and Czerwinski, M. Impromptu: a new interaction framework for supporting collaboration in multiple display environments and its field evaluation for co-located software development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2008), ACM, pp. 939–948.

[32] Bjørn, P., and Hertzum, M. Artefactual multiplicity: A study of emergency-department whiteboards. *Computer Supported Cooperative Work (CSCW) 20*, 1-2 (2011), 93–121.

[33] Bødker, S. Through the interface-a human activity approach to user interface design. *DAIMI Report Series 16*, 224 (1987).

[34] Bødker, S. A human activity approach to user interfaces. *Human-Computer Interaction 4*, 3 (1989), 171–195.

[35] Bødker, S. When second wave hci meets third wave challenges. In *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles* (2006), ACM, pp. 1–8.

[36] Bødker, S., and Klokmose, C. N. The human–artifact model: An activity theoretical approach to artifact ecologies. *Human–Computer Interaction 26*, 4 (2011), 315–371.

[37] Bødker, S., and Klokmose, C. N. Dynamics in artifact ecologies. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design* (2012), ACM, pp. 448–457.

[38] Bragdon, A., DeLine, R., Hinckley, K., and Morris, M. R. Code space: touch+ air gesture hybrid interactions for supporting developer meetings. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (2011), ACM, pp. 212–221.

[39] Bunde-Pedersen, J. B. *Distributed Interaction for Activity-Based computing*. PhD thesis, Aarhus Universitetsforlag, 2009.

[40] Card, S. K., Newell, A., and Moran, T. P. The psychology of human-computer interaction.

[41] Carroll, J. M. *HCI models, theories, and frameworks: Toward a multidisciplinary science*. Morgan Kaufmann, 2003.

[42] Chang, T.-H., and Li, Y. Deep shot: a framework for migrating tasks across devices using mobile phone cameras. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2011), ACM, pp. 2163–2172.

[43] Chantler, C., Clarke, T., and Granger, R. Information technology in the english national health service. *JAMA 296*, 18 (2006), 2255–2258.

[44] Chen, G., Kotz, D., et al. A survey of context-aware mobile computing research. Tech. rep., Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.

[45] Chen, N., Guimbretiere, F., and Sellen, A. Designing a multi-slate reading environment to support active reading activities. *ACM Transactions on Computer-Human Interaction (TOCHI) 19*, 3 (2012), 18.

[46] Chen, X., Grossman, T., Wigdor, D. J., and Fitzmaurice, G. Duet: exploring joint interactions on a smart phone and a smart watch. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems* (2014), ACM, pp. 159–168.

[47] Chen, Y. Documenting transitional information in emr. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2010), ACM, pp. 1787–1796.

[48] Christensen, H. B., and Bardram, J. E. Supporting human activities - exploring activity-centered computing. In *UbiComp 2002: Ubiquitous Computing*. Springer, 2002, pp. 107–116.

[49] Coiera, E. Clinical communication: a new informatics paradigm. In *Proceedings of the AMIA Annual Fall Symposium* (1996), American Medical Informatics Association, p. 17.

[50] Czerwinski, M., Horvitz, E., and Wilhite, S. A diary study of task switching and interruptions. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2004), ACM, pp. 175–182.

[51] Dearman, D., and Pierce, J. S. It's on my other computer!: computing with multiple devices. In *Proceedings of the SIGCHI Conference on Human factors in Computing Systems* (2008), ACM, pp. 767–776.

[52] Dourish, P. *Where the action is: the foundations of embodied interaction*. MIT press, 2004.

[53] Dourish, P., and Button, G. On" technomethodology": Foundational relationships between ethnomethodology and system design. *Human-computer interaction 13*, 4 (1998), 395–432.

[54] Dragunov, A. N., Dietterich, T. G., Johnsrude, K., McLaughlin, M., Li, L., and Herlocker, J. L. Tasktracer: a desktop environment to support multi-tasking knowledge workers. In *Proceedings of the 10th international conference on Intelligent user interfaces* (2005), ACM, pp. 75–82.

[55] Ducheneaut, N., and Bellotti, V. E-mail as habitat: an exploration of embedded personal information management. *interactions 8*, 5 (2001), 30–38.

[56] Edwards, W. K., Bellotti, V., Dey, A. K., and Newman, M. W. The challenges of user-centered design and evaluation for infrastructure. In

Proceedings of the SIGCHI conference on Human factors in computing systems (2003), ACM, pp. 297–304.

[57] Edwards, W. K., Newman, M. W., and Poole, E. S. The infrastructure problem in hci. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2010), ACM, pp. 423–432.

[58] Edwards, W. K., Newman, M. W., Sedivy, J., Smith, T., and Izadi, S. Challenge: recombinant computing and the speakeasy approach. In *Proceedings of the 8th annual international conference on Mobile computing and networking* (2002), ACM, pp. 279–286.

[59] Edwards, W. K., Newman, M. W., and Sedivy, J. Z. The case for recombinant computing. *Xerox Palo Alto Research Center Technical Report CSL-01-1* (2001).

[60] Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F., Janssen, W., Lee, D., McCall, K., Pedersen, E., Pier, K., et al. Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1992), ACM, pp. 599–607.

[61] Engeström, Y. *Learning by expanding: An activity-theoretical approach to developmental research*. Orienta-Konsultit Oy., 1987.

[62] Engeström, Y. Enriching activity theory without shortcuts. *Interacting with Computers 20*, 2 (2008), 256–259.

[63] Engeström, Y., et al. Activity theory and individual and social transformation. *Perspectives on activity theory* (1999), 19–38.

[64] Fisher, D., Brush, A., Gleave, E., and Smith, M. A. Revisiting whittaker & sidner's email overload ten years later. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work* (2006), ACM, pp. 309–312.

[65] Fitzpatrick, G. Integrated care and the working record. *Health Informatics Journal 10*, 4 (2004), 291–302.

[66] Fitzpatrick, G., and Ellingsen, G. A review of 25 years of cscw research in healthcare: contributions, challenges and future agendas. *Computer Supported Cooperative Work (CSCW) 22*, 4-6 (2013), 609–665.

[67] Fitzpatrick, G., and Ellingsen, G. A review of 25 years of cscw research in healthcare: contributions, challenges and future agendas. *Computer Supported Cooperative Work (CSCW) 22*, 4-6 (2013), 609–665.

[68] for Computing Machinery (ACM), A. Definition of human-computer interaction, Nov. 2014.

[69] Garlan, D., Siewiorek, D. P., Smailagic, A., and Steenkiste, P. Project aura: Toward distraction-free pervasive computing. *Pervasive Computing, IEEE 1*, 2 (2002), 22–31.

[70] Geyer, W., Muller, M. J., Moore, M. T., Wilcox, E., Cheng, L.-T., Brownholtz, B., Hill, C., and Millen, D. R. Activity explorer: activity-centric collaboration from research to product. *IBM Systems Journal 45*, 4 (2006), 713–738.

[71] Gibson, J. J. The theory of affordances. *Hilldale, USA* (1977).

[72] Gibson, J. J. *The ecological approach to visual perception*. Psychology Press, 2013.

[73] Gjerlufsen, T., Klokmose, C. N., Eagan, J., Pillias, C., and Beaudouin-Lafon, M. Shared substance: developing flexible multi-surface applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2011), ACM, pp. 3383–3392.

[74] González, V. M., and Mark, G. Constant, constant, multi-tasking craziness: managing multiple working spheres. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2004), ACM, pp. 113–120.

[75] González, V. M., Tentori, M. E., Morán, E. B., Favela, J., and Martínez, A. I. Understanding mobile work in a distributed information space: implications for the design of ubicomp technology. In *Proceedings of the 2005 Latin American conference on Human-computer interaction* (2005), ACM, pp. 52–63.

[76] Greenberg, S., Marquardt, N., Ballendat, T., Diaz-Marino, R., and Wang, M. Proxemic interactions: the new ubicomp? *interactions 18*, 1 (2011), 42–50.

[77] Grudin, J. Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2001), ACM, pp. 458–465.

[78] Hall, E. T., and Hall, E. T. *The hidden dimension*, vol. 1990. Anchor Books New York, 1969.

[79] Hardy, R., and Rukzio, E. Touch & interact: touch-based interaction of mobile phones with displays. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services* (2008), ACM, pp. 245–254.

[80] Harrison, S., Tatar, D., and Sengers, P. The three paradigms of hci. In *Alt. Chi. Session at the SIGCHI Conference on Human Factors in Computing Systems San Jose, California, USA* (2007), pp. 1–18.

[81] Heerwagen, J. H., Kampschroer, K., Powell, K. M., and Loftness, V. Collaborative knowledge work environments. *Building research & information 32*, 6 (2004), 510–528.

[82] Henderson Jr, D. A., and Card, S. Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Transactions on Graphics (TOG) 5*, 3 (1986), 211–243.

[83] Hertzum, M. The distributed use of electronic emergency-department whiteboards. In *MIE2012: Proceedings of the XXIVth International Congress of the European Federation for Medical Informatics* (2012), pp. 26–29.

[84] Hinckley, K. Synchronous gestures for multiple persons and computers. In *Proceedings of the 16th annual ACM symposium on User interface software and technology* (2003), ACM, pp. 149–158.

[85] Houben, S., Bardram, J. E., Vermeulen, J., Luyten, K., and Coninx, K. Activity-centric support for ad hoc knowledge work: a case study of

co-activity manager. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2013), ACM, pp. 2263–2272.

[86] Hutchins, E. Distributed cognition. *Internacional Enciclopedia of the Social and Behavioral Sciences* (2000).

[87] Ishii, H., and Ullmer, B. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems* (1997), ACM, pp. 234–241.

[88] Izadi, S., Brignull, H., Rodden, T., Rogers, Y., and Underwood, M. Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media. In *Proceedings of the 16th annual ACM symposium on User interface software and technology* (2003), ACM, pp. 159–168.

[89] Jaspers, M. W. A comparison of usability methods for testing interactive health technologies: methodological aspects and empirical evidence. *International journal of medical informatics 78*, 5 (2009), 340–353.

[90] Johanson, B., Fox, A., and Winograd, T. The interactive workspaces project: Experiences with ubiquitous computing rooms [version# 2, 4/11/02]. *IEEE Pervasive Computing 1* (April 2002), 67–74.

[91] Johanson, B., Hutchins, G., Winograd, T., and Stone, M. Pointright: experience with flexible input redirection in interactive workspaces. In *Proceedings of the 15th annual ACM symposium on User interface software and technology* (2002), ACM, pp. 227–234.

[92] 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 *Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges* (2008), ACM, pp. 201–210.

[93] Kane, S. K., Karlson, A. K., Meyers, B. R., Johns, P., Jacobs, A., and Smith, G. Exploring cross-device web use on pcs and mobile devices. In *Human-Computer Interaction–INTERACT 2009*. Springer, 2009, pp. 722–735.

[94] Kaptelinin, V. *Computer-Mediated Activity: Functional Organs in Social and Developmental Contexts*. MIT Press, 1996, ch. 3, pp. 45–68.

[95] Kaptelinin, V. *Computer-mediated activity: Functional organs in social and developmental contexts*. 1996, pp. 45–68.

[96] Kaptelinin, V. Creating computer-based work environments: an empirical study of macintosh users. In *Proceedings of the 1996 ACM SIGCPR/SIGMIS conference on Computer personnel research* (1996), ACM, pp. 360–366.

[97] Kaptelinin, V. Umea: translating interaction histories into project contexts. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2003), ACM, pp. 353–360.

[98] Kaptelinin, V. *Acting with technology: Activity theory and interaction design*. Mit Press, 2006.

[99] Kaptelinin, V. Activity theory. *Soegaard, Mads and Dam, Rikke Friis (eds.). "The Encyclopedia of Human-Computer Interaction, 2nd Ed.". Aarhus, Denmark: The Interaction Design Foundation.* (2013).

[100] Kaptelinin, V., and Czerwinski, M. *Beyond the desktop metaphor: designing integrated digital work environments*, vol. 1. The MIT Press, 2007.

[101] Kaptelinin, V., and Nardi, B. Affordances in hci: toward a mediated action perspective. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), ACM, pp. 967–976.

[102] Kaptelinin, V., and Nardi, B. A. *Acting with Technology: Activity Theory and Interaction Design (Acting with Technology)*. The MIT Press, 2006.

[103] Karlson, A. K., Meyers, B. R., Jacobs, A., Johns, P., and Kane, S. K. Working overtime: Patterns of smartphone and pc usage in the day of an information worker. In *Pervasive computing*. Springer, 2009, pp. 398–405.

[104] Kawsar, F., Vermeulen, J., Smith, K., Luyten, K., and Kortuem, G. Exploring the design space for situated glyphs to support dynamic work environments. In *Pervasive Computing*. Springer, 2011, pp. 70–78.

[105] Kreijns, K., and Kirschner, P. A. The social affordances of computer-supported collaborative learning environments. In *Frontiers in Education Conference, 2001. 31st Annual* (2001), vol. 1, IEEE, pp. T1F–12.

[106] Kuutti, K. *Activity Theory as a Potential Framework for Human-Computer Interaction Research*. MIT Press, 1996, ch. 2, pp. 17–44.

[107] Lamming, M. G., and Newman, W. M. Activity-based information retrieval technology in support of personal memory. In *IFIP Congress (3)* (1992), vol. 14, Citeseer, pp. 68–81.

[108] Latour, B. Reassembling the social-an introduction to actor-network-theory. *Reassembling the Social-An Introduction to Actor-Network-Theory, by Bruno Latour, pp. 316. Foreword by Bruno Latour. Oxford University Press, Sep 2005 1* (2005).

[109] Lave, J., and Wenger, E. *Situated learning: Legitimate peripheral participation*. Cambridge university press, 1991.

[110] Leonhardt, U., Magee, J., and Dias, P. Location service in mobile computing environments. *Computers & Graphics 20*, 5 (1996), 627–632.

[111] Leontiev, A. N. Activity and consciousness. *Moscow: Personality* (1974).

[112] Leontiev, A. N. *Activity, Consciousness, and Personality*. Prentice-Hall, 1979.

[113] Li, Y., and Landay, J. A. Activity-based prototyping of ubicomp applications for long-lived, everyday human activities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2008), ACM, pp. 1303–1312.

[114] Luff, P., and Heath, C. Mobility in collaboration. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work* (1998), ACM, pp. 305–314.

[115] Lynch, M. *Scientific practice and ordinary action: Ethnomethodology and social studies of science*. Cambridge University Press, 1997.

[116] MacIntyre, B., Mynatt, E. D., Voida, S., Hansen, K. M., Tullio, J., and Corso, G. M. Support for multitasking and background awareness using interactive peripheral displays. In *Proceedings of the 14th annual ACM symposium on User interface software and technology* (2001), ACM, pp. 41–50.

[117] Mackay, W. E., and Fayard, A.-L. Hci, natural science and design: a framework for triangulation across disciplines. In *Proceedings of the 2nd conference on Designing interactive systems: processes, practices, methods, and techniques* (1997), ACM, pp. 223–234.

[118] Marquardt, N., Ballendat, T., Boring, S., Greenberg, S., and Hinckley, K. Gradual engagement: facilitating information exchange between digital devices as a function of proximity. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces* (2012), ACM, pp. 31–40.

[119] Marquardt, N., Hinckley, K., and Greenberg, S. Cross-device interaction via micro-mobility and f-formations. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (2012), ACM, pp. 13–22.

[120] Mayer, S., and Sörös, G. User interface beaming - seamless interaction with smart things using personal wearable computers. In *Proceedings of the 2014 International Conference on Wearable and Implantable Body Sensor Networks (BSN 2014)* (Zurich, Switzerland, June 2014).

[121] Millen, D. R., Muller, M. J., Geyer, W., Wilcox, E., and Brownholtz, B. Patterns of media use in an activity-centric collaborative environment. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2005), ACM, pp. 879–888.

[122] Morán, E. B., Tentori, M., Gonzalez, V. M., and Favela, J. Mobility in hospital work: towards a pervasive computing hospital environment. *International Journal of Electronic Healthcare 3*, 1 (2007), 72–89.

[123] Moran, T. P., and Zhai, S. *Beyond the desktop in seven dimensions*. The MIT Press, 2007, ch. 11, pp. 335–354.

[124] Morrison, C., Fitzpatrick, G., and Blackwell, A. Multi-disciplinary collaboration during ward rounds: Embodied aspects of electronic medical record usage. *International journal of medical informatics 80*, 8 (2011), e96–e111.

[125] Muller, M. J., Geyer, W., Brownholtz, B., Wilcox, E., and Millen, D. R. One-hundred days in an activity-centric collaboration environment based

on shared objects. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2004), ACM, pp. 375–382.

[126] Mynatt, E. D., Igarashi, T., Edwards, W. K., and LaMarca, A. Flatland: new dimensions in office whiteboards. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (1999), ACM, pp. 346–353.

[127] Nardi, B. A. Studying context: A comparison of activity theory, situated action models, and distributed cognition. *Context and consciousness: Activity theory and human-computer interaction* (1996), 69–102.

[128] Neisser, U. Cognitive psychology.

[129] Norman, D. A. Affordance, conventions, and design. *interactions 6*, 3 (1999), 38–43.

[130] Norman, D. A. *The invisible computer: why good products can fail, the personal computer is so complex, and information appliances are the solution*. The MIT press, 1999.

[131] Norman, D. A. The way i see it signifiers, not affordances. *interactions 15*, 6 (2008), 18–19.

[132] Odom, W., Sellen, A., Harper, R., and Thereska, E. Lost in translation: understanding the possession of digital things in the cloud. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), ACM, pp. 781–790.

[133] Oleksik, G., Wilson, M. L., Tashman, C., Mendes Rodrigues, E., Kazai, G., Smyth, G., Milic-Frayling, N., and Jones, R. Lightweight tagging expands information and activity management practices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2009), ACM, pp. 279–288.

[134] Østerlund, C. S. Documents in place: demarcating places for collaboration in healthcare settings. *Computer Supported Cooperative Work (CSCW) 17*, 2-3 (2008), 195–225.

[135] Oulasvirta, A. When users do the ubicomp. *interactions 15*, 2 (2008), 6–9.

[136] Oulasvirta, A., and Sumari, L. Mobile kits and laptop trays: managing multiple devices in mobile information work. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2007), ACM, pp. 1127–1136.

[137] Park, S. Y., and Chen, Y. Adaptation as design: Learning from an emr deployment study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), ACM, pp. 2097–2106.

[138] Parker, J., and Coiera, E. Improving clinical communication a view from psychology. *Journal of the American Medical Informatics Association 7*, 5 (2000), 453–461.

[139] Pederson, T., Janlert, L.-E., and Surie, D. A situative space model for mobile mixed-reality computing. *IEEE pervasive computing 10*, 4 (2011), 73–83.

[140] Perry, M., and Brodie, J. *Virtually connected, practically mobile*. Springer, 2006.

[141] Prekop, P., and Burnett, M. Activities, context and ubiquitous computing. *Computer Communications 26*, 11 (2003), 1168–1176.

[142] Raposo, A. B., Magalhães, L. P., Ricarte, I. L. M., and Fuks, H. Coordination of collaborative activities: A framework for the definition of tasks interdependencies. In *Groupware, 2001. Proceedings. Seventh International Workshop on* (2001), IEEE, pp. 170–179.

[143] Raskar, R., Welch, G., Cutts, M., Lake, A., Stesin, L., and Fuchs, H. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (1998), ACM, pp. 179–188.

[144] Rattenbury, T., and Canny, J. Caad: an automatic task support system. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2007), ACM, pp. 687–696.

[145] Reddy, M. C., Dourish, P., and Pratt, W. Coordinating heterogeneous work: Information and representation in medical care. In *ECSCW 2001* (2001), Springer, pp. 239–258.

[146] Rekimoto, J. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *Proceedings of the 10th annual ACM symposium on User interface software and technology* (1997), ACM, pp. 31–39.

[147] Rekimoto, J., and Saitoh, M. Augmented surfaces: a spatially continuous work space for hybrid computing environments. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (1999), ACM, pp. 378–385.

[148] Robertson, G., Van Dantzich, M., Robbins, D., Czerwinski, M., Hinckley, K., Risden, K., Thiel, D., and Gorokhovsky, V. The task gallery: a 3d window manager. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2000), ACM, pp. 494–501.

[149] Rogers, Y. A brief introduction to distributed cognition. *Retrieved July 24* (1997), 1997.

[150] Rogers, Y. New theoretical approaches for human-computer interaction. *Annual review of information science and technology 38*, 1 (2004), 87–143.

[151] Rogers, Y. Moving on from weiser's vision of calm computing: Engaging ubicomp experiences. In *UbiComp 2006: Ubiquitous Computing*. Springer, 2006, pp. 404–421.

[152] Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R. H., and Nahrstedt, K. A middleware infrastructure for active spaces. *IEEE pervasive computing 1*, 4 (2002), 74–83.

[153] Saha, D., and Mukherjee, A. Pervasive computing: a paradigm for the 21st century. *Computer 36*, 3 (2003), 25–31.

[154] Salvucci, D. D., Taatgen, N. A., and Borst, J. P. Toward a unified theory of the multitasking continuum: from concurrent performance to task switching, interruption, and resumption. In *Proceedings of the SIGCHI*

Conference on Human Factors in Computing Systems (2009), ACM, pp. 1819–1828.

[155] Salzman, M., and Palen, L. The tools we live by: a description of personal support media in work life. Tech. rep., Technical Reports, Computer Science, University of Colorado, Boulder, 2004.

[156] Santosa, S., and Wigdor, D. A field study of multi-device workflows in distributed workspaces. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing* (2013), ACM, pp. 63–72.

[157] Schilit, B., Adams, N., and Want, R. Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on* (1994), IEEE, pp. 85–90.

[158] Schmidt, A., Beigl, M., and Gellersen, H.-W. There is more to context than location. *Computers & Graphics 23*, 6 (1999), 893–901.

[159] Schmidt, D., Chehimi, F., Rukzio, E., and Gellersen, H. Phonetouch: a technique for direct phone interaction on surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology* (2010), ACM, pp. 13–16.

[160] Schmidt, D., Seifert, J., Rukzio, E., and Gellersen, H. A cross-device interaction style for mobiles and surfaces. In *Proceedings of the Designing Interactive Systems Conference* (2012), ACM, pp. 318–327.

[161] Schmidt, K., and Bannon, L. Taking cscw seriously. *Computer Supported Cooperative Work (CSCW) 1*, 1-2 (1992), 7–40.

[162] Schopenhauer, A. *The world as will and representation*, vol. 1. Courier Dover Publications, 2012.

[163] Shen, J., Li, L., Dietterich, T. G., and Herlocker, J. L. A hybrid learning system for recognizing user tasks from desktop activities and email messages. In *Proceedings of the 11th international conference on Intelligent user interfaces* (2006), ACM, pp. 86–92.

[164] Sicard, L., Tabard, A., Hincapié-Ramos, J. D., and Bardram, J. E. Tide: Lightweight device composition for enhancing tabletop environments with

smartphone applications. In *Human-Computer Interaction–INTERACT 2013*. Springer, 2013, pp. 177–194.

[165] Smith, G., Baudisch, P., Robertson, G., Czerwinski, M., Meyers, B., Robbins, D., and Andrews, D. Groupbar: The taskbar evolved. In *Proceedings of OZCHI* (2003), vol. 3, p. 10.

[166] Strauss, A. The articulation of project work: An organizational process. *The Sociological Quarterly 29*, 2 (1988), 163–178.

[167] Strauss, A. L. *Creating sociological awareness*. Transaction Publishers, 1991.

[168] Streitz, N. A., Geissler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., and Steinmetz, R. i-land: an interactive landscape for creativity and innovation. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (1999), ACM, pp. 120–127.

[169] Streitz, N. A., Rexroth, P., and Holmer, T. Does "roomware" matter? In *Proceedings of the Fifth European Conference on Computer Supported Cooperative Work* (1997), Springer, pp. 297–312.

[170] Suchman, L. A. *Plans and situated actions: the problem of human-machine communication*. Cambridge university press, 1987.

[171] Tabard, A., Hincapié Ramos, J. D., and Bardram, J. The elabbench in the wild: supporting exploration in a molecular biology lab. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), ACM, pp. 3051–3060.

[172] Tabard, A., Hincapié-Ramos, J.-D., Esbensen, M., and Bardram, J. E. The elabbench: an interactive tabletop system for the biology laboratory. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (2011), ACM, pp. 202–211.

[173] Tang, C., and Carpendale, S. Evaluating the deployment of a mobile technology in a hospital ward. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work* (2008), ACM, pp. 205–214.

[174] Tell, P., Babar, M. A., and Grundy, J. A preliminary user evaluation of an infrastructure to support activity-based computing in global software development (abc4gsd). In *Global Software Engineering (ICGSE), 2013 IEEE 8th International Conference on* (2013), IEEE, pp. 100–109.

[175] Terrenghi, L., Quigley, A., and Dix, A. A taxonomy for and analysis of multi-person-display ecosystems. *Personal and Ubiquitous Computing 13*, 8 (2009), 583–598.

[176] Trigg, R. H., Blomberg, J., and Suchman, L. Moving document collections online: The evolution of a shared repository. In *ECSCW'99* (1999), Springer, pp. 331–350.

[177] Voida, S., and Mynatt, E. D. It feels better than filing: everyday work experiences in an activity-based computing system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2009), ACM, pp. 259–268.

[178] Voida, S., Mynatt, E. D., and Edwards, W. K. Re-framing the desktop interface around the activities of knowledge work. In *Proceedings of the 21st annual ACM symposium on User interface software and technology* (2008), ACM, pp. 211–220.

[179] Voida, S., Mynatt, E. D., MacIntyre, B., and Corso, G. M. Integrating virtual and physical context to support knowledge workers. *IEEE Pervasive Computing 1*, 3 (2002), 73–79.

[180] Vyas, D., Chisalita, C. M., and Van Der Veer, G. C. Affordance in interaction. In *Proceedings of the 13th Eurpoean conference on Cognitive ergonomics: trust and control in complex socio-technical systems* (2006), ACM, pp. 92–99.

[181] Vygotsky, L. S. *Mind in society: The development of higher psychological processes*. Harvard university press, 1980.

[182] Vygotsky, L. S. *The collected works of LS Vygotsky: Problems of the theory and history of psychology*, vol. 3. Springer, 1997.

[183] Want, R., Hopper, A., Falcao, V., and Gibbons, J. The active badge location system. *ACM Transactions on Information Systems (TOIS) 10*, 1 (1992), 91–102.

[184] Want, R., Schilit, B. N., Adams, N. I., Gold, R., Petersen, K., Goldberg, D., Ellis, J. R., and Weiser, M. An overview of the parctab ubiquitous computing experiment. *Personal Communications, IEEE 2*, 6 (1995), 28–43.

[185] Weiser, M. The computer for the 21st century. *Scientific american 265*, 3 (1991), 94–104.

[186] Weiser, M. Hot topics-ubiquitous computing. *Computer 26*, 10 (1993), 71–72.

[187] Weiser, M. Some computer science issues in ubiquitous computing. *Communications of the ACM 36*, 7 (1993), 75–84.

[188] Weiser, M. The world is not a desktop. *interactions 1*, 1 (1994), 7–8.

[189] Weiser, M., and Brown, J. S. The coming age of calm technology. In *Beyond calculation*. Springer, 1997, pp. 75–85.

[190] Wilson, A. D., and Sarin, R. Bluetable: connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In *Proceedings of Graphics interface 2007* (2007), ACM, pp. 119–125.

[191] Winograd, T., and Flores, F. *Understanding computers and cognition: A new foundation for design*. Intellect Books, 1986.

[192] Zhou, X., Ackerman, M. S., and Zheng, K. I just don't know why it's gone: maintaining informal information use in inpatient care. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2009), ACM, pp. 2061–2070.

Part II

Systems and Technology

Paper 1: co-Activity Manager

Title of paper

Activity-Centric Support for Ad Hoc Knowledge Work – A Case Study of co-Activity Manager

Authors:

Steven Houben, *Jakob E. Bardram, Jo Vermeulen, Kris Luyten and Karin Coninx*

Published:

*In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems **(CHI '13)**. ACM, 2013.*

Abstract:

Modern knowledge work consists of both individual and highly collaborative activities that are typically composed of a number of configuration, coordination and articulation processes. The desktop interface today, however, provides very little support for these processes and rather forces knowledge workers to adapt to the technology. We introduce co-Activity Manager, an activity-centric desktop system that (i) provides tools for ad hoc dynamic configuration of a desktop working context, (ii) supports both explicit and implicit articulation of ongoing work through a built-in collaboration manager and (iii) provides the means to coordinate and share working context with other users and devices. In this paper, we discuss the activity theory informed design of co-Activity Manager and report on a 14 day field deployment in a multi-disciplinary software development team. The study showed that the activity-centric workspace supports different individual and collaborative work configuration practices and that activity-centric collaboration is a two-phase process consisting of an activity sharing and per-activity coordination phase.

Activity-Centric Support for Ad Hoc Knowledge Work – A Case Study of co-Activity Manager

Steven Houben, Jakob E. Bardram
The Pervasive Interaction Technology Laboratory
IT University of Copenhagen, Rued
Langgaardsvej 7, DK-2300 Copenhagen, DK
{shou, bardram}@itu.dk

Jo Vermeulen, Kris Luyten, Karin Coninx
Expertise Centre for Digital Media
Hasselt University – tUL – iMinds,
Wetenschapspark 2, B-3590 Diepenbeek, BE
firstname.lastname@uhasselt.be

ABSTRACT
Modern knowledge work consists of both individual and highly collaborative activities that are typically composed of a number of configuration, coordination and articulation processes. The desktop interface today, however, provides very little support for these processes and rather forces knowledge workers to adapt to the technology. We introduce co-Activity Manager, an activity-centric desktop system that (i) provides tools for ad hoc dynamic *configuration* of a desktop working context, (ii) supports both explicit and implicit *articulation* of ongoing work through a built-in collaboration manager and (iii) provides the means to *coordinate* and share working context with other users and devices. In this paper, we discuss the activity theory informed design of co-Activity Manager and report on a 14 day field deployment in a multidisciplinary software development team. The study showed that the activity-centric workspace supports different individual and collaborative work configuration practices and that activity-centric collaboration is a two-phase process consisting of an activity sharing and per-activity coordination phase.

Author Keywords
Activity Theory, Desktop Interface, Activity-Centric Computing, Collaborative Work

ACM Classification Keywords
H.5.2 Information interfaces and presentation:User Interfaces. - Graphical user interfaces.

INTRODUCTION
Knowledge work is typically composed of both *individual* and highly *collaborative* work. This means that knowledge workers use personal computing devices to perform individual tasks and activities that are part of a larger collaborative working context. The individual work is in many cases dependent on and driven by information that is provided by coworkers or other stakeholders that are part of the overall activities. There is thus a high demand for collaboration amongst co-workers which results in an increased level of project fragmentation due to the large number of tasks and activities one typically performs at the same time [7, 11, 25]. Nevertheless, despite this highly collaborative nature of knowledge work, many users still want to tailor their part of the work to their personal preferences.

CHI 2013, April 27–May 2, 2013, Paris, France.

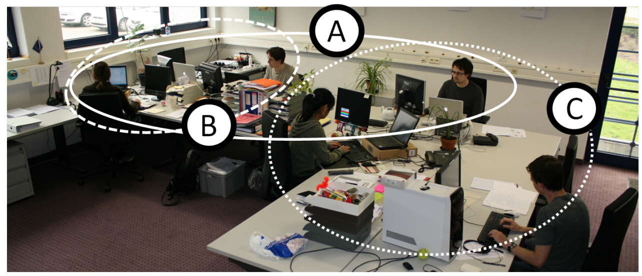


Figure 1. A multidisciplinary software development team. The groups (a,b,c) symbolize some of the collaborative relations between the members.

Figure 1 shows a multidisciplinary software development team. This team collaborates on different projects with different colleagues in partially overlapping subgroups (marked as *a*, *b* and *c*). For these projects the team members continuously collaborate on shared parallel activities, which require some form of coordination (e.g. sharing files and being aware of each other's updates on those files) between individually performed work.

Current desktop interfaces, however, provide very little support for this type of activity-centric collaboration. Although many stand-alone tools are available to users, studies have shown that these are frequently the source of interruptions and project fragmentation because they are disconnected from other tools and functions, which are used in the same activity [10, 18, 21] .

In general, research has pointed out that there is a fundamental mismatch between the design and functionality of modern desktop systems and the need for more activity-oriented support in a work setting [5, 23]. As we will discuss in the 'Related Work' section, many of the proposed solutions focus very much on the individual interaction with the desktop interface, often minimizing the importance of the collaborative aspect of knowledge work.

This paper introduces *co-Activity Manager* (cAM), an activity-centric multi-user desktop manager integrated with the Windows 7 operating system. co-Activity Manager was specifically designed to facilitate collaboration by supporting:

1. **Activity-centric desktop management** to minimize the configuration work for different parallel activities;
2. **Activity sharing** that allows users to share and deploy activities in a collaborative multi-device setup; and
3. **Activity-centric collaboration tools** (including file sharing and messaging) thereby moving communication channels into the activity abstraction.

We present the design and implementation of co-Activity Manager, report on a 14 day field deployment in a multi disciplinary software development team, and discuss the lessons learned from the implementation and deployment of the system.

RELATED WORK

The earliest approach to update the desktop interface was the introduction of virtual workspaces in the Rooms system [17]. There are, however, many other systems and applications that use virtual workspaces as an approach to solve some of the problems of the desktop interface. Systems such as Groupbar [33], Quickspace [20] and Task Gallery [31] use the desktop workspace in different ways. Unfortunately, although virtual desktops partly solve the workspace shortage, they increase the cognitive load when more than four desktops are used, as users must remember themselves what information is located in which virtual desktop [30].

Amongst others, Kaptelinin and Czerwinski [23] summarized a number of novel approaches that have been proposed to restructure the desktop interface. These novel approaches are based on fundamental concepts such as time (e.g., LifeStreams [15]), the relation between information (e.g., Haystack [1]), physics (e.g., Bumptop [2]), virtual workspaces (e.g., Rooms [17] and GroupBar [4, 33]), tasks (e.g., TaskTracer [8]) or activities (e.g., ABC [4], Giornata [35], CAAD [29] and Umea [22]). In this paper, we follow the latter approach.

Several approaches have introduced higher-level structures in the form of tasks or activities. Taskmaster [6], e.g. uses a traditional email layout to organize activities. This implementation is very helpful as a tool to enhance communication and to keep track of work flow but is too limiting as the users can only use a set of pre-defined objects inside the application rather than their normal desktop workspace. Some solutions try to reduce the mental load by automatically generating context (files and folders) that is based on data collected by monitoring the user and the system. CAAD [29] automatically generates context structures based on the user's work flow.

Task Tracer [13] creates task profiles based on user actions. Umea [22] monitors users' actions in order to create activity histories that it uses to determine the relevance of resources. Umea also includes personal information management tools and communication capabilities. These are, however, centralized in an standalone application rather than integrated into existing technology (operating systems). This forces users to work in two separate modes which could lead to the development of two mental models: one of the desktop system and one of the activity system.

A number of approaches aim to address the problems of knowledge workers in the digital age by integrating activity management into the desktop interface (e.g [22, 29, 33]). Project Colletta [28], Giornata [35] and the ABC system [4] closely integrate with the operating system by using a virtual desktop-like system as a structuring mechanism for activities. The latter two also consider communication and collaboration. Giornata [35] provides a contextually populated *contact palette* that can be used to share files via email and also serves as a visual cue on the amount of unread emails. In the ABC system, file sharing and real-time collaboration are supported through a pervasive framework that was designed for hospital environments [5]. Finally, Activity Explorer [27] successfully introduced an activity sharing system but limits its approach to predefined objects that are confined inside the application.

In summary, prior work has primarily focussed on either integrating individual activity/task management in the desktop interface or shared collaborative task management in separate, stand-alone applications. However, as pointed out by Moran and Zhai [26], the issues they address are dimensions of the same problem. We argue that in order to support a system that allows users to truly manage the activities they perform, the configuration of tools, coordination with collaborators and articulation (meaning distribution of awareness) of their ongoing work needs to be integrated into one desktop system. We therefore position this work at the intersection of these related works and re-analyse these three related problems informed by activity theory.

The core contribution of this paper is the design of a desktop manager that supports personal and collaborative activity-centric workflows with integrated activity-centric collaboration and interruption management tools. The novelty of this work is in how configuration, collaboration and awareness tools are included in a desktop interface; thus allowing them to become an inherent part of the integrated activity-centric workflow, rather than another tool that introduced interruptions or that needs repeated (re-)configuration.

MOTIVATION

Based on the literature review above and informed by the seven dimensions of change as described by Moran and Zhai [26], we focus on the following three problems that knowledge workers experience in working with contemporary desktop interfaces:

Pr1 : Configuration – it is cumbersome to structure the desktop and manage documents according to the wide variety of tasks one typically performs. Especially in a work setting with many parallel activities, switching between these working contexts requires a constant reconfiguration of the desktop environment thereby increasing the mental and work load of the users. The general movement from "low-level tasks to higher-level activities" (dimension 7 in [26]) requires a re-conceptualisation of the desktop interface.

Pr2 : Articulation – despite the shift "from personal to interpersonal to group to social interaction" (dimension 6 in [26]), communication and collaboration tools are separated from the tasks that use them. Most are also explicit, which means that unless users actively share information on their work context and progress, this information is not available for other users.

Pr3 : Coordination – as we are moving "from interaction with one device to interaction with information through many devices" (dimension 3 in [26]) there is still little support for sharing activities between different users and devices. The desktop work setting is intrinsically tied to the individual work context, and there is no structural support for sharing this work context with other users (which might depend on its content) or other devices.

Survey

To explore if these three problems $Pr1$, $Pr2$ and $Pr3$ are actually considered real problems by modern knowledge workers, we conducted a large scale survey. For this survey we recruited 145 participants (58% male, 42% female, average age of 33 (σ=10,06)) to reflect on these three problems through 25 representative 5-point Likert scale questions (1 = strongly disagree; 5 = strongly agree) and an open comment section (we received 112 comments). Most respondents (89,66%) described themselves as knowledge workers while 98,62% claimed to use a computer for work. Participants rated their computer literacy very high (*μ= 3.93; $\tilde{x}$= 4; σ= 0.87 on a 5-point likert scale*) and have different backgrounds (28% academic/research, 27% education, 22% IT and 23% from various other backgrounds including government, healthcare, design and media).

On average, participants owned about three devices (*μ= 2.70; σ= 1.59*) that they actively use, which is in line with the findings of Dearman and Pierce [12] and demonstrates that knowledge workers nowadays indeed use multiple devices. Interestingly, participants were very divided on whether the general management of the desktop is a problem (μ= 3.03; $\tilde{x}$= 3; σ= 1.26) and if the desktop is cluttered with to many icons and windows (μ= 2.8; $\tilde{x}$= 3; σ= 0.97). In the open question section, however, there were numerous comments that did in fact expose serious issues:

> *"If you don't have a well thought out workflow, your desktop/computer becomes a mess. And if you want to customize your interface, you need programming skills".*

> *"Because of the diversity in our team (Windows/OSX users, men/women, French/English speaking, working at the office or from home,..)', we need an accessible, robust and flexible system".*

Email is considered one of the most important means for collaboration (μ= 4.4; $\tilde{x}$= 5; σ= 0.83). This was emphasized by the fact that most respondents generally do not mind being interrupted from their work by email (μ= 3.5; $\tilde{x}$= 4; σ= 1.09). Interrupts via instant messaging on the other hand were not appreciated (μ= 2.4; $\tilde{x}$= 2; σ= 1.14). Participants also confirmed that they regularly share documents with colleagues (μ= 4.09; $\tilde{x}$= 4; σ= 0.90).

It was clear that sharing files and folders with contacts (μ= 3.3; $\tilde{x}$= 4; σ= 1.01) and devices (μ= 3.5; $\tilde{x}$= 4; σ= 1.03) should be much easier. Surprisingly, most respondents experience few problems in managing their open windows (μ= 2.6; $\tilde{x}$= 2; σ= 1.02). This might be explained by the small number of open windows (7) that populate their desktop (μ= 6.48; $\tilde{x}$= 5; σ= 4.29). Other comments revealed that managing child windows of a multi-window application often causes frustration and is considered to be more difficult to manage as single window applications.

In general, the survey responses demonstrated issues with the lack of separation between applications, tasks and data by the operating system. Users have to deal with the difficult task of organizing the different activities in a *workflow*[1] ($Pr1$). Unfortunately, no appropriate mechanisms are provided by current systems to automate this which leads to task fragmentation as demonstrated by [7, 11, 25]. The results of the survey also confirmed problem $Pr2$: most communication tools (such as email, instant messaging,...) require the user to provide nearly instant feedback, even if the communication received is not related to the task at hand. This is an interruption of the workflow [3, 9, 10, 36] and an important cause of fragmentation of work over time. Furthermore, users expect the availability of multiple devices [12] for similar or identical tasks, and also expect to be able to make smooth transitions between devices dedicated for work purposes ($Pr3$).

Activity Theory

To deal with these three fundamental problems, we ground our design in Activity Theory (AT) [14, 24], a descriptive psychological framework that seeks to explain human activity as a mediated and asymmetrical relation between a subject, an object and a community. An activity is engaged by a *subject* (S) that translates a need into a motive or *object* (O). This S - O relation is embedded in a *community* (C) involved in the creation of this relation. As such, the community plays an important role in the creation, development and outcome of the activity. AT suggests making these social structures part of the activity itself rather than defining them as merely external influences [24]. Finally, the S - O - C relation is mediated by *tools*, *rules* and *division of labour*. These mediators determine *how* the activity is engaged and *how* it is framed in a broader social context.

In order to make Activity Theory more concrete in context of the three problems of the contemporary desktop interface, we present three guidelines (labelled $G1$, $G2$ and $G3$ for future reference) that map directly on the problems discussed earlier. Even though these guidelines are very high level, the next section will show how these guidelines are refined into design properties that help apply Activity Theory to the design of activity-centric interactive systems.

$G1$: *Provide a shared higher level structure for organizing tasks, documents and resources*

An Activity Theory informed system should organize work using meaningful structures. Since human activity can not be reflected in a static structure but needs an evolving and dynamic structure, users should be able to *redefine* and *change* activities in the course of their work. This implies that users should be able to define and use activities according to their personal preferences. Users might use the same activities but in a different context, or just switch between activities

[1] With workflow we refer to the work practices of users.

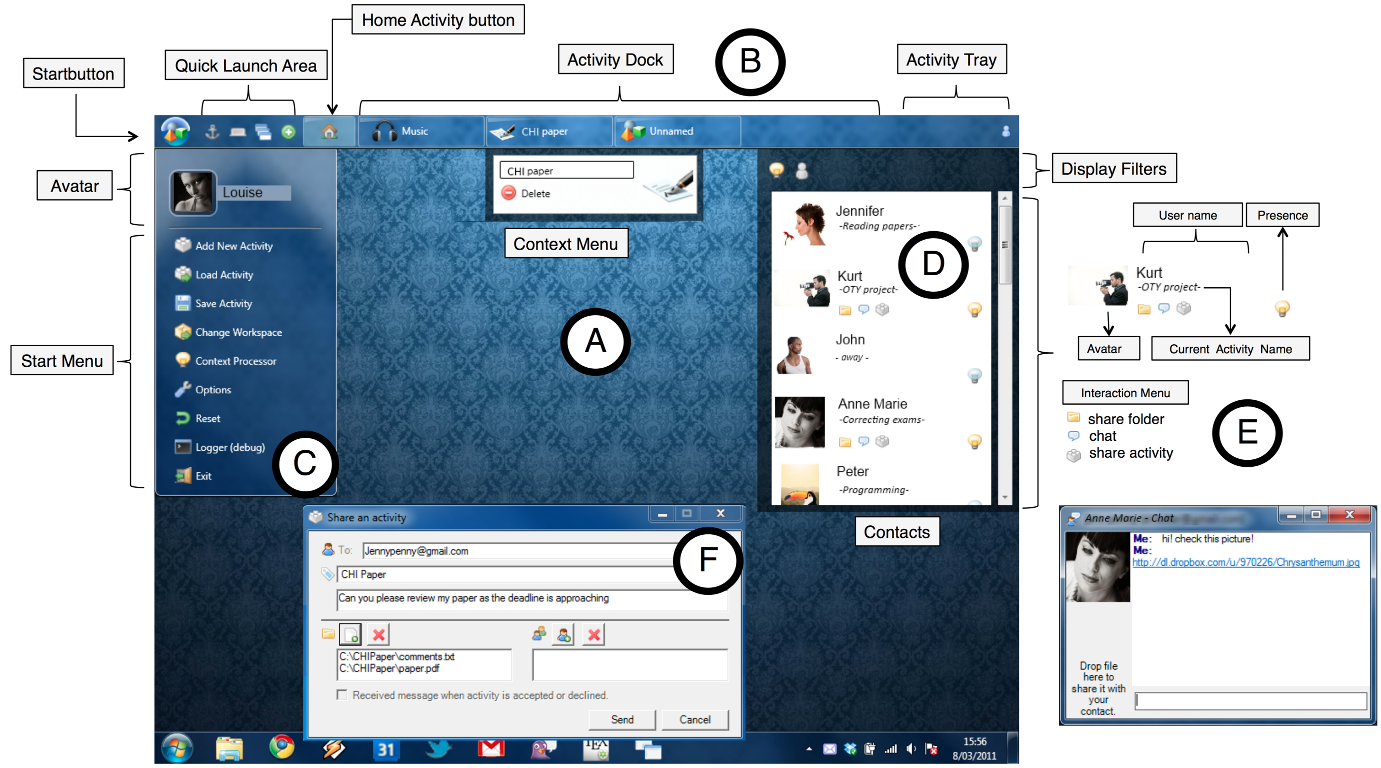


Figure 2. The interface of co-Activity Manager consists of (A) a per-activity workspace, (B) an activity task bar to visualize activities to the user, (C) an activity start menu to manage activities and applications, and (D) a collaboration manager to interact and share with contacts. Each contact is visualized with an avatar, name and status field. The interaction menu (E) can be used to share a folder, chat or share an activity (F).

and save the current one to resume it later. Activity-centric software structures the workflow by connecting resources, data and activities in a semantic network specifying what resources and data are required or used by which activities. The user can manually link resources to activities or have it done automatically by the software. (Targeting $Pr1$).

$G2$: *Support collaboration and sharing*
Activity Theory includes the community explicitly and an activity-centric system should reflect this. We identify *"community"* as the participants of a social network that is making use of an activity centric system. This means that people have a shared interest in the system which should thus allow multiple people to participate in its usage and provide functionality to share (parts of) the activity of one user with other user. There must be support to distribute representations of the ongoing activities to both the user itself and other collaborators. Consistent with guideline $G1$, sharing an activity implies providing the structure of the activity as it was defined by the initiator of an activity. Collaboration and communication are part of the structure of activity and become part of an activity itself. Because communication and collaboration are an integral part of the activity, it must also be included in the representation. (Targeting $Pr2$).

$G3$: *Ensure transparent context-aware views on activities*
Most activities are not limited to a specific type of device, environment or setting which exposes the necessity of a *portable activity structure* that has the ability to transcend the individual device or user. A device-specific view on activities can be built around this portable structure. Additionally, besides the different platforms and devices, other contextual events come into play. E.g. the physical location of an individual could be used to determine the relevance of an activity. Because activity is a description that depends on space and time, it is relevant for an activity-centric system to capture both the history and context of each activity. The combination of *activity transparency* and *context awareness* provides a computational representation that maps very well on the Activity Theoretical structure of human activity. (Targeting $Pr3$)

CO-ACTIVITY MANAGER

co-Activity Manager (cAM) (Figure 2) [19] is a collaborative activity-centric desktop interface that (i) provides an activity workspace that supports ad hoc configuration of the active desktop working context, (ii) includes an activity sharing mechanism that allows for the distribution of an activity workspace, and (iii) provides built-in tool support for activity-centric collaboration. cAM deals with project fragmentation as well as communication interruptions by providing users with an activity-centric interface that allows them to (re-)organize their documents, applications and files as well as their communication and collaboration with other participants in activities.

Activity-Centric Design Properties

Activity-centric computing has proven to be a useful computing paradigm [23, 35], but remains difficult to translate into concrete software features. We therefore refine our three guidelines into six design properties which we used to inform the design of co-Activity Manager.

$D1$: *Activity-Oriented Workspace*: the design of interactive systems for knowledge workers should be focused on supporting activities. Because activities are computational representations of human activity, they should be visible in the user interface. To optimize this effect, the system should integrate with existing technology. Users must be able to create, edit, manage, consume and switch between existing computational representations of activity (refinement of $G1$).

$D2$: *Activity Sharing*: because activities are the central focus of this paradigm, activities must also (like files) be interchangeable with other users and devices. This will allow users to externalize (parts of) their activities to other contexts. Activities must therefore be platform and device independent as well as interchangeable. Activity sharing is thus not limited to the interchange between users, but also between devices (refinement of $G2$ and $G3$).

$D3$: *Activity-Centric Communication and Collaboration*: instant messaging, file sharing and other community related actions should be embedded into the activity structure to minimize “out-of-context” interruptions: interruptions generated in the context of another activity that interfere with the workflow on the current activity (refinement of $G2$).

$D4$: *Activity-Centric Presence*: as users will be able to choose their workflow as well as collaborators in the context of each activity, an advanced presence system that allows users to define availability according to the current activity is required (refinement of $G2$).

$D5$: *Activity-Centric Cloud Storage*: because activities must be interchangeable between devices and platforms and typically need to be persistent over longer periods of time, they have to be transparently stored in the cloud instead of on the local device (refinement of $G3$).

$D6$: *Activity-Oriented Context Recognition*: an Activity Theory-informed system should not only focus on providing mechanisms to create and use activities, but also to recognize changing context, such as changes in location (refinement of $G3$).

Activity-Oriented Workspace

‘Activities’ are implemented in co-Activity Manager as a data model that includes all resources, contacts and other (meta) information relevant to the ongoing desktop work context in an effort to reflect a physical task or activity a user is performing. The system thus provides a first class object that aligns the computational representation of data with the intention of use; the task or activity of the user.

co-Activity Manager extends the Windows 7 desktop interface with an activity-centric workspace. For each activity, a separate virtual desktop (Figure 3), that confines the working context defined by the activity, is constructed. We label this augmented virtual desktop as an *activity workspace* (on Figure 2 A). The scope of all opened windows, files and documents is limited to the activity workspace related to that activity (although windows can be transferred between or duplicated over desktops). When the user switches between activities, the workspace and Windows task manager is repopulated with windows that are related to that activity. Each activity workspace is equipped with a custom desktop folder that contains the data related to that activity. Users can simply *pile* their files and documents on the desktop per activity rather than using the hierarchical structure of the inherent file system (though this is still possible).

In order to present activities to the user, we designed an activity taskbar (Figure 2 B) that is used to manage and work on activities. By clicking the ’add’ action button or by using the start menu (on Figure 2 C) the user can create a new activity. Newly created activities are by default *anonymous* and given a *default* name and icon. The user can choose to keep the activity anonymous or configure it for more persistence. The name, icon and other information can be changed at any time by simply launching the context menu. For each newly created (or loaded) activity, the system adds a new activity button to the dock. By pressing an activity button, the associated activity workspace is loaded causing the desktop icons, windows and the build-in collaboration manager (Figure 2 D) to update.

All activities that are located on the activity dock of the taskbar are part of the same *activity workbench*. Users can create and delete workbenches or change existing workbenches to configure them according to their personal work preferences. This can happen both before and after sharing an activity with others. Activity workbenches are introduced as a feature to deal with both activity clutter or overpopulation of the activity taskbar as well as provide a meta structure to manage activities.

For the design of this activity taskbar, we mimicked the standard Windows 7 taskbar but redesigned it to be suited for activity management (Figure 2 B) as demonstrated in prior work [4, 33]). By exploiting user’s *familiarity* with the taskbar, we expected to get a higher level of user acceptance since activity management can be operated similar to how one manages applications. Documents, windows and applications are no longer loosely coupled elements that float around the desktop but are embedded in an activity. When one switches activities, all documents, windows and applications will also change accordingly.

Activity-Centric Cloud Services

To overcome the fragmentation that is caused by the wide variety of devices being used [12], we make use of cloud storage. This means that files, documents and activities are stored online but are transparently accessible through the desktop interface. This approach is an effort to integrate the Personal Information Cloud [26] into cAM. cAM allows users to save and load activities, and their containing files and configuration from the cloud storage. cAM automatically builds an XML file of the activity and saves this along with all resources. This file can afterwards be imported on another devices that runs cAM or another piece of software that supports this format. In the current version, cAM integrates with the

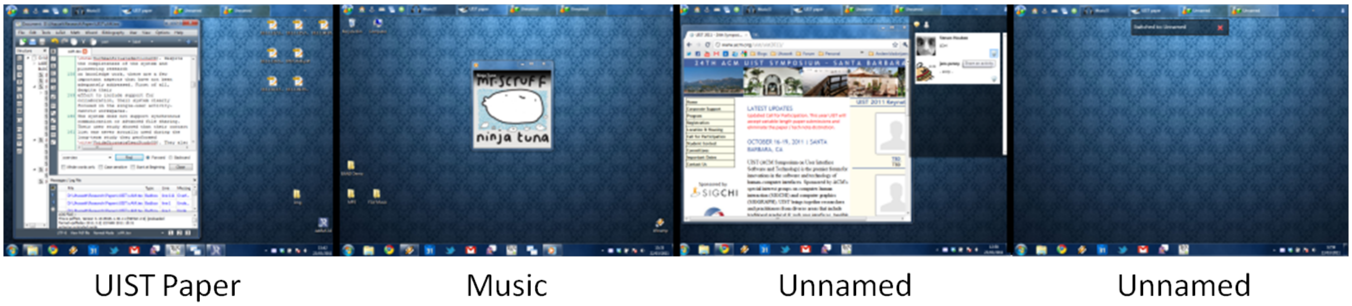


Figure 3. Each activity has its own activity workspace.

Dropbox API [2] to support cloud storage, but because of the use of the XML standard, any type of storage can be used.

Activity-Centric Communication and Collaboration

The activity management system of co-Activity Manager does not only consider windows, applications and documents but also includes tools for communication and collaboration. Since most local task fragmentation is caused by interruptions of instant messengers and other communication tools, we argue that the workflow can not be restructured without explicitly including built-in communication and collaboration. By including both communication and collaboration tools in the structure of an activity, we thus aim to decrease work fragmentation.

cAM includes a collaboration manager (Figure 2 D) that can be used to interact with collaborators. First of all, we support standard *chat* messages. The chat window is equipped with an automatic sharing system, that allows users to drag and drop documents they want to share on top of the chat windows. These files are then automatically uploaded to the cloud storage and shared through the chat window. Secondly, the user can define a *shared folder* for each contact per activity. All these folders and their content are stored in the cloud storage and can be used as a persistent sharing mechanism or to share large amounts of data. Finally, users can also *share activities* (on Figure 2 F).

Because one of our goals was to create a *usable* and *scalable* system, we use the Gmail infrastructure for communication and collaboration. Since their email system is free, already widespread, very robust and also provides a distributed contact list, it was best suited for integration into co-Activity Manager. Additionally, Gmail also provides XMPP [32] support for real-time communication, which allows us to develop custom XMPP extensions for other purposes (including activity sharing). Summarized, cAM integrates messaging, file sharing and activity sharing into the desktop interface.

Activity-Centric Presence

As the collaboration manager (on Figure 2 D) is included in the design of the activity workspace, it can be populated for each activity. Users can choose which contacts they finds relevant for each activity. By adding these contacts to the collaboration list of the specific activity, they define a group of contacts who are allowed to communicate with them and who are made visible for the users themselves. By default, all collaborators are added to the list, but not added to the activity.

The collaboration manager also provides association and online/offline filters (Figure 2 E) to customize the list to personal preferences for each activity. These filters remove or show offline/online or associated/not associated contacts. When users switch between activities, the collaboration list is updated and the contacts will see the user appear as offline or online depending on if they are added to the loaded activity or not. If the user is online, the name of the activity the user is currently engaging will be distributed to all collaborators.

This activity-centric presence system allows users to not only control their workflow but also their communication flow. We believe that by allowing users to control who is allowed to interrupt them, they can also better control the entire workflow, therefore decreasing task and work fragmentation. In order to add or remove a user, the user simply clicks the light bulb (on Figure 2 E) button that is located on the interaction menu of each contact. When the light is on, the contact is associated with the activity and thus allowed to interact with the user. In this case, the user will be reported as available in the contact list. When the light is off, the contact is not associated with the activity. In this case, the user will be reported as unavailable in the contact list (but still online) and cannot interrupt the current activity by sending a direct message. In order to distribute these presence changes to all contacts, we use a custom XMPP extension that carries the activity-based presence information to all collaborators.

Activity Sharing

Since the entire approach is focused on activities and its components, we also argue that this structure must be interchangeable between users and devices. This implies that the user is not only defining their own ad hoc workflow, but can also suggest the workflow of their collaborators or other devices. This vision also embraces an entire activity life cycle rather than perceiving activities as a individual local organizational structure.

In the collaboration manager (Figure 2 D) the user can launch an activity delegator window (on Figure 2 F), that can be used to describe the activity the user wants to share. These descriptions include a name, a textual description of the goals and motivation of the activity, a set of resources (e.g. local documents, folders, applications), relevant contacts and the previously recorded history as configured in the local activity

[2] https://www.dropbox.com/developers

workspace. After resources are stored in the cloud, the system sends the description of the activity to the selected contact.

The receiving user is notified that there is a new activity; accepting it will cause the system to construct a new local activity, based on the description in the XML file that was received. The name, icon, description and other activity information is used to define the local activity; the related resources are downloaded from the cloud to the workspace of the activity and the collaboration manager is populated with relevant contacts. The activity is grabbed from the cloud and locally deployed. This mechanism thus allows users to pre-configure an activity and share this configuration with other collaborators, who in turn can deploy and customize it based on their personal preferences.

Activity-Oriented Context Recognition

Although we want to emphasize the importance of context recognition for a holistic approach to activity representation, this has not been the main focus of co-Activity Manager. As a proof of concept, we developed a workbench switcher that dynamically changes activity workbench depending on the connected wireless network SSID. This implies that the user can couple a set of related activities to a specific physical environment. The contextual workbench switcher was included to deal with activity clutter. We did not include any additional support for context awareness in this prototype, but included an API that can be used by external context-awareness or monitoring systems such as Subtle [16] and PersonalVibe [8] in order to update or change the interface based on external events.

FIELD DEPLOYMENT

co-Activity Manager was deployed for a two week period in a five-person multidisciplinary software development team – consisting of software engineers, a graphic designer and historian. The team specializes in developing interactive setups for cultural heritage sites such as museums or tourist attractions. We selected this software development team for four reasons: (i) they spend several hours a day using a desktop interface; (ii) they all tend to collaborate in (partially) overlapping subgroups (see Figure 1); (iii) due to its multidisciplinary character, the team is composed of people with varying computer expertise; and (iv) they each currently tend to structure their workflow in different ways, which allowed us to discuss the value and potential of activities as a workflow structuring mechanism.

The primary goal of this evaluation was not to assess the *usability* of co-Activity Manager but rather to explore the *feasibility* of a collaborative *activity-centric desktop interface* for knowledge workers through a case study. Because this cannot be tested in a lab environment [34], we deployed the system on the primary computer of all participants and asked them to use it for their daily work activities during office hours. Although this study provides valuable insights in the multi-user usage patterns and immediate issues arising from using this type of system in a real world setting, additional longitudinal studies are needed to confirm the usability for a broader spectrum of users.

Experimental Setup

The team consisted of five people in total (3 male, 2 female; mean age = 31): three software engineers (P1, P2 and P3), a graphic designer (P4) and a historian (P5). Before deploying the system, we conducted a short pre-study interview in which we discussed with participants how they structure their work on the desktop and with which other team members they regularly collaborate. Before the deployment, participants received a short demonstration of the features of co-Activity Manager.

For the two-week deployment, participants were instructed to use the system in the course of their day-to-day work. During this period, participants were observed and interviewed regularly to detect potential problems and discover emerging behaviour. At the end of this period, participants were asked to complete a short questionnaire which was used as a basis for a semi-structural interview in which we discussed the user's experiences and opinions of the usefulness of the concepts of co-Activity Manager.

Results

During the pre-study interviews, all participants reported doing both *individual* and *highly collaborative* work. Moreover, we also observed that participants used quite diverse ways of structuring their workflow, including the way they stored important documents, organized their windows and managed communication and collaboration with others. We were interested to see how these differences would affect their appreciation of co-Activity Manager, and if co-Activity Manager would be flexible enough to cope with these different ways of working.

Table 1 lists the results of the 5-point likert scale survey which ranges from "not at all useful" (1) to "very useful" (5).

Questions	*Min*	*Q1*	$\tilde{x}$	*Q3*	*Max*	*Iqr*
Usefulness cAM	3	3.75	4	4	4	0.25
Activity-centric concept	1	3.25	4	4.25	5	1
Sharing Activity Workspace	1	3.25	4	4	4	0,75
Activity-centric collaboration	3	3.75	4	4	4	0.25
Activity Cloud support	3	3.75	4	4	4	0.25

Table 1. The result of the 5-point likert survey that was used as a basis for the interview. The table shows an overview of the minimum, maximum, median ($\tilde{x}$) and the inter quartile range (iqr).

Activity Workspace

When asked about the design of co-Activity Manager, participants reported that they liked the overall design of the system as it helped them focus better on the active working context as well as find the information inside this working context faster ($\tilde{x}$= 4; *iqr*= 0.25). Participants also appreciated the general notion of *activities* ($\tilde{x}$= 4; *iqr*= 1). However, in strong contrast to his colleagues one participant (P3) did not like the concept of activity at all as he argued that it would increase complexity rather than decrease it. Surprisingly, during our observations he in fact did use activities (to show or hide a remote desktop connection).

The graphic designer (P4) also liked the idea of activities but did not like the implementation because it did not match with his Mac OS X workspace. P1 felt that activities were very useful as they allowed him to structure his workflow based on the parallel ongoing projects. He also felt that using activities helped him focus better on his work because he felt less distracted. P2 used activities in a similar manner as P1 as she created an activity for each project. The historian (P5) used activities in a rather unexpected way. At the beginning of each workday, she created a set of activities which mapped directly to tasks she was planning to do that day. During the day she would work through the list of activities one by one, keeping the finished ones as a reference. Although we did not anticipate participants would use activities for managing a to-do list, this approach worked very well for her.

Activity Sharing

Activity sharing ($\tilde{x}$= 4; *iqr*= 0.75) did not occur frequently but was rather used as an initiation process for a new long term collaborative project. During the observation, we noticed that when P1 received information (documents, images and other resources) for a new project that also involved another colleague (P2), he created a new activity that contained all this information and shared it with P2, who then accepted the activity and deployed it on her own machine as a local activity workspace. They both liked that instead of having to copy all the documents and resources manually, they could simply *share an entire context* which is automatically deployed on a new desktop. We also noticed that after receiving and deploying the activity, P1 and P2 would both customize (i) the activity itself as well as (ii) the activity workspace to their specific role. During the interview, P2 explained:

> *"It is much easier to just receive an entire activity and then customize it, than to find and collect all information separately".*

Both users reorganized the activity workspace based on their own preferences but more importantly deleted files, folders and contacts that were not relevant for their specific part in the activity.

Activity-Centric Collaboration

The integrated collaboration features, such as folder sharing and per-activity contact lists were used much more than the activity sharing mechanism. Most participants argued that the per-activity communication filter was very useful ($\tilde{x}$= 4; *iqr*= 0.25) as a tool to channel communication streams into the active working context. It was especially considered important for people that do a lot of multitasking or that are working at several active projects on the same time. The historian (P5) found this way of working very valuable as she argued that now all relevant information as well as contacts were visible and managed inside one desktop workspace. During the deployment it also quickly became clear that the integrated instant messenger would not be used very much. Because all participants in this study were physically collocated there was no active need to have a chat communication channel with all participants.

P1, e.g., used the per-activity contact list as a starting point for file and activity sharing but confirmed our observation that the IM functionality was a *second communication back channel* since most of the discussions were done face to face in the room. He also requested to add functionality that would allow certain contacts to be automatically added to all activities. In his case, he wanted to be available to his wife at all times regardless of what working context he was in. Finally, the historian (P5) also saw the potential in per-activity IM functionality, but she argued that the main advantage of this would be in the case when external collaborators that were not physically collocated would also use cAM.

During the study, a number of privacy issues arose from the mixture of private and work-related communication. As co-Activity Manager integrates with the Google Mail infrastructure, the activity-centric presence of all users was automatically distributed to all contacts in their Google Talk contact list. This side-effect of the chosen technology raised some serious privacy, reliability and confidentiality issues. A friend of P2, e.g., confronted her with a still undisclosed name of a project she was working on. Because she named one of her activities according to this new project, this name was distributed to all Google Talk contacts.

Automatically distributing people's current work activity was perceived to be very useful but also intrusive. In general people liked having an overview of who was working on what project since this knowledge would spark ad hoc meetings or collaborative work. However, it also had some disadvantages. P3, e.g., renamed one of his activities so that it would seem as work-related, even though in fact the activity only contained personal content such as a chat window to his wife and his music player:

> *"I didn't like that other people could see I was in an activity that was not strictly related to my work. In the end I renamed the activity, but you should have the ability to determine for each activity if you want to share the content".*

He felt that this information was irrelevant for his colleagues and he therefore did not want his colleagues to be informed of this. Finally, the interviews exposed that the per-activity contact list would, at times, cause inconvenient situations. Participants sometimes switched activities during a conversation which resulted in muting the other contact (if they did not happen to be part of that other activity). The muted contact would then have no way of leaving a message or response. P2 argued that asynchronous messaging should therefore be integrated into the system (as also proposed earlier by Voida et al. [35]), but P1 disagreed with this as he argued that email could be easily launched as part of the ongoing activity.

Activity Cloud

All participants in our study were actively using two to four devices so most of them valued the idea ($\tilde{x}$= 4; *iqr*= 0.25) of saving their activities in a cloud store as it would enable them to load and save activities on different devices that was equipped with a co-Activity Manager. P1 liked cloud support not only because it could be used to distribute activities over multiple devices, but also because the cloud storage mechanism allowed him to backup contextually meaningful structures rather than just a set of files.

DISCUSSION: LESSONS LEARNED

The central focus of co-Activity Manager is to provide an integrated activity-centric solution for the (i) configuration, (ii) articulation and (iii) coordination problem that occurs in modern knowledge work. In this section, we discuss the lessons learned from the implementation and deployment of the study as well as future directions.

Our study demonstrates that a relatively lightweight and open activity workspace allows for flexible activity management in different work practices. All participants in the study used activities in different ways, e.g., to organize projects, as a to-do list or simply as an extra desktop. And both the *duration* as well as the *scope* of the activities greatly differed between users.

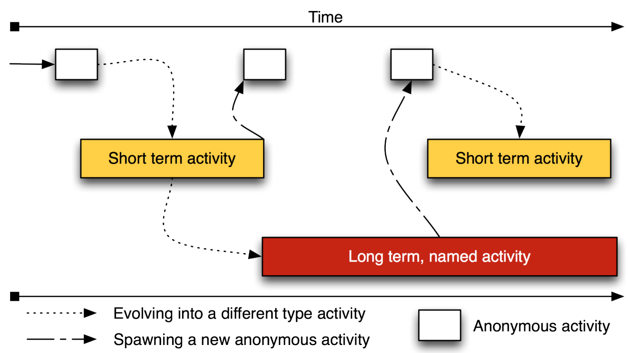


Figure 4. Activity Lifecycle – activities always start as empty anonymous ad hoc activities before disappearing or evolving into short or long-term defined activities. From within these defined activities, new ad hoc activities would emerge, that in turn would disappear or evolve.

Activities created and used during the study, can be categorized as either (i) long-term defined activities, (ii) short-term defined activities and (iii) ad hoc anonymous activities. The long term activities were given a proper name, icon and description, and were mostly used for long term projects or permanent activities (such as personal activities containing a music player and open email client). The short-term activities were also marked with an icon and named but had a shorter life span. The to-do approach of one participant is an example of using short-term activities, which were created in the morning and deleted at the end of the day. The difference between the different types of activities is not necessary the duration but rather the intentionality of the activity; the reason why the activity was created and used. Finally, most users also used ad hoc anonymous activities. Because creating a new activity is easy (one button press and no a priori configuration requirement), it was used for basic operations that required a clean desk or separate working context (e.g. writing a quick email or copying files from one folder to another).

The typical lifecycle of an activity is illustrated in Figure 4. Each activity starts as an empty ad hoc anonymous activity but evolves over time to a short or long-term activity with a proper name, definition and visualization depending on the content or the ongoing work context. From within emerging activities new ad hoc activities would be spawned that either died very quickly or evolved into another short or long-term defined activity. By allowing users to store the configured activity into a cloud store, they become usable beyond the individual device.

A key goal of co-Activity Manager was to include communication and articulation channels into the activity abstraction. By providing a collaboration manager that is customizable for each activity, we provide a mechanism for users to tunnel communication into the appropriate working context. In our implementation, we focus specifically on frequently used collaboration and communication tools.

Activity-centric collaboration emerged in two distinct phases: (i) a *sharing* phase in which a short- or long term activity would be prepared, shared and deployed by collaborations in order to start the (ii) *coordination* phase in which other users would use the per-activity collaboration manager to actually consume the collaboration inside a workspace. This process allowed users to use the configurations of their collaborator as a starting point for the configuration process of their own activity as part of the collaboration. The effect of this is that users can essentially template a work context and share it as a joint starting point in a collaborative setup for a team. Participants themselves proposed the notion of *workflow* to define this process and argued that this mechanism could even be used in a more restricted way.

As new activities would emerge for a local user, new collaborations would be spawned from within existing activities by using the collaboration manager. This per-activity collaboration manager, however, had several other roles. First of all, it was used as a default starting point for all types of communication and collaboration. Despite the fact that our prototype had technical limitations (e.g. we only support one IM protocol), it was used as intended and participants had many suggestions for improvements. During interviews, however, it became clear that for users to accept this type of system, it needs to *integrate* with the tools (protocols) they know and use now. Second, the manager also functioned as an awareness tool as it distributed the working context of all collaborators, thereby sparking face-to-face discussions and collaborations.

During our study, a number of *privacy* and *confidentiality* problems arose because of the automatic distribution of information of the active work context. Some of these problems were related to the technological implementation but further investigation exposed a more complex problem at the intersection of organizational policies and personal preferences. The team that was part of our user study was allowed by its employer to use email and instant messaging for both work and personal purposes. This, of course, greatly complicates the process of distributing information as private and public space is mingled into one interface. Potential solutions to this could include a higher level of control (e.g. Access Control Lists) over the distribution process (which would greatly complicate the process) or organizational policies that define the scope of the distribution on an infrastructural or protocol level.

CONCLUSION

In this paper we introduced co-Activity Manager (cAM), an activity-centric extension of the Window 7 interface that aims to deal with the problems and limitations of the desktop interface in context of collaborative knowledge work. cAM includes (i) an activity-centric workspace to minimize the configuration load, (ii) an activity sharing mechanism that can be used to distribute collaborative activities, and (iii) a per-activity collaboration manager that helps users tunnel communication channel and setup collaborations. We reported on a 14 days deployment in a multi-disciplinary software development team. Our study showed that the activity workspace is flexible enough to accommodate different individual and collaborative work practices and that activity-centric collaboration is a two-phase process consisting of an activity sharing and per-activity coordination phase.

ACKNOWLEDGEMENTS

This work was supported by the EU Marie Curie Network iCareNet under grant number 264738.

REFERENCES

1. E. Adar, D. Karger, and L. A. Stein. Haystack: per-user information environments. In *Proc. of CIKM '99*, pages 413–422. ACM.
2. A. Agarawala and R. Balakrishnan. Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In *Proc. of CHI '06*, pages 1283–1292. ACM.
3. B. P. Bailey, J. A. Konstan, and J. V. Carlis. The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In *Proc. of INTERACT '01*, pages 593–601. IOS Press.
4. J. Bardram, J. Bunde-Pedersen, and M. Soegaard. Support for activity-based computing in a personal computing operating system. In *Proc. of CHI '06*, pages 211–220. ACM.
5. J. E. Bardram. Activity-based computing for medical work in hospitals. *ACM Trans. Comput.-Hum. Interact.*, 16:10:1–10:36, June 2009.
6. V. Bellotti, N. Ducheneaut, M. Howard, and I. Smith. Taking email to task: the design and evaluation of a task management centered email tool. In *Proc. of CHI '03*, pages 345–352. ACM.
7. O. Bergman, R. Beyth-Marom, and R. Nachmias. The project fragmentation problem in personal information management. In *Proc. of CHI '06*, pages 271–274. ACM.
8. A. B. Brush, B. R. Meyers, D. S. Tan, and M. Czerwinski. Understanding memory triggers for task tracking. In *Proc. of CHI '07*, pages 947–950. ACM.
9. A. F. Cameron and J. Webster. Unintended consequences of emerging communication technologies: Instant Messaging in the workplace. *Computers in Human Behavior*, 21(1):85–103, Jan. 2005.
10. E. B. Cutrell, M. Czerwinski, and E. Horvitz. Effects of instant messaging interruptions on computing tasks. In *In CHI '00 EA*, pages 99–100. ACM.
11. M. Czerwinski, E. Horvitz, and S. Wilhite. A diary study of task switching and interruptions. In *Proc. of CHI '04*, pages 175–182. ACM.
12. D. Dearman and J. S. Pierce. It's on my other computer!: computing with multiple devices. In *Proc. on Human factors '08*, CHI '08, pages 767–776. ACM.
13. A. N. Dragunov, T. G. Dietterich, K. Johnsrude, M. McLaughlin, L. Li, and J. L. Herlocker. Tasktracer: a desktop environment to support multi-tasking knowledge workers. In *Proc. of IUI '05*, pages 75–82. ACM.
14. Y. Engestrom. *Learning by expanding: an activity-theoretical approach to developmental research*. Orienta-Konsultit Oy., 1987.
15. S. Fertig, E. Freeman, and D. Gelernter. Lifestreams: an alternative to the desktop metaphor. In *Proc. of CHI '96*, pages 410–411.
16. J. Fogarty and S. E. Hudson. Toolkit support for developing and deploying sensor-based statistical models of human situations. In *Proc. of CHI '07*, pages 135–144.
17. D. A. Henderson, Jr. and S. Card. Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Trans. Graph.*, 5(3):211–243, 1986.
18. E. Horvitz and J. Apacible. Learning and reasoning about interruption. In *Proc. of ECMI '03*, pages 20–27. ACM.
19. S. Houben, J. Vermeulen, K. Luyten, and K. Coninx. Co-activity manager: integrating activity-based collaboration into the desktop interface. In *Proc. AVI 2012*, pages 398–401. ACM.
20. D. R. Hutchings and J. Stasko. Quickspace: new operations for the desktop metaphor. In *Proc. of CHI '02*, pages 802–803. ACM.
21. J. Jin and L. A. Dabbish. Self-interruption on the computer: a typology of discretionary task interleaving. In *Proc. of CHI '09*, pages 1799–1808. ACM.
22. V. Kaptelinin. Umea: translating interaction histories into project contexts. In *Proc. of CHI '03*, pages 353–360. ACM.
23. V. Kaptelinin and M. Czerwinski, editors. *Beyond the Desktop Metaphor: Designing Integrated Digital Work Environments*, volume 1 of *MIT Press Books*. The MIT Press, 2007.
24. V. Kaptelinin and B. A. Nardi. *Acting with Technology: Activity Theory and Interaction Design (Acting with Technology)*. The MIT Press, 2006.
25. G. Mark, V. M. Gonzalez, and J. Harris. No task left behind?: examining the nature of fragmented work. In *Proc. of CHI '05*, pages 321–330. ACM.
26. T. P. Moran and S. Zhai. *Beyond the desktop in seven dimensions*, chapter 11, pages 335–354. The MIT Press, 2007.
27. M. J. Muller, W. Geyer, B. Brownholtz, E. Wilcox, and D. R. Millen. One-hundred days in an activity-centric collaboration environment based on shared objects. In *Proc. of CHI '04*, pages 375–382. ACM.
28. G. Oleksik, M. L. Wilson, C. Tashman, E. Mendes Rodrigues, G. Kazai, G. Smyth, N. Milic-Frayling, and R. Jones. Lightweight tagging expands information and activity management practices. In *Proc. of CHI '09*, pages 279–288. ACM.
29. T. Rattenbury and J. Canny. Caad: an automatic task support system. In *Proc. of CHI '07*, pages 687–696. ACM.
30. M. Ringel. When one isn't enough: an analysis of virtual desktop usage strategies and their implications for design. In *Proc. of CHI '03*, pages 762–763. ACM.
31. G. Robertson, M. van Dantzich, D. Robbins, M. Czerwinski, K. Hinckley, K. Risden, D. Thiel, and V. Gorokhovsky. The task gallery: a 3d window manager. In *Proc. of CHI '00*, pages 494–501. ACM.
32. P. Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. RFC 3921 (Proposed Standard), Oct. 2004. Obsoleted by RFC 6121.
33. G. Smith, P. Baudisch, G. Robertson, M. Czerwinski, B. Meyers, D. Robbins, and D. Andrews. Groupbar: The taskbar evolved. In *Proc. of OZCHI '03*, pages 34–43.
34. S. Voida and E. D. Mynatt. "it feels better than filing": Everyday work experiences in an activity-based computing system. In *Proc. of CHI 09*, pages 259–268. ACM Press.
35. S. Voida, E. D. Mynatt, and W. K. Edwards. Re-framing the desktop interface around the activities of knowledge work. In *Proc. of UIST '08*, pages 211–220. ACM.
36. F. R. Zijlstra, R. A. Roe, A. B. Leonora, and I. Krediet. Temporal factors in mental work: Effects of interrupted activities. Open Access publications from Maastricht University urn:nbn:nl:ui:27-18293, Maastricht University, 1999.

Paper 2: NooSphere

Title of paper

NooSphere: An Activity-Centric Infrastructure for Distributed Interaction

Authors:

Steven Houben, *Søren Nielsen, Morten Esbensen and Jakob E. Bardram*

Published:

In Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia ***(MUM '13)****. ACM, 2013.*

Abstract:

Distributed interaction is a computing paradigm in which the interaction with a computer system is distributed over multiple devices, users and locations. Designing and developing distributed interaction systems is intrinsically difficult as it requires the engineering of a stable infrastructure to support the actual system and user interface. As an approach to this re-engineering problem, we introduce NooSphere, an activity-centric infrastructure and programming framework that provides a set of fundamental distributed services that enables quick development and deployment of distributed interactive systems. In this paper, we describe the requirements, design and implementation of NooSphere and validate the infrastructure by implementing three canonical real deployable applications constructed on top of the NooSphere infrastructure.

NooSphere: An Activity-Centric Infrastructure for Distributed Interaction

Steven Houben, Søren Nielsen, Morten Esbensen, Jakob E. Bardram
The Pervasive Interaction Technology Laboratory
IT University of Copenhagen, Rued Langgaardsvej 7, DK-2300 Copenhagen,Denmark
{shou,snielsen,mortenq,bardram}@itu.dk

ABSTRACT

Distributed interaction is a computing paradigm in which the interaction with a computer system is distributed over multiple devices, users and locations. Designing and developing distributed interaction systems is intrinsically difficult as it requires the engineering of a stable infrastructure to support the actual system and user interface. As an approach to this re-engineering problem, we introduce NooSphere, an activity-centric infrastructure and programming framework that provides a set of fundamental distributed services that enables quick development and deployment of distributed interactive systems. In this paper, we describe the requirements, design and implementation of NooSphere and validate the infrastructure by implementing three canonical real deployable applications constructed on top of the NooSphere infrastructure.

Keywords

Distributed Interaction, Activity-Centric Computing, Distributed Computing, Activity Cloud, NooSphere

Categories and Subject Descriptors

H.5.m. [**Information Interfaces and Presentation**]: Miscellaneous

General Terms

Design, Human Factors

1. INTRODUCTION

With the widespread introduction of mobile devices (such as tablets and smartphones) and increased availability of large displays (such as situated displays and tabletops), setups in which users are engaged with multiple devices at the same time are becoming more common outside of the traditional smart space environment (such as Gaia [35] or iLand [37]). Heterogeneous multi-device environments have the potential to introduce new cross-device interaction techniques, support seamless shared information spaces based on the users' tasks or provide a new platform to explore collaborative setups. More general, *distributed interaction* refers to a paradigm in which interaction with a computer system is dynamically distributed over one or multiple (i) users, (ii) devices and (iii) locations. These three elements change over time and are in an ad-hoc and dynamic way related to the tasks or activities people are doing. Information is no longer tied to one specific personal device but multiple personal devices are rather mediators to the ubiquitous personal information space provided by the internet.

MUM '13, December 02 - 05 2013, Luleå, Sweden

http://dx.doi.org/10.1145/2541831.2541856

However, as pointed out by Edwards et al. in "*The infrastructure problem in HCI*", applications and user interfaces are not designed in isolation but on top of toolkits and infrastructures [15]. In fact, underlying technology determines to a great extend the capabilities and limitations of the interactive application. The problems with designing systems for distributed interaction lies in the fact that support for a large number of important distributed interconnected services needs to be in place. In an environment of changing users, devices and locations, these requirements pose a challenging and time consuming task to developers. First, there are a large number of problems related to the storage, sharing, replication, synchronization and contextualization of *data and information*. In order to provide seamless support for these, developers need to deal with complex dynamic network setups, synchronization between network protocols and concurrency issues. Second, there are a number of challenges related to the discovery and pairing of heterogeneous devices as well as the integration of distributed interaction systems in application-oriented platforms. Designing, prototyping and developing stable distributed interaction applications that are deployable in the wild is extremely challenging because it requires developers to engineer a stable infrastructure to support a number of distributed services that support the user interfaces. Moreover, the complexity of defining and managing a data and context model that provides the appropriate support for distributed interaction systems, is intrinsically tied to the architecture of the underlying infrastructure.

As an approach to this re-engineering problem and exploration into infrastructure design, we introduce *NooSphere*, an activity-centric infrastructure that provides a flexible platform for the prototyping of distributed interaction systems. We report of the architecture and components of NooSphere and describe three canonical applications built on top of NooSphere. The paper concludes with a reflection on the merits and limitations of our approach.

2. RELATED WORK

Smart space approaches have originally initiated the research into infrastructures and architectures that support distributed user interfaces (DUI). The seminal work of Marc Weiser [41] at Xerox PARC describing a ubiquitous vision that includes pads, tabs and large displays originated a vast body of research in interactive smart spaces. One of the earliest smart spaces is iLand [37] which used the roomware BEACH to create shared information spaces that stretch different displays and devices. Another classic smart space system, iRoom [22], provides the ability for pointer redirection, content replication and collaboration through an infrastructure- centric approach based on an event and data heap.

Project Aura [16] is a pervasive smart space system based on an adaptive infrastructure that supports surrogate clients that amplify the capabilities of mobile devices, nomadic file systems through data staging and network advisors. Aris [8] supports the redirection of windows and input between different types of devices including private devices and public displays in an effort to support a multi-user legacy user interface environment. Other approaches that provide support for pointer redirection are Pointright [23] and Impromptu [9]. *XICE* allows for the extension of input and output of mobile devices by *annexing* it to a smart space wall or table displays [1]. The Gaia Operating system [35] is a meta operating system that provides support for the coordination of software entities distributed over heterogeneous networked devices contained inside a physical space. Finally, *Shared Substances* [17] is a novel data oriented middleware that proposes to decouple functionality from data to support the design of multi-surface applications. A number of systems have been proposed to move beyond the smart room into smart buildings. ReticularSpaces [5], e.g., is built on top of a peer-to-peer event system that supports distributed hash maps to share data between different devices. More recently, the Window Brokers system [2] introduced an approach to annex devices into a shared workspace using display servers.

In literature, many existing approaches to pervasive middleware have been described over the years (including [6, 10, 12]). Several approaches focus specifically on collaboration inside a pervasive environment using platform compositions [33], context-awareness and semantic technologies [39], missions [24], proxy devices [38], services [34] and roles [18]. Ecora [32] is an agent-based pervasive framework for the construction of context-aware applications that focuses on heterogeneity, scalability, communication and usability. A similar infrastructure is GlobeCon [27] which provides support for distributed and pervasive computing in large-scale environments, thus moving beyond rooms or buildings. For a complete overview and classification of context-aware systems, we refer to [19] and [3]. Recently, a number of cloud platforms [25] including Google Apps Engine, Microsoft Azure and Amazon S3, have empowered developers to create applications in which data and services transcend the individual device and can be consumed on any device with internet access. This model adheres much more to the idea of a personal information cloud [28].

Most of these smart space systems and pervasive infrastructures however, suffer from three main issues. First, they are contained in one physical environment (room or building), neglecting some of the impact of mobility and interconnectivity. Second, most of these systems are extremely complex to deploy and do not integrate with existing applications and platforms, putting a great strain on developers and users. Finally, because these systems transcend individual devices, they introduce context or aggregation models to support end-users. However, many of these models are arbitrary context models and do not reflect the tasks or activities people do with these systems. NooSphere draws from this previous work to provide a lightweight infrastructure and programming framework that unifies the *interconnectivity* of cloud platforms with the *dynamics* of smart room technology. The core contribution of this paper is a novel generic and reusable infrastructure that represents *data and context* in an *activity-centric approach*, which reflects the actual tasks people do with these systems. Informed by prior studies on activity-centric computing approaches [5, 20], the infrastructure and all its services are thus designed specifically around this notion of *activity*.

3. REQUIREMENTS

Based on the related work discussed above and prior research into approaches for distributed environments, we have derived 7 core requirements for distributed interaction systems:

R1: **Persistence** – the infrastructure should provide a persistence mechanism that can be used to store data and information from any location, device type, or context. This will allow data and information to transcend the local space and context and become a truly ubiquitous concept based on an extensive life cycle re-use as described by Moran and Zhai [28]. Additionally, to support persistence during offline sessions, the infrastructure should cache data and events locally.

R2: **Distribution** – the infrastructure should support *synchronous* and *asynchronous* distribution of data models and files both in a local space as well as outside of this space. This allows for the connection of different distributed systems from different domains into one shared distributed interaction space.

R3: **Discovery and Pairing** – to allow applications and devices to seamlessly join a distributed space, the system must provide built-in support for (i) discovery of services, and (ii) an automatic pairing system that annexes different devices or applications into one seamless space. Since computer use is shifting from device specific applications and files to cross-device information and services, support for aggregation and composition of multiple devices should be an inherent part of the infrastructure.

R4: **Coordination and Communication** – as information and data is increasingly being used at the same time by multiple users, the infrastructure should support the notion of *user multiplicity* by allowing the attachment of specific artifacts, events, workflows or messages to the underlying data model. This ensures that applications or devices can support multi-user coordination and communication tools.

R5: **Configuration** – configuration refers to the process of determining the state of an application or device. To seamlessly move information between different applications or devices, the infrastructure should support the configuration of information on one or multiple devices for one or multiple users. This will allow users to manage data and services on different devices or applications.

R6: **Context Handling** – to support the use of context-aware functionality (e.g. location trackers) or embedded systems (such as Gadgeteer or Arduino), the infrastructure should provide a mechanism for system specific context processors that allow for the distribution of context information over all connected applications and devices.

R7: **Interoperability** – to fully support a multi-device configuration, the infrastructure should provide support for different operating systems and platforms by implementing a platform independent protocol. This will allow future inclusion of new technologies or platforms.

In order to support quick prototyping, development and deployment of distributed interaction systems, the requirements should be implemented with an appropriate level of abstraction, leaving e.g. networking and file management transparent to the developer. At the same time, the infrastructure should be extensible to meet new requirements. NooSphere is a lightweight and scalable toolkit that implements these requirements and can be used to design, prototype and develop interconnected activity systems for distributed interaction.

4. NOOSPHERE

NooSphere is an activity-centric service-based infrastructure and programming framework to support the development and deployment of distributed interactive systems. It is built on the concept of *communicating activity systems* [21]: by using a standardized data model and a two layered infrastructure consisting of a cloud and local distributed system, it allows for the deployment of different distributed interaction applications that can be interconnected. This implies that data and services are not confined within one system but can be consumed in all interconnected systems through adaptation of the context. This allows developers to built very complex distributed applications that consist of different domain specific system that are interconnect through the cloud. As illustrated in Figure 1, NooSphere is therefore composed of two fundamental components: (i) NooCloud, a cloud platform, and (ii) NooSystem, a distributed activity system.

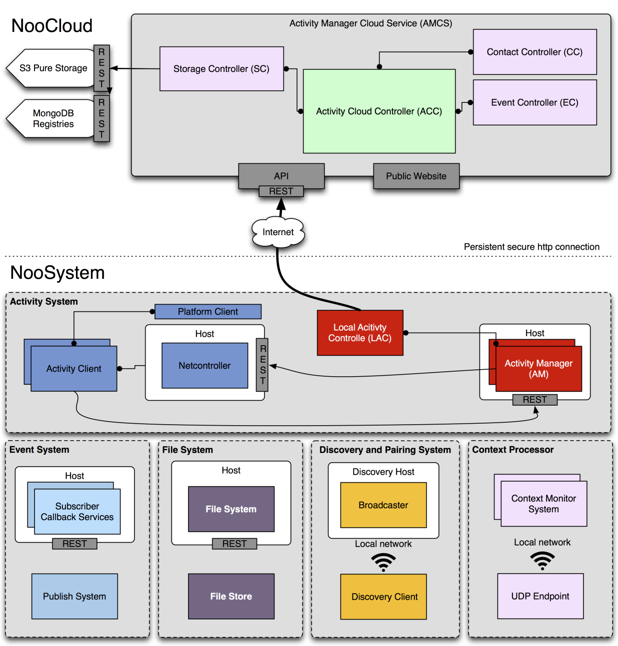


Figure 1: The architecture of NooSphere is composed of two components: (i) NooCloud, a cloud infrastructure that supports data storage, event distribution and activity management, and (ii) NooSystem, a dynamic distributed system that supports cross-device activity management, file and event distribution as well as a discovery and pairing mechanism.

4.1 Activity-Centric Computing

Activity-centric computing (a concept that was originally introduced by Apple Research [30]) is an interaction paradigm that provides support for the users' activities, rather than the tools they use to perform those activities. An activity is a higher level structure that encapsulates all resources and tools relevant to a specific task in order to represent an intention of work. Over the years several approaches to activity-centric computing have been successfully deployed to support (i) desktop multitasking, (ii) context handling and (iii) augmented interaction. The paradigm has been applied to different areas in Human-Computer Interaction (HCI), ranging from task management to collaborative work on the desktop interface [4, 13, 20, 29, 40]. However, more recently the approach has also been applied to distributed user interfaces and pervasive computing [5, 7, 11, 26] demonstrating its merit as a context model in a multi-device environment. Based on the experiences and lessons learned from successfully deploying these activity-centric systems, we propose to move to a more generalized activity-centric infrastructure, which will allow for more advanced prototyping and deployment.

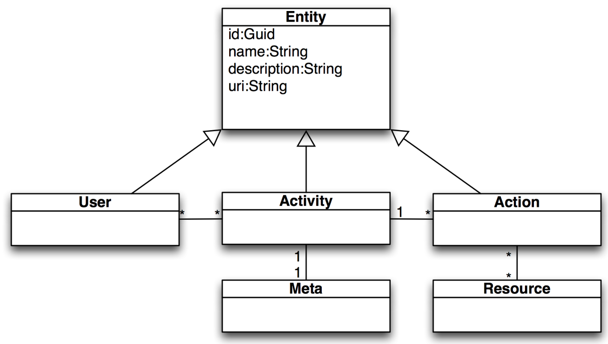


Figure 2: NooSphere uses activity as a first class object in all operations. The minimal activity object is composed of (i) users, (ii) meta information and (iii) actions, which are subtasks that contain resources such as files or links to web services..

4.2 Activity Model

The main advantage of activity-centric computing is that the activity model is a reflection of real physical tasks and activities that people do. Compared to other arbitrary context models (like those discussed in the related work section), the close mapping between the intention of the user's interaction and the digital representation of that intention allows users to easily use, appropriate and configure the model thereby minimizing intelligibility problems [5, 20]. The activity model includes *information* (such as files and resources), *coordination information* (such as users and roles) and *configuration states* (such as application descriptions) making it a well suited match for a multi-user, multi-device and multi-location environment. The first class object in the NooSphere infrastructure is thus an *activity*, as illustrated in Figure 2. In its minimal configuration, an activity is composed of (i) users, (ii) actions which contains resources and (iii) meta data.

Users. Users are digital representations of human agents that interact with the activity. They are part of the activity, they own, shape, define, consume, and share activities by interacting with the system. While the activity has one owner (the creator of the activity), it can be accessed by multiple users based on roles or other limitations imposed on the user object. All changes to the activity are shared with all users associated with the activity. The *user multiplicity* is an inherent part of the infrastructure and can thus be used to determine how actions and resources of an activity should be shared or consumed by the application.

Actions. Each activity is subdivided into a set of actions, which are tasks that are part of the activity. Actions structure how users interact with the different resources, such as files, folders and web services. Additionally, actions can be modelled as workflows, which are structured or unstructured sequences that are imposed or defined by the user. Actions thus describe *functions of work* as well as resources that are part of the activity.

Meta Data. Each event that occurs within the activity is logged and stored into the activity itself for persistence and reflection. The history can be used to track changes in parts the activity, create awareness on different actions or simply to visualize the development. Each activity is uniquely defined by an identity which consists of meta data such as a name, image or description and a unique reference number (e.g. GUID). An activity can be connected to other activities creating hierarchical relationships or references between activities.

4.3 NooCloud

NooCloud is an event-driven cloud-based service platform that supports (i) persistence of the activity model, (ii) storage capabilities for files, events and activity models, and (iii) cloud-based event distribution. The main component of NooCloud is the activity cloud controller (ACC), a cloud infrastructure exposed by a RESTful HTTP API. The ACC is the entry point of the framework and provides a number of activity-centric services grouped into a set of controllers, based on the Front Controller design pattern. Each controller is responsible for every create, read, update and delete (CRUD) operation performed upon its certain area of responsibility and related tasks are thus forwarded by the ACC to the correct controller. All controllers are aggregated in the ACC and exposed as an activity manager cloud service (AMCS). This service accepts HTTP requests for all functionality, ranging from file to activity management and is thus the public access point. Through the API, users can create, read, update and delete activity models and containing resources through pure HTTP requests. On top of this service runs a public website, which has two main purposes: (i) providing documentation on the API and (ii) allowing users to create an account or manage existing accounts.

4.3.1 Storage Controller

The storage controller (SC) provides support for the storage of all the infrastructure primitives, which are entire activity models, actions, users, devices and events. The SC supports two types of storage: (i) registries, which are entries that are stored in a NoSQL database and (ii) pure storage, which is binary serialized data. Registries store basic information for quick lookup and support advanced search and retrieval. The registry supports retrieval of entire activity models, actions, devices and users based on the GUID described in the entity base class. Because the activity model in NooCloud needs to be very flexible, a pure storage is implemented that stores the serialized extensions of the data model objects as well as actual files in the object cloud storage. This means that the activity itself and all its resources are saved directly in the cloud storage while keeping a reference in the lookup registry. *(Implementing R1: persistence)*

4.3.2 Event Controller

A local activity system can connect to the NooCloud through an HTTP endpoint that is exposed by the AMCS and handled by the event controller (EC). When a user or local system connects to the AMCS, it passes the login attempt to the EC, which establishes a persistent HTTP connection and returns a connection ID if the account is found. This ID is then used by the local system as a basic authentication token and is passed on as a parameter in each of the following HTTP requests. Users can connect several devices or local systems to the same cloud account causing the EC to register the connection to multiple devices or systems. Whenever a controller (e.g. storage controller) has handled a request that either changes data, or is relevant for other users or devices, the EC is asked to notify the relevant connected devices. The event is pushed to local users or systems over the persistent HTTP connection. The EC uses a centralized key-value store to store events before forwarding them. Through the EC, the infrastructure supports synchronous updates and work on multiple devices involving multiple users. *(Implementing R2: distribution)*

4.3.3 Contact Controller

Through the AMCS, a contact controller (CC) takes care of the handling of user management. A user can prompt another user to become a contact, and a contact request is thereafter stored in the registry as well as send to the prompted user. They can then choose to accept the request, after which the two users are connected and automatically subscribed to each others activities upon connect. A contact can be removed at any time, disabling the notifications from the other user. *(Implementing R4: communication and coordination)*

4.4 NooSystem

NooSystem is a dynamic service-based infrastructure that supports the distribution of activity model instantiations, files, communication and coordination messages and context representations in a local multi-device information space. The infrastructure uses a flexible service model that during compile time can be accessed through a DLL or class files, while each service at runtime is accessible through a REST HTTP service that is hosted in its own url-based service host. The infrastructure is composed of five distributed subsystems: (i) the activity system, (ii) publish-subscribe event system, (iii) a file system, (iv) a discovery and pairing system and (v) a context monitoring system.

4.4.1 Activity System

The activity system is subdivided into two components: activity manager and activity client. The activity manager (AM) is used as a coordinator between different local clients and/or managers that are connected inside the local NooSystem and is directly connected to one account of the activity cloud controller (ACC). The activity client on the other hand, is used to consume the activities on local devices and is directly connected to a local AM, which distributes the activities to the clients.

The activity manager (AM) is typically connected to an instance of NooCloud and thus is synchronized with all activities stored in the cloud for a particular user account. The AM is connected to the ACC through a Local Activity Controller (LAC), which creates and maintains a persistent HTTP connected with NooCloud. All events that are distributed by the ACC are handled by the LAC and passed on to the AM, which distributes the events through a local distributed publish / subscribe system to all connected clients. The AM caches all activities locally to deal with network interruptions and to speed up the distribution of activities and resources. The AM is exposed to the local NooSystem through RESTful http services. The AM can also be used only in a local setting (without the cloud link) and can even be connected to the activity store of another AM. The activity client (AC) is composed of a netcontroller and a platform client. Each AC is connected to a local activity manager (AM) through the netcontroller which registers itself with that AM with a callback service address that runs a client REST HTTP service to which the AM can send distributed events. The platform client (PC) converts the netevents into native events (e.g. C# delegates) which are then exposed to the platform integration layer. Both the activity client (AC) and activity manager (AM) are composed of four local distributed subsystems: (i) event system, (ii) file system, (iii) discovery system and (iv) context monitor system. *(Implementing R5: configuration)*

4.4.2 Event System
The distributed event system (ES) is an HTTP REST publish/subscribe service that supports the distribution of messages, activity events, device events, user events, file events and external events. When an activity client is connected to an activity manager, the client sends an HTTP request to the manager that contains a device object, describing the device, and a callback service address on which the activity client device is running the callback services. The activity manager registers the activity client and active device and publishes all events and messages to the host address of the activity client. These events include system messages which contain either user content (such as chat messages) or control messages (such as reconnect requests). The event system also distributes changes to the activity collection (such as added, removed, changed or locked) to inform all connected clients to update the local visualization. Next, every time a device containing an activity client connects to or disconnects from the activity manager, device messages containing information about changes in the device collection are sent to all connected clients. *(Implementing R2: distribution)*

4.4.3 File System
Both the activity client and activity manager are equipped with a file system. The file system is composed of a file server, which is responsible for saving (to disk) and loading files (to stream), and a file store, which is a key-value store that registers all files that are part of the loaded activities (NoSQL database). The file service distributes file events through the event service (request to download, request to upload and deleted), when files are changed at the activity manager. The source can be both local (in the NooSystem) but also external (by an external NooSystem that is connected through the NooCloud). When a new file is added to an activity by an activity client, the activity model is first updated to the activity manager, which will in turn send a *request to upload* message to the client. When the client uploads the file to the file store of the manager, the activity manager sends an *request to download* to all other connected activity clients as well as the NooCloud. The NooCloud in its turn will request the local activity manager to upload the file to the cloud storage. Because all updates are sent to all attached activity clients and managers, local applications using the infrastructure can decide how to handle file consistency and potential conflicts. *(Implementing R1: persistence and R2: distribution)*

4.4.4 Discovery and Pairing System
Each activity manager is equipped with a broadcast service, which broadcasts the manager's name, host address and device information (type, physical location and ID) over the local network. When the broadcast service is started, a separate host with a dedicated address is launched. The broadcast service can be dynamically (re-)configured and (de-)activated at runtime, to allow developers to toggle discovery support. In order to find activity managers on a local network, both the activity client as well as activity manager are equipped with a discovery service. This service searches the local network for available activity managers and exposes them to the main controller for consumption. The current version of NooSystem implements both Webservice Discovery as well as Apple Bonjour protocols to support different types of devices and operating systems. *(Implementing R3: discovery and pairing)*

4.4.5 Context Monitor System
To add support for context-aware and embedded devices (such as Arduino or location trackers), the activity manager and activity client are equipped with a context processor. This processor tracks a collection of *IContextService* objects which are monitored in separate threads. Each event triggered by the context processor is distributed through either the event system (as a context message) or via a UDP multicast system (for real-time services) that is dynamically launched and attached to the context processor. *(Implementing R6: context handling)*

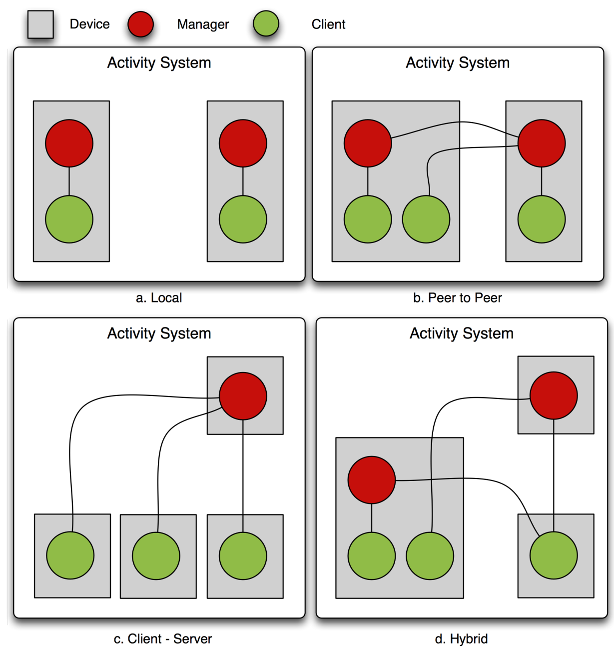


Figure 3: The local infrastructure NooSystem, can be deployed in different configurations: (a) manager and client on the same devices, (b) peer to peer connection between manager on different devices, (c) traditional client server approach with a manager on a dedicated devices or (d) hybrid setup composed of both dedicated and local managers.

4.5 Deployment
The infrastructure is designed to be modular and scalable and can thus be used in combination (NooCloud and NooSystem) but also separate. Since NooCloud provides an open API, any developer can design a custom activity system using any language or platform that supports REST HTTP calls. NooSystem on the other hand has the ability to work with local activities only, thereby removing the necessity of connecting to the cloud part to use activities. *(Implementing R7: interoperability)*

The NooSystem is also modular on a local level. Since both the client and manager are lightweight services, they can be spawned on either the same or different devices (Figure 3 a) using HTTP services. Activity managers can also connect to each other, emulating a peer to peer system (Figure 3 b). With this approach, different managers can exchange activities of different users. In case one device is dedicated for the activity manager, the setup can also be configured as client-server (Figure 3 c). Because of the service-based approach, hybrid approaches that merge dedicated managers and local managers can be connected to form one NooSystem (Figure 3 d). Finally, the code library provides the ability to use the activity manager and client directly in code without the need for dynamic REST HTTP services. Because of the architecture of the system, the network code is completely transparent for developers, who simply need to connect to a running service from within their application. All network code is made transparent by wrappers that translate network events send by the NooSystem to local delegates.

The NooCloud infrastructure is built on top of the ASP.NET Web API Framework and runs on the AppHarbour cloud platform. The event system is implemented using SignalAR in order to support a cloud-based real-time publish-subscribe mechanism. The NoSQL database used by NooCloud to create the registries is MONGODB The cloud infrastructure is exposed to the web through a REST HTTP API. The Noo System is implemented using Mono WCF (Windows Communication Foundation), Web API and runs on Windows, Linux OS X and Android. Each service (activity, discovery and file) runs in a custom built service host (using Owin) which exposes the service through a REST HTTP service. The local storage is implemented using RavenDB.

5. CASE STUDIES

To validate the *functionality* of the infrastructure and test the *stability*, *performance* and *feasibility* of the architecture, we present three canonical distributed activity-centric case studies [14] that represent real deployable and testable system similar to those found and tested in research and industry. Table 1 provides an overview of the three applications and their use of the underlying features provided by NooSphere.

Building different reference applications on top of an infrastructure has been proposed as a robust research method for evaluating infrastructures [14, 15]. These applications demonstrate the functionality of the infrastructure and can be used as input for its applicability for supporting application development. In this section, we present three such case studies, and discuss how they benefit from using the NooSphere platform. In the case studies we present:

- **Case Study #1: co-Activity Manager** – reimplementation of an existing application [20].
- **Case Study #2: Dynamic Device Composition** – construction of an application using the basic infrastructure.
- **Case Study #3: SmartWard** – rapid prototyping of an advanced research application by leveraging the features of the infrastructure.

5.1 CS #1: co-Activity Manager

Description. Task- and activity-based desktop systems have been proposed as a mean to contextualize desktop usage as they re-organize information based on tasks people do. With the more widespread availability and usage of tablets and e-readers, significant research have been investigating how to seamlessly integrate the exchange of contextual files between different types of devices and users. In this application, we reimplemented co-Activity Manager [20] (cAM), an activity-centric desktop manager using the NooSphere infrastructure and extended its functionality so it seamlessly supports resource sharing with a tablet containing an e-reading application.

Figure 4: co-Activity Manager [20] (cAM) is a multi-user activity centric desktop interface that contains an activity bar at the top of the screen (A). Each activity button represents an ongoing activity; clicking the button will load the virtual desktop related to the activity as well as repopulate the desktop background. Files can be related to the activity by dragging and dropping them on the desktop background of a loaded activity. The tablet (B) can be paired to cAM by selecting the device from the auto-detect menu. Users can simply drag and drop files they want to use on a tablet on the activity bar located at the top of the screen.

Using the Infrastructure. Using NooSphere as underlying infrastructure, the entire network, activity management and communication code base from the original project (roughly 3000 lines of code) could be replaced by 50 lines of code required to initialize the infrastructure. The user interface and features are identical as in the original implementation. However, the reimplementation that uses the *activity client* and *activity manager* from NooSphere, simply needed to setup the activity system and hook the user interface components to the events produced by the infrastructure. Figure 5 shows the code used to set up the activity client using the NooSphere infrastructure. The code sample first demonstrates how to create and initialize a new user and device and associate them with the new activity client and activity manager. It also shows a number of example event available through the activity client object. The callbacks hooked to these events are part of the infrastructure and designed to de-serialize the JSON objects used by the infrastructure into native typed c# objects. After opening the activity client, an activity repository is created (or loaded in case a previous session is detected), a file server is started and the REST and discovery services are run automatically in the background. Because the infrastructure automatically includes support for sharing activities and their containing resources and contacts with other devices, we extended the system with a tablet application that can be used for active reading and resource browsing. The tablet application also implements an activity client, which connects automatically to the activity manager of cAM using the built-in discovery mechanism. All changes to resources are automatically synced by the

Feature	#1: Reimplementation	#2: Device Composition	#3: Extending the infrastructure
Persistence	Activities and files across devices in the cloud	Activities and files in the cloud	Local data store and file system
Distribution	Synchronisation of activities and resources	Sharing and synchronisation of files	Synchronisation of activity states
Discovery	Automatic detection of tablet through background discovery service	Ad hoc pairing triggered by vision systems	Automatic ad hoc pairing based on location /proximity
Coordination	Friend list, sharing activity models through cloud	Device control (master-slave)	Activity-centric messaging (nurses records)
Configuration	Structure desktop using activities that can be deployed on any device.	File management across multiple users and devices	Multi-user synchronized activity manager
Context	No	Pointer and touch redirect	Location tracker, Arduino support for RFID scanner
Interoperability	Abstracts activities into a shareable form (JSON)	Resources to encapsulate native files	Abstracts activities into a shareable form (JSON)

Table 1: An overview of the three canonical applications and how they each utilize different features provided by the NooSphere infrastructure.

```
private void StartClient(string activityManagerHttpAddress)
{
    var device =  new Device(){ DeviceRole = DeviceRole.Master,DeviceType = DeviceType.Desktop}
    var user = new User(){ Name = "Username", Email = "use@mail.com" }
    var client = new ActivityClient(@"c:/myFiles/", device) { CurrentUser = user };

    client.ActivityAdded += (object sender, ActivityEventArgs e)=>{
        AddActivityUi(e.Activity);
    };

    client.MessageReceived += ClientMessageReceived;
    client.FriendRequestReceived += ClientFriendRequestReceived;
    client.ConnectionEstablished += ClientConnectionEstablished;
    client.ServiceIsDown += ClientServiceIsDown;

    client.ContextMonitor.AddContextService(new InputRedirect(PointerRole.Controller));

    client.ContextMessageReceived += _client_ContextMessageReceived;

    client.Open(activityManagerHttpAddress);

    client.AddActivity(new Activity(){Name="default activity";}
}
```

Setup | Event Handler | Context | Use

Figure 5: The basic infrastructure programming framework provides developers with a number of objects that hide all the underlying complexity. This short code sample e.g. exposes all underlying network, synchronization, serialisation and context handling through basic events that send back a native usable object.

infrastructure and presented to the UI by native events. Some minor updates to the UI of cAM were required to deal with immediate updates from the tablet device (e.g. dispatch the received object to the UI thread), but no extra infrastructure code was required to add an additional device. Table 1 shows how the reimplementation of cAM uses almost all basic services except context handling, which was not a requirement for this particular project.

User Experience. The basic services provided by NooSphere allows for a seamless multi-device experience, in which users need to perform very little manual setup, or *configuration work*, in order to pair devices or share information with other devices. Because of the built-in discovery system, the user can simply pair devices by clicking a button. There is thus no need to setup shared folders, add credentials or install third party applications. Through the user interface of cAM, the user can simply drag and drop resources (such as files or contacts) on top of an activity button, which causes the infrastructure to automatically transfer it to the attached tablet that visualizes the newly added resources in context of the existing activities. The consistent use of activities as structuring mechanism thus provides users with a consistent mental model across devices.

5.2 CS #2: Dynamic Device Composition

Description. Mobile devices such as tablets and smart phones provide users with a high degree of mobility in accessing information such as emails, images and other resources. However, because of the limited size of their screens, these devices are not appropriate for accessing large quantities of resources. Because of this, a body of work (e.g. [36]) has explored the connection between small mobile devices and large situated horizontal displays like tabletops. In this case study, we demonstrate the ability to pair a mobile device (a tablet computer) to a tabletop display (Figure 6).

Figure 6: The public tabletop (A) connected to a tablet (B). The user walks up to an empty public interactive table (C) and simply places his device on top of the tabletop. The table will recognize the device and use the discovery to find and connect to the activity manager of the tablet. All resources related to the active activity on the mobile device are deployed on the interactive surface (D). When the user switches between activities, the interactive desk is repopulated with resources related to that activity. Users can utilize the desk to exchange, modify or manage files and resources. Additionally, the system allows the user to redirect input from the table, thereby providing remote control over the surface.

Using the Infrastructure. The main complexity and challenges in any application that supports multi-device composition on an interactive surface are (i) detecting and pairing with devices, (ii) file and resource synchronization and (iii) multi-user context (e.g. what resources belongs to what user). The interactive surface application was designed on top of the NooSphere *activity client* and *activity manager* (as seen in Table 1). However, in contrast to the co-Activity Manager application, the activity clients are dynamically started when a new device is detected. When the interactive surface detects a new device (using the vision system and static markers), it automatically launches the built-in discovery system to find a device with a running activity manager (in this case a tablet) of which the broadcast code matches the byte value of the detected tag. The broadcast code is taken from the device object that is initialized when the activity manager on the tablet is created. When the activity client on the surface computer pairs with the detected activity manager, all shared resource (images in this case) are automatically synchronized between both devices, and visualized on the surface. Because each loaded image is associated to a specific activity and its user, the system can distinguish the image set of each user. Additionally, rather than simply distributing files, the infrastructure provides a *Resources* object which encapsulates a file and annotates it with meta data. This data can be used by the application to do version control or check the association with multiple users and their activities. To support pointer redirect between different devices (allowing for remote control), the surface and tablet applications both implement a basic Context Service, which translates touch events from one device to another. The infrastructure adds the services to the running activity clients (e.g. as seen in the code sample in Figure 5) and automatically sets up and distributes the context information over a UDP multicast connection. Again, all complex multi-threaded network code, context modelling and device synchronization is hidden for the developer, allowing them to focus on the user interface and experiences. Because of all the supported services of the infrastructure, the total lines of code for the surface application is less than 800 lines.

User Experience. The synchronized activity state allows users to easily swap their set of resources on the interactive table. In a multi-user experience, this thus means that by simply selecting a different activity on the tablet, the user updates his part of the shared view. Because the infrastructure allows for easy addition of new resources, users can simply drag and drop resources from other users to their device. Again, very little *configuration work* is needed to exchange information or update the shared view. The pointer redirect can be enabled with a simple button click. Although some work on the side of the developer is required to support relative mapping, the master-slave negotiation and distribution of coordinates over the built-in UDP connection, creates an easy to use and transparent system for the end user. The user is thus given a very simple interface with advanced functionality hidden in NooSphere.

5.3 CS #3: SmartWard Research Prototyping

Description. In hospital patient wards, the whiteboard and patient record are two important artifacts to coordinate information concerning patients. In this case study, we demonstrate the first rapid prototype implementation of an ongoing research project in which we are constructing a distributed patient management system which supports multi-device configuration of patient cases as well as coordination through a number of automatic tracking, awareness and communication tools.

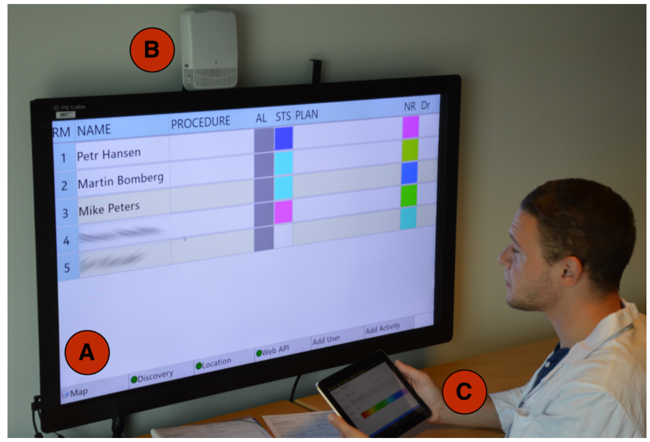


Figure 7: The SmartWard system consists of a large interactive whiteboard to display shared information on patients registered on the ward using RFID (A), a location tracker used to detect which patients are at the ward or in surgery (one node visible in B) and a tablet (C) used by doctors and nurses for more detailed information on the patient.

Using the Infrastructure. As illustrated in Table 1, this prototype uses all basic services provided by NooSphere. As in the other case studies, SmartWard uses *activity clients* and *activity managers* to synchronize activities, user information and resources across all devices. However, for this more domain specific application, we extended the infrastructure with a *WardNode* layer and specialized activity models (e.g. Patient, Nurses, Doctors,...). Figure 8 shows the code used by the application to (i) launch an activity system (manager or client) and (ii) and connect the distributed patient repository to a *ObservableCollection* that can be consumed by the UI. The WardNode is also connected to a Sonitor ultrasound location tracker (Figure 7 B) using NooSphere. The location tracker is a specialized context processor which runs in a separate service host managed by NooSphere. This means that the developer can simply enable the location tracking and use native events to deal with detections. Finally, the NooSphere event systems allows for the decoration of activities with custom tags and messages. In the SmartWard system, the patient records are attached to the custom patient model as activity-centric messages.

```
public BoardViewModel()
{
    WardNode = WardNode.StartWardNodeAsSystem(WebConfiguration.DefaultWebConfiguration);

    Patients = new ObservableCollection<PatientViewModel>();
    Patients.CollectionChanged += Patients_CollectionChanged;

    WardNode.PatientAdded += WardNode_PatientAdded;
    WardNode.PatientRemoved += WardNode_PatientRemoved;

    WardNode.PatientChanged += WardNode_PatientChanged;
    WardNode.Patients.ToList().ForEach(p =>
        Patients.Add(new PatientViewModel(p) {RoomNumber = _roomNumber++}));
}
```

Figure 8: The Wardnode class is a thin infrastructure extension which transforms NooSphere into a domain specific deployment. The code allows developers to create a distributed synchronized patient repository, which can be easily consumed by a MVVM application.

User Experience. To support coordination between clinicians, the infrastructure provides a patient activity for each active patient at the ward. This patient activity model is used to keep a strict synchronized whiteboard view but can also be used to share information with other clinicians. Since the information exchange services (such as writing nurse logs or updating the color state of a patient) are coupled directly to the patient activity model, there is no additional *configuration work* in locating the relevant contacts or starting an additional tool. Simply attaching information to the patient case will automatically distribute it to all relevant clinicians. The built-in support for location tracking provides clinicians with an easy to use and transparent search tool for other clinicians or artefacts (such as e.g. the patient record) at the ward.

6. DISCUSSION

Distributed interaction is a concept that has been around for many years, yet very few *infrastructures*, *toolkits* or *programming frameworks* that support the prototyping, development and deployment of these types of systems are actively in use. The central goal of NooSphere is to introduce a new *intermediate* [15] activity-centric infrastructure and programming framework that is aimed at providing a set of fundamental services required to design and deploy distributed interaction systems.

NooSphere uses activity as a *first class object* in an effort to reflect the intention of users in the modelling of information spaces that are spread over multiple devices, multiple locations and multiple people. Compared to traditional smart space and pervasive computing systems, this data model maps to the real physical tasks and activities people do in the information spaces provided by the infrastructure. This close match between the users' psychological interpretation of work and the digital aggregation of the resources required to perform this work, provides users with a stable mental model that has the capability to transcend the individual device. The model supports the notion of *actions* to structure work and resources and *user multiplicity* to allow multi-user access to the same activity model. Because of this activity model, the infrastructure allows for the creation of activity-centric coordination, configuration and communication tools that are part of the same *activity system*.

A central contribution of NooSphere is the aggregation of a cloud infrastructure, that is used for persistence and distribution of events, and a local dynamic distributed roomware infrastructure (similarly to COAST [37]). However, one of the core differences to prior smartspace systems is that NooSphere does not require specialized equipment or user interface frameworks but is usable with existing operating systems and UI toolkits. NooSphere thus encapsulates a number of complex services and systems into one activity-centric infrastructure, which is exposed through an API or standard REST interface. The main purpose of this approach is to provide a truly distributed and persistent platform that provides the ability to interconnect systems distributed over different locations all over the world. Combining a dynamic smart room environment with the persistence of an integrated cloud platform opens up possibilities for new collaborative setups distributed over multiple locations. This simplification of interconnections between distributed services or "'*Power in combination*" [31] results in a new design and prototyping platform. By providing a standard architecture and model for activity-centric computing, we provide developers with a framework to built interconnectable tools.

Prototyping complex activity-centric distributed system is *easier* as a developer is provided with a set of basic services which are *flexible*, easy to set up and *transparent*. All network code, discovery mechanisms, file and activity synchronization, and context handling are abstracted into the infrastructure and presented to the developer as basic Mono `C#` objects and delegates. Because of this, prototyping and designing distributed user interfaces is significantly *faster* as it requires less lines of code (to debug). Because of the abstract model iterative changes to the design (e.g. induced by user-centric design) do not require the re-engineering of (parts of) the infrastructure. The architecture of the infrastructure is extensible as controllers and services can be added, allowing for modifications, extensions and integration with other platforms. The infrastructure is designed to support a broad range of technical setups ranging from traditional local client-server-cloud (e.g. Case Study #1) and peer to peer (e.g. Case Study #2) setups to large complex cloud-based hybrid setups (e.g. Case Study #3). Because of the two-layered architecture and component based design, the infrastructure is *scalable* and *reusable* for complex distributed applications.

The infrastructure currently also has a number of *challenges* and *limitations*. Some services, such as the context processor or discovery mechanism, provided in the local activity system are not usable in the cloud. Although the infrastructure allows for messaging between activity systems using the cloud event controller, this approach is practically not feasible for e.g. discovery or high bandwidth real-time context data. The current two-tier architecture of NooSphere is grounded in the design rational that any device that is part of the activity system is connected to a local network. This implies that a local device can always be setup as a local activity manager, thus providing a node with the necessary services. However, there are a number of use case (e.g. using smartphones on a 3G network) where these services can currently not be provided. E.g. if the tablet from Case Study #1 would be connected to the activity cloud over 3G, the local system would not be able to detect it. Although the device would be in the same room, the event distribution would be done over the cloud, not the local system.

Because of the high level of abstraction of the activity model and infrastructure design, some use cases require a thin infrastructure layer on top of the standard NooSphere API. E.g. in Case Study #3, a domain specific layer was constructed to encapsulate some of the dynamic ad-hoc node creation as well as an implementation of the location tracker. This thin layer is not a formal requirement as the same functionality can be achieved on the bare framework code. However, adding this thin layer can facilitate development and help to manage the complexity of more advanced setups. Although NooSphere provides developers with a number of `C#` objects and event and a REST API for other programming environments, there is currently still an *integration problem*. Because the infrastructure is primarily focused on data, event and context distribution, the development and integration of these concepts into the user interface is still left in the hands of the developers. Although NooSphere greatly reduces the amount of work on the distribution part, building activity-centric user interfaces (such as [4, 13, 20, 29, 40]) is still a challenging task. A next step could thus be to extend the API of the infrastructure to deeply integrate with existing operating systems and widely used systems and tools, to provide an even broader development platform, or *activity-based toolkit*.

7. CONCLUSION

In this paper we introduced NooSphere, an activity-centric service-based infrastructure for the prototyping of distributed interaction systems. We described the motivation, architecture and components, and presented three example applications build using NooSphere. We are currently using the infrastructure for different research projects aiming at deploying multi-device computing support in hospitals, interactive desks for knowledge workers, and distributed collaboration in global software development.

8. ACKNOWLEDGEMENTS

This work was supported by the EU Marie Curie Network iCareNet under grant number 264738. Thanks to Steven Jeuris for his suggestions and feedback on early drafts of this paper.

9. REFERENCES

[1] Arthur, R., and Olsen, Jr., D. R. Xice windowing toolkit: Seamless display annexation. *ACM Transactions on Computer-Human Interaction 18* (August 2011), 14:1–14:46.

[2] Arthur, R., and Olsen, Jr., D. R. Window brokers: Collaborative display space control. *ACM Transactions on Computer-Human Interaction 19*, 3 (Oct. 2012), 17:1–17:21.

[3] Baldauf, M., Dustdar, S., and Rosenberg, F. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing 2*, 4 (2007), 263–277.

[4] Bardram, J., Bunde-Pedersen, J., and Soegaard, M. Support for activity-based computing in a personal computing operating system. In *Proc. of CHI '06*, 211–220.

[5] Bardram, J., Gueddana, S., Houben, S., and Nielsen, S. Reticularspaces: Activity-based computing support for physically distributed and collaborative smart spaces. In *Proc. of CHI 2012*, ACM (2012).

[6] Bardram, J. E. The Java Context Awareness Framework (JCAF) - A Service Infrastructure and Programming Framework for Context-Aware Applications. *Pervasive Computing* (2005), 98–115.

[7] Bardram, J. E., Bunde-Pedersen, J., Doryab, A., and Sørensen, S. Clinical surfaces - activity-based computing for distributed multi-display environments in hospitals. In *Proc. of INTERACT '09*, 704–717.

[8] Biehl, J. T., and Bailey, B. P. Aris: an interface for application relocation in an interactive space. In *Proc. of GI 2004*, 107–116.

[9] Biehl, J. T., Baker, W. T., Bailey, B. P., Tan, D. S., Inkpen, K. M., and Czerwinski, M. Impromptu: a new interaction framework for supporting collaboration in multiple display environments and its field evaluation for co-located software development. In *Proc. of CHI '08*, 939–948.

[10] Black, J. P., Segmuller, W., Cohen, N., Leiba, B., Misra, A., Ebling, M. R., and Stern, E. Pervasive computing in health care: Smart spaces and enterprise information systems. In *Proc. of the MobiSys 2004*,

[11] Consolvo, S., McDonald, D. W., Toscos, T., Chen, M. Y., and Froehlich, J. e. a. Activity sensing in the wild: a field trial of ubifit garden. In *Proc. of CHI '08*, 1797–1806.

[12] Dey, A., Abowd, G., and Salber, D. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human–Computer Interaction 16*, 2-4 (2001), 97–166.

[13] Dragunov, A. N., Dietterich, T. G., Johnsrude, K., McLaughlin, M., Li, L., and Herlocker, J. L. Tasktracer: a desktop environment to support multi-tasking knowledge workers. In *Proc. of IUI '05*, 75–82.

[14] Edwards, W. K., Bellotti, V., Dey, A. K., and Newman, M. W. The challenges of user-centered design and evaluation for infrastructure. In *Proc. of CHI '03*, 297–304.

[15] Edwards, W. K., Newman, M. W., and Poole, E. S. The infrastructure problem in hci. In *Proc. of CHI '10*, 423–432.

[16] Garlan, D., Siewiorek, D. P., Smailagic, A., and Steenkiste, P. Project Aura: Toward Distraction-Free Pervasive Computing. *IEEE Pervasive Computing 1*, 2 (April 2002), 22–31.

[17] Gjerlufsen, T., Klokmose, C. N., Eagan, J., Pillias, C., and Beaudouin-Lafon, M. Shared substance: developing flexible multi-surface applications. In *Proc. of CHI '11*, 3383–3392.

[18] Haibin Zhu, M. Z. Role-based collaboration and its kernel mechanisms. In *Proc. of the 3rd IEEE International Workshop on Distributed Intelligent Systems* (2006).

[19] Hong, J.-y., Suh, E.-h., and Kim, S.-j. Context-aware systems: A literature review and classification. *Expert Syst. Appl. 36*, 4 (May 2009), 8509–8522.

[20] Houben, S., Bardram, J., Vermeulen, J., Luyten, K., and Coninx, K. Activity-centric support for ad hoc knowledge work - a case study of co-activity manager. In *Proc. of CHI '13*, 2263–2272.

[21] Houben, S., Esbensen, M., and Bardram, J. A situated model and architecture for distributed activity-based computing. In *Proc. of MODIQUITOUS '12*, 9–13.

[22] Johanson, B., Fox, A., and Winograd, T. The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE Pervasive Computing 1* (April 2002), 67–74.

[23] Johanson, B., Hutchins, G., Winograd, T., and Stone, M. Pointright: experience with flexible input redirection in interactive workspaces. In *Proc. of UIST 2002*, 227–234.

[24] Kumar, M., and Shirazi, B. A. Pico: A middleware framework for pervasive computing. *IEEE Pervasive Computing 2* (2003), 72–79.

[25] Lenk, A., Klems, M., Nimis, J., Tai, S., and Sandholm, T. What's inside the cloud? an architectural map of the cloud landscape. In *Proc. of the ICSE Workshop on Software Engineering Challenges of Cloud Computing 2009*, 23–31.

[26] Li, Y., and Landay, J. A. Activity-based prototyping of ubicomp applications for long-lived, everyday human activities. In *Proc of CHI '08*, 1303–1312.

[27] Lu, K., Nussbaum, D., and Sack, J.-R. Globecon: A scalable framework for context aware computing. In *Smart Sensing and Context*, G. Kortuem, J. Finney, R. Lea, and V. Sundramoorthy, Eds., Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007.

[28] Moran, T. P., and Zhai, S. *Beyond the desktop in seven dimensions*. The MIT Press, 2007, ch. 11, 335–354.

[29] Muller, M. J., Geyer, W., Brownholtz, B., Wilcox, E., and Millen, D. R. One-hundred days in an activity-centric collaboration environment based on shared objects. In *Proc. of CHI '04*, 375–382.

[30] Norman, D. A. *The invisible computer: why good products can fail, the personal computer is so complex, and information appliances are the solution*. The MIT press, 1999.

[31] Olsen, Jr., D. R. Evaluating user interface systems research. In *Proc. of UIST '07*, 251–258.

[32] Padovitz, A., Loke, S. W., and Zaslavsky, A. The ecora framework: A hybrid architecture for context-oriented pervasive computing. *Pervasive Mob. Comput. 4*, 2 (Apr. 2008), 182–215.

[33] Pering, T., Want, R., Rosario, B., Sud, S., and Lyons, K. Enabling pervasive collaboration with platform composition. In *Pervasive Computing*, H. Tokuda, M. Beigl, A. Friday, A. Brush, and Y. Tobe, Eds., vol. 5538 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2009, 184–201.

[34] Reiff-Marganiec, S., Truong, H.-L., Casella, G., Dorn, C., Dustdar, S., and Moretzky, S. The incontext pervasive collaboration services architecture. In *Proceedings of ServiceWave '08*, Springer-Verlag (2008), 134–146.

[35] Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R. H., and Nahrstedt, K. Gaia: a middleware platform for active spaces. *SIGMOBILE Mob. Comput. Commun. Rev. 6*, 4 (Oct. 2002), 65–67.

[36] Shen, C. From clicks to touches: enabling face-to-face shared social interface on multi-touch tabletops. In *Online Communities and Social Computing*, Springer (2007), 169–175.

[37] Streitz, N. A., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., and Steinmetz, R. i-land: an interactive landscape for creativity and innovation. In *Proc. of CHI 1999*, 120–127.

[38] Thompson, M. S., Plymale, W. O., Hager, C. T., Henderson, K., Midkiff, S. F., Dasilva, L. A., Davis, N. J., and Park, J. S. Ad-hoc networking support for pervasive collaboration. In *in Adjunct Proc. of UbiComp04*,

[39] Toninelli, A., Montanari, R., Kagal, L., and Lassila, O. A semantic context-aware access control framework for secure collaborations in pervasive computing environments. In *International Semantic Web Conference* (2006), 473–486.

[40] Voida, S., Mynatt, E. D., and Edwards, W. K. Re-framing the desktop interface around the activities of knowledge work. In *Proc. of UIST '08*, 211–220.

[41] Weiser, M. The computer for the 21st century. *Scientific American 265*, 3 (Sept. 1991), 66–75.

Paper 3: ActivityDesk

Title of paper

ActivityDesk: Multi-Device Configuration Work using an Interactive Desk

Authors:

Steven Houben *and Jakob E. Bardram*

Published:

In CHI '13 Extended Abstracts on Human Factors in Computing Systems ***(CHI EA '13)****. ACM, 2013.*

Abstract:

Recent studies have shown that knowledge workers are increasingly using multiple devices, such as notebooks, tablets and smartphones to interact with different types of information that are part of their daily activities. Using multiple devices introduces a configuration overhead as users have to manually reconfigure all devices according to ongoing activities. Especially in an environment such as an office, where the use of multiple devices is more common, the process of configuring them in context of ongoing activities is cumbersome. In this paper, we present the initial explorations of the ActivityDesk *system, an interactive desk that supports multi-device configuration work and workspace aggregation into a personal ad hoc smart space for knowledge workers. The main goal of* ActivityDesk *is to reduce the configuration work required to use multiple devices at the same time by using an interactive desk as a configuration space.*

ActivityDesk: Multi-Device Configuration Work using an Interactive Desk

Steven Houben
The Pervasive Interaction Technology Laboratory
IT University of Copenhagen
shou@itu.dk

Jakob E. Bardram
The Pervasive Interaction Technology Laboratory
IT University of Copenhagen
bardram@itu.dk

Figure 1: The (1) desktop application, (2) tablet resource viewer and (3) ActivityDesk application.

CHI 2013 Extended Abstracts, April 27 – May 2, 2013, Paris, France.
ACM 978-1-4503-1952-2/13/04.

Abstract
Recent studies have shown that knowledge workers are increasingly using multiple devices, such as notebooks, tablets and smartphones to interact with different types of information that are part of their daily activities. Using multiple devices introduces a configuration overhead as users have to manually reconfigure all devices according to ongoing activities. Especially in an environment such as an office, where the use of multiple devices is more common, the process of configuring them in context of ongoing activities is cumbersome. In this paper, we present the initial explorations of the ActivityDesk system, an interactive desk that supports multi-device configuration work and workspace aggregation into a personal ad hoc smart space for knowledge workers. The main goal of ActivityDesk is to reduce the configuration work required to use multiple devices at the same time by using an interactive desk as a configuration space.

Keywords
Interactive Desk, Configuration Work, Multiple Devices

ACM Classification Keywords
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces, graphical user interfaces, windowing system

Introduction

Studies have shown that knowledge workers spend on average 30% of their work time outside their office [11]. Many computing devices, such as notebooks, tablets and smartphones facilitate this mobility greatly by allowing users to carry information with them. Rather than being tied to one environment, they allow people to roam between different places. Because of the different properties of these devices, users can tailor interaction with information according to the current context, situation or environment.

Recent studies [5, 9] confirm this trend and show that users increasingly use multiple devices to access information in office settings but in the meantime also struggle to manage this information as it is scattered across devices. As concluded by Dearman et al. [5]:

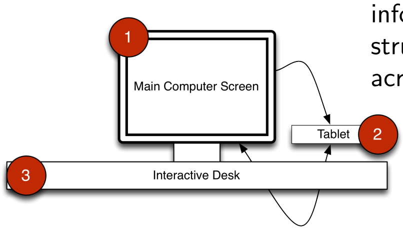


Figure 2: Devices can be used as (1) master, (2) slave or (3) mediator.

> *" Using multiple devices is increasingly the norm." However, "[..] participants reported that managing information across their devices as the worst part of using multiple devices."*

The device multiplicity and poor support for information exchange techniques and distributed workflows has increased the load on users as they are left with the burden of manually reconfiguring the devices based on the ongoing work context. Reconfiguration in this context refers to (i) finding the relevant documents and launching the correct application while moving or distributing work between one device to another device, (ii) dealing with interruptions generated by the different devices in use and (iii) updating the settings of a device. Especially, in an office setting, where the use of multiple devices is more common, the process of reconfiguring them in context of ongoing work is cumbersome because information and services are scattered across different devices.

We introduce the ActivityDesk system (Figure 1), an interactive desk that supports (i) multi-device configuration work and (ii) workspace aggregation into a personal ad hoc smart space for knowledge workers. The main goal of ActivityDesk is to reduce the reconfiguration work required to use multiple devices at the same time by using an interactive desk as a configuration space. In this paper, we describe the motivation for this work through an exploratory field study of the use of desks and the initial user-centric design of the ActivityDesk system.

Multi-Device Configuration Work

Configuration work is the amount of work required to set up an environment so it enables the user to perform a task or activity. It is the effort required to *control*, *manage* and *understand* information, applications and services that are distributed over all used devices. The reconfiguration problem when multitasking on a single device has already been recognized and widely addressed by a myriad of approaches. However, with the introduction of device multiplicity and multi-user interaction with information, this problem is greatly amplified.

Technological advances have lead to a myriad of approaches (including iLand [12], Gaia [6], Interactive Workspaces / iRos [8] and Impromptu [4]) that deal with information exchange between displays and devices in complex smart space setups. In contrast to these systems, in which the central focus is on the interaction with applications and services across devices, we propose the use of *activity* [2, 7] as a central configuration mechanism that can be used across devices.

Activity is a description of a work context (including files, applications and other meta information) that is a reflection of the real ongoing activity of the user. By

Figure 3: A selection of desks used for the analysis.

making all used devices activity aware, activities can span different devices and thus form one *activity system*. Similarly, prior work has tried to reduce the reconfiguration overhead by representing and structuring the ongoing work context into tasks or activities. The majority of related work focuses on the configuration or re-framing of the desktop interface [7] but more recently the approach has also been applied to distributed user interfaces and pervasive computing [2]. The focus of these approaches are set on a per-device configuration of information. However, the concept of activities has the potential to transcend the device and be used as a mechanism to relate and structure distributed information, services and interruptions.

The *role* of a device can play an important factor in supporting personal workflow [5] as it allows the device to adapt to the situation based on a specific configuration. A tablet computer could e.g. be used as the main device when moving from or to the office or when participating in a meeting. When the device, however, is imported into the office and put on the desk, the tablet may become secondary to another device and thus extend the digital workspace of the main device. Additionally, the type of device plays an important factor in determining its role. While tablets, smart phones and notebooks are an excellent candidate for both master and slave role, there is also the potential for devices to play a mediating role. An interactive table e.g. has the potential of replacing paper documents but also to be used as a mediator between other devices that are placed on the table.

We thus envision three types of device roles: (i) master, (ii) slave and (iii) mediator (Figure 2). A master device holds control of a specific environment as it has the central focus of the user. In an office environment, this is typically a stationary desktop computer or notebook. Since the master device has the focus of the user, any other device in the space is a slave, that is linked to the main device and serves as an extension to the periphery of the main device. The mediator finally, is used to facilitate, visualize or manage the connection between a master and slave device. Note that any device can hold multiple roles and evolve from one role to another based on the configuration of the user or predefined rules.

Interactive Desk

Exploratory Field Study

During an exploratory field study, we performed a contextual inquiry as well as an analysis of the desk space of 15 knowledge workers (Figure 3). The study showed that many participants use multiple devices in their office setting (including tablets, notebooks and phones) but also peripherals to extend screen space. Participants also seem to use these devices as part of the same general activity. Especially the tablet computer seems a popular device for use in combination with a desktop computer or notebook.

Participants claimed high ownership over their desk. The arrangement of artefacts and devices on the desk was highly personal but differed slightly between different participants. All desks could be abstracted into a meta-desk (Figure 4) that consists of two planes with different zones. First, the visualisation plane is used for the output of interaction on the desk. Participants seemed to use external screens to widen the periphery and argued that the added screen space helped them configure the different applications they are using. Second, the interaction plane (or the desk itself) consists of three different zones: (i) storage and archiving zone, (ii) input and interaction zone and (iii) the active space.

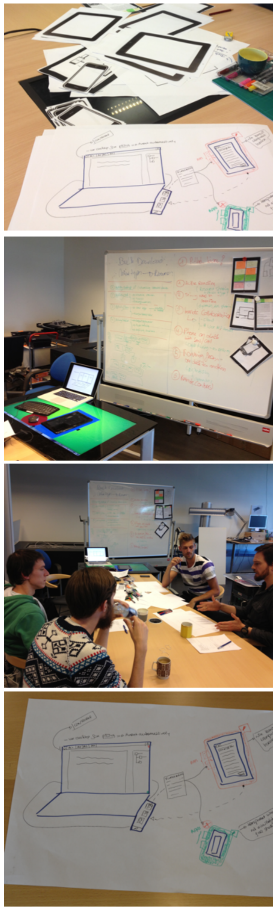


Figure 5: The desk was designed through a user-centric design process involving a number of knowledge workers.

The storage and archiving zone is primarily used to store documents, objects or other artefacts that are not used. The input and interaction zone is a space that is used for input devices such as a mouse or keyboard but also for physical input devices such as pens, markers or pencils that are used in conjunction with notebooks and post-its. Additionally, this space is also used to store and use mobile devices such as tablets or mobile phones. Finally, the active zone is the space right in front of the user that is used for highly focused work. The artefacts in this zone are objects and tools brought over from the other zones.

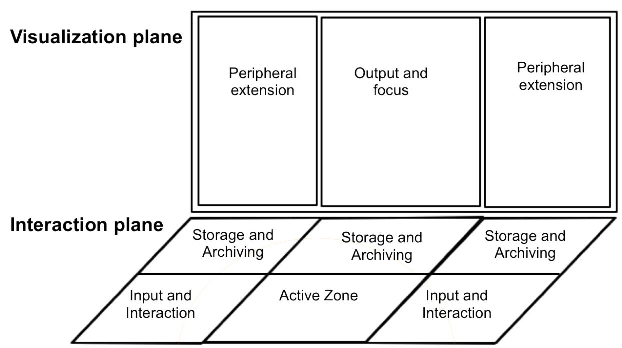


Figure 4: The zones defined in this meta-desk were found in all analysed desks.

Previous studies and our exploratory field study show that modern knowledge work is composed of both (i) mobile (or nomadic) work, which refers to a scenario where the users move between locations while doing work, and (ii) stationary work, in which the user uses a space over which they claim ownership. In the latter context, the work desk seems to play an important role in managing documents, devices and artefacts related to knowledge work. Our purpose is to design a system that allows users to seamlessly move devices between a nomadic and stationary context by using the office desk as an interactive mediating device.

Similar observations have been made in prior work which explored how tabletops or augmented desks can be used to (i) expand the screen space to a broader periphery [14], (ii) include support for real objects and artefacts [1], (iii) create aggregated information spaces [3, 10] or (iv) support multi-plane desktop interaction [13]. ActivityDesk builds on this prior work to explore the *usefulness* and *impact* of an activity-centric approach to the multi-device configuration problem.

Based on the field study and prior work, we conducted 3 participatory design workshops with 5 knowledge workers to co-design the interactive desk and applications for notebook and tablet to interact with the desk. During the workshops, we used a scenario-based approach as a starting point for paper-based mockups (designed by participants), that were discussed and evaluated. The last workshop also included a technology probe, to spark discussion on previous designs and expose participants to the actual technology (Figure 5).

ActivityDesk

The current prototype consists of the interactive table (a Microsoft Pixelsense [1] mounted in a normal desk) running the ActivityDesk (Figure 6) application, and a notebook and tablet. When the user places a notebook or tablet on the desk, it will attempt to pair with the recognized device and start a new activity session. Removing the device will stop the session and remove all shared information or resources on the desk. The desk is thus a semi-public mediator and configuration space that can be used by different people.

[1] http://www.microsoft.com/en-us/pixelsense

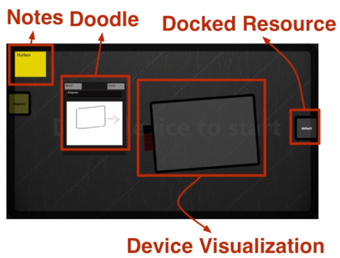


Figure 6: The anatomy of the bare ActivityDesk system.

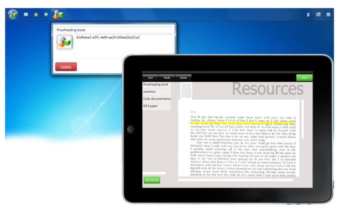


Figure 7: The desktop activity bar and resource viewer on the tablet.

Figure 8: Managing deployed resources on the desk.

The notebook runs an activity manager (Figure 7) that visualizes ongoing activities on a taskbar [7]. Each activity is assigned to a dedicated virtual desktop, which contains all files and applications that are part of that activity. Adding files to the active desktop workspace will add them to the ongoing activity and clicking an activity button on the taskbar updates the working context to the appropriate virtual desktop and/or all other connected devices. The tablet (Figure 7) is equipped with an *active reading*, *resource viewer* and an *image editing* application. Each application is activity-aware, and has the ability to interpret and visualize resources (documents, images and activities themselves).

When the desk successfully pairs with a notebook or tablet, it will display the resources (documents) that are marked on the detected devices as *shared*, and allow the users to interact with them. Adding additional devices (that can be recognized and are authorized) will add them to the ongoing activity. All connected devices are merged into one workspace (activity system) with access to all shared resources and services. To allow for local mobility, a paired device can be pinned to the desk. By clicking a button on the side of the visualisation of the detected device, the resources remain shared on the desk, but the device can be physically removed from the desk without losing the connection. A thumbnail representing the device is added to the table to indicate a still open connection with the device.

The desk can be used for multiple purposes. First of all, it functions as an *extension* of the workspace of the users, as it allows users to move documents and resources to a large surface (Figure 8). The multi-touch desk includes the ability to annotate documents, create to-do notes or doodles, or simply organize documents in activities. In addition, to facilitate interaction with large amounts of resources, users can dock resources to the side of the desk causing the desk to only show a thumbnail and short name for overview.

Second, the desk can be used to engage in cross-device resource management as users can simply drag and drop files between devices, using the table. When the user drags e.g. a pdf file to the tablet, the built-in activity manager will propose a number of applications that can handle that particular resource. The user can then select an application by clicking the appropriate button on the desk. In this mode, the desk is used for meta- or configuration work, which refers to setting up the device for use in a particular activity.

Finally, the desk can be used to select input mode. All devices that have the ability (and are configured) to share input devices can be connected through the activity session. This means that multi-touch input of the tablet application can e.g. be redirected to the desktop system.

The ActivityDesk system is built on top of an infrastructure, that is composed of a number of cloud- and local distributed activity services (including file syncing, HTTP REST publish/subscribe event system, bonjour discovery and a distributed UDP context processor). Each device has an activity client installed that hooks into these services and thus allows the device to become activity-aware. For long term persistence and distributed collaboration, the local activity information is replicated into a cloud-service and exposed through a web service. All activity related events (CRUD [2], discovery, device added,...) are automatically distributed to all devices that are connected to a local activity system.

[2] Create, Read, Update and Delete

Future Work

We presented the initial explorations of the ActivityDesk system, an activity-centric interactive desk that supports multi-device configuration work and workspace aggregation into a personal ad hoc smart space for knowledge workers. We are currently in the process of evaluating and refining the design of the first iteration of the prototype. We hypothesize that by using activities as a *configuration mechanism* and the interactive desk as *configuration space*, there will be a decrease in overhead when using multiple devices in a personal working context. However, we are also interested in side effects of the use of an interactive desk such as the positioning of devices, the possibilities of using the extra screen real estate and the perception of the user towards an interactive desk.

Acknowledgements

This work was supported by the EU Marie Curie Network iCareNet under grant number 264738.

References

[1] T. Arai, K. Machii, S. Kuzunuki, and H. Shojima. Interactivedesk: a computer-augmented desk which responds to operations on real objects. In *Proc of CHI '95*, pages 141–142, New York, NY, USA, 1995. ACM.

[2] J. Bardram, S. Gueddana, S. Houben, and S. Nielsen. Reticularspaces: Activity-based computing support for physically distributed and collaborative smart spaces. In *Proc. of CHI '12*, New York, NY, USA, 2012. ACM.

[3] X. Bi, T. Grossman, J. Matejka, and G. Fitzmaurice. Magic desk: bringing multi-touch surfaces into desktop work. In *Proc. of CHI '11*, pages 2511–2520, New York, NY, USA, 2011. ACM.

[4] J. T. Biehl, W. T. Baker, B. P. Bailey, D. S. Tan, K. M. Inkpen, and M. Czerwinski. Impromptu: a new interaction framework for supporting collaboration in multiple display environments and its field evaluation for co-located software development. In *Proc. of CHI '08*, pages 939–948, New York, NY, USA, 2008. ACM.

[5] D. Dearman and J. S. Pierce. It's on my other computer!: computing with multiple devices. In *Proc. on CHI '08*, CHI '08, pages 767–776, New York, NY, USA. ACM.

[6] C. Hess, M. Romn, and R. Campbell. Building Applications for Ubiquitous Computing Environments. In *Proc. of Pervasive '02*, pages 16–29. Springer-Verlag, 2002.

[7] S. Houben, J. Vermeulen, K. Luyten, and K. Coninx. Co-activity manager: integrating activity-based collaboration into the desktop interface. In *Proc. AVI '12*, pages 398–401. ACM.

[8] B. Johanson, A. Fox, and T. Winograd. The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE Pervasive Computing*, pages 67–74, 2002.

[9] A. Oulasvirta and L. Sumari. Mobile kits and laptop trays: managing multiple devices in mobile information work. In *Proc. of CHI '07*, pages 1127–1136, New York, NY, USA. ACM.

[10] J. Rekimoto and M. Saitoh. Augmented surfaces: a spatially continuous work space for hybrid computing environments. In *Proc. of CHI '99*, pages 378–385, New York, NY, USA, 1999. ACM.

[11] S. T. Ronald Maier. Flexible office: Assignment of office space to enhance knowledge work productivity. *Journal of Universal Computer Science*, pages 118–127, 2007.

[12] N. A. Streitz, J. Geissler, T. Holmer, S. Konomi, C. Müller-Tomfelde, W. Reischl, P. Rexroth, P. Seitz, and R. Steinmetz. i-land: an interactive landscape for creativity and innovation. In *Proc. of CHI '99*, pages 120–127, New York, NY, USA, 1999. ACM.

[13] M. Weiss, S. Voelker, C. Sutter, and J. Borchers. Benddesk: dragging across the curve. In *Proc. of ITS '10*, pages 1–10, New York, NY, USA, 2010. ACM.

[14] R. Ziola, M. Kellar, and K. Inkpen. Deskjockey: exploiting passive surfaces to display peripheral information. In *Proc. of Interact '07*, INTERACT'07, pages 447–460, Berlin, Heidelberg, 2007. Springer-Verlag.

Paper 4: ActivitySpace

Title of paper

ActivitySpace: Managing Device Ecologies in an Activity-Centric Configuration Space

Authors:

Steven Houben, *Paolo Tell and Jakob E. Bardram*

Published:

In Proceedings of the 2014 ACM international conference on Interactive tabletops and surfaces ***(ITS '14)****. ACM, 2014.*

Abstract:

Mobile devices have become an intrinsic part of people's everyday life. They are multi-functional devices providing ubiquitous access to many different sources of information. Together with traditional personal computers, these devices form a device ecology that provides access to an overlapping information space. Previous studies have shown that users encounter a number of fundamental problems when interacting with these device ecologies, such as lack of transparency, control, intelligibility and context. To mitigate these problems, we introduce ActivitySpace*: an activity-centric configuration space that enables the user to integrate and work across several devices by utilizing the space between the devices. This paper presents the conceptual background and design of* ActivitySpace *and reports on a study with nine participants. Our study shows that* ActivitySpace *helps users to easily manage devices and their allocated resources while also exposing a number of usage patterns.*

ActivitySpace: Managing Device Ecologies in an Activity-Centric Configuration Space

Steven Houben, Paolo Tell, Jakob E. Bardram
The Pervasive Interaction Technology Laboratory
IT University of Copenhagen, Rued Langgaardsvej 7, DK-2300 Copenhagen, DK
{shou,pate,bardram}@itu.dk

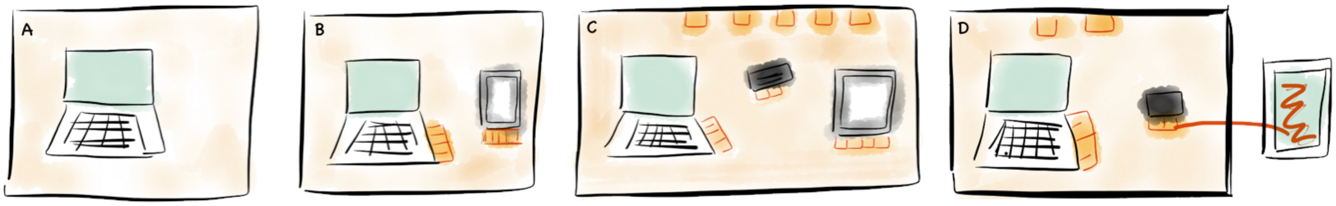


Figure 1. The configuration space is activated by placing the master device in the space (A). All other devices that are added to the space become part of the same activity space (B). The space visualizes the resource attached to each device (B). Resources can be moved between devices or stored in the space (C). Devices can be pinned to the space, to maintain the distributed configuration while allowing for device mobility (D).

ABSTRACT
Mobile devices have become an intrinsic part of people's everyday life. They are multifunctional devices providing ubiquitous access to many different sources of information. Together with traditional personal computers, these devices form a device ecology that provides access to an overlapping information space. Previous studies have shown that users encounter a number of fundamental problems when interacting with these device ecologies, such as lack of transparency, control, intelligibility and context. To mitigate these problems, we introduce *ActivitySpace*: an activity-centric configuration space that enables the user to integrate and work across several devices by utilizing the space between the devices. This paper presents the conceptual background and design of *ActivitySpace* and reports on a study with nine participants. Our study shows that *ActivitySpace* helps users to easily manage devices and their allocated resources while also exposing a number of usage patterns.

Author Keywords
Activity-Centric Computing; Configuration Work; Configuration Space; Activity Configuration

ACM Classification Keywords
H.5.2. **[Information Interfaces and Presentation]**: User Interfaces

INTRODUCTION
Mobile devices, such as smart phones and tablets, have become an intrinsic part of people's everyday life. These devices provide ubiquitous access to many different sources of information and allow users to consume as well as to produce information by selectively choosing the appropriate modality, input and output bandwidth, and interaction techniques. Together with laptops and desktop computers, these devices have become part of a device ecology in which each device acts as a specialized portal into users' personal or shared information space. The user-device mapping is quickly changing from being a one-to-one to a one-to-many or even to a many-to-many relation. In this setup, complex device ecologies are constructed and maintained by users to access collaborative distributed information spaces.

Although file sharing systems such as DropBox work across several devices, they are mostly designed according to a traditional single device paradigm, providing little or no support for more complex workflows that engage multiple devices simultaneously in the interaction. Previous studies [14, 31, 36] have highlighted numerous problems when using these tools, including lack of transparency, control, intelligibility, and context. In response to these issues, a large body of prior work has explored interaction techniques used to move information visually from one device to another (e.g., [12, 18, 34]). However, these approaches neither consider the users' ongoing activity nor how users construct these cross-device configurations. Therefore, these attempts are oblivious to the set of resources relevant to the user's activity, the ecology of devices used to manipulate those resources, and the role that such resources have within the activity. In essence, employing multiple devices to execute a particular task requires users to put a significant effort in what we call *configuration work*, the effort required to *setup*, *manage*, *understand* and *use* information, applications and services, which are distributed across several devices.

To mitigate this distributed configuration problem, we introduce *ActivitySpace*, a distributed activity-centric information management system that visualizes the active work setup of the user across all connected devices using the surrounding

ITS 2014, November 16–19, 2014, Dresden, Germany.

ACM 978-1-4503-2587-5/14/11 ...$15.00.
http://dx.doi.org/10.1145/2669485.2669493

space between the active devices. As illustrated in Figure 1, *ActivitySpace* provides a configuration space that works as a mediating interface between the user interfaces of all connected devices. A configuration space allows users to (i) manage and aggregate all devices that are part of the same device ecology, (ii) visualize all resources that are part of the ongoing activity across all devices, and (iii) configure devices by manipulating resources through interaction or by the physical properties of the configuration space.

In this paper, we first present the conceptual background of activity-centric configuration spaces. We continue by discussing the design, interaction techniques and technical implementation of the *ActivitySpace* system. Finally, we present a scenario-based user study and conclude the paper by discussing the lessons learned from the implementation and evaluation of the system.

RELATED WORK

ActivitySpace draws upon different fields of related work: (i) smart spaces and distributed user interfaces, (ii) interactive tabletops, and (iii) activity-centric computing.

The seminal work by Weiser [43] has originated a large body of research into smart spaces and distributed user interfaces (DUI). Early systems such as iLand [41] and iRoom [23] provided the first information spaces spanning across multiple screens. Aris [6] focused on supporting legacy application relocation through an interactive space window manager. Impromptu [7] supports the sharing and distribution of legacy application across different devices. Shared Substances [17] proposed to explicitly decouple data from functionality to increase support for multi-device environments. These multi-device environments have opened up a design space for cross-device interactions ranging from basic techniques such as Pick and Drop techniques [34], Touch and Interact [18], and Touch and Point combinations [10] to more advanced coupling of devices such as Deepshot [11], which uses a camera to move applications between devices.

Augmented Surfaces [35] is one of the earliest attempts of using projectors to augment a table for creating a seamless workspace and cross-device interaction techniques. The DiamondTouch system [15] is a multi-user touch system supporting collaborative work. Similarly, UbiTable [38] used the DiamondSpin toolkit [39] to support quick and seamless collaboration using shared and private zones on the table. DeskJockey [45] moved away from the explicit interaction on the table and explored its use for passive extension of the workspace. Bluetable [44] introduced a technique based on computer vision and Bluetooth to pair mobile devices to a surface. The FourBySix [19] system extends the tabletop with a flexible mouse and keyboard input system. PhoneTouch [37] allows users to touch the surface with their phones to support a range of interaction techniques. Tide [40] is a lightweight device composition system that allows users to access their smartphone applications on a tabletop using a VNC protocol. Finally, MagicDesk [5] augments the physical desk to bridge the gap between multi-touch interfaces and traditional WIMP interfaces. Despite the success of tabletops in a range of scenarios and domains, supporting complex device ecologies around tabletops remains an open issue [4]. The main difference to prior work is that the primary role of the desk in *ActivitySpace* is to mediate the interconnections between different devices that are in use. Although the desk can be used to interact with resources, it is primarily a visualizer of cross-device information exchange.

Task- or activity-centric computing has been proposed as a computing paradigm that supports users' activities rather than the resources and tools used to perform such activity. Activities are computational representations of work that encapsulate all resources and tools relevant for a specific work setting. Much of the prior work focused on the re-framing of the desktop interface [29, 16, 3, 42, 21], but a number of systems have also explored the sharing of activities across different heterogenous devices [1, 2, 26]. These systems primarily focus on simply replicating the activity model on other devices. However, in a multi-device setting this approach has severe limitations as it lacks support for distributing parts of the activity over all devices used by the same user.

PROBLEMS IN MULTI-DEVICE MANAGEMENT

On average, users own about three to six computing devices [14, 21, 36], including desktop computers, laptops, smartphones, tablets, game systems and e-readers. With the inclusion of additional devices, such as cameras, music players and smart watches, this number can increase up to 10 devices per user [24]. Although a number of tools provide support for cross-device management, studies (e.g., [14, 31, 36]) show that users encounter several challenges when doing so. These challenges can be categorized as problems associated with (i) managing one coherent work activity across several devices; (ii) aggregating and pairing devices; (iii) getting a clear model of what role a device plays in a multi-device setup; and (iv) managing resources across multiple devices.

Activity Management

One of the core problems in multi-device management is that devices are designed with a focus on applications and files, not on the activities people are using them for [14, 21]. Often users' tasks are not confined within a single device, but span these devices in different configurations based on the work condition. In highly mobile situations, people might prefer to use information on one mobile device, but once they are in an office, they might want to change that configuration. In essence, we need to move away from viewing devices as a single source of information to considering them as portals into an information space [28]. Supporting users to seamlessly move or partition parts of their task in the form of resources or UI controls across devices, allows them to better appropriate the interactive capabilities of the different devices used for working on the task (activity) they are performing.

Using and pairing devices

Device multiplicity allows users to choose the appropriate form factor, input bandwidth and interaction techniques for a particular resource. However, although using multiple devices has become common practice, devices are designed and optimized for a single-user/single-device user experience. This implies that devices are not aware of each other or their capabilities, unless they are equipped with special sensors. Basic operations such as moving files, redirecting input or quickly changing tasks are therefore cumbersome, as they require multiple steps and interactions with all devices in use [14]. Additionally, incorporating and pairing new devices in ongoing work creates a device setup overhead, which influences and determines whether a device is used at all [31]. To allow multiple devices to form one seamless distributed workspace, devices would benefit from mutual awareness about the information they contain, including their location and proximity, sensors and input capabilities [27]. This would also allow for more advanced interaction techniques such as cross-device drag and drop or push and pull information between devices [12, 18, 34].

Device role
Users frequently consider their devices as being either the *primary or master device* or being a *secondary or slave device* [24]. Most applications running on these devices, however, do not represent or incorporate this notion of *device role*. Specially, in the recent shift of mobile devices from supporting specific tasks to becoming full information accessors, the changing role of these devices can play an important part in cross-device interaction. If devices can be aware of their role and use pattern, they would facilitate cross-device application and resource management [14]. One example of such role-based functionality, is the ability of a laptop (used as primary device) to send SMS's from a desktop interface over a connected smartphone (attached as secondary device) [31].

Resource management
Many modern tools such as Dropbox or iCloud provide useful functionality such as automatic synchronization, but often lack visibility and control. Although they are a valuable technical distribution mechanism for file sharing, they neither immediately communicate which other users have access to the shared data nor whether the device can actually meaningfully consume the data [36]. Additionally, users do not always trust automatic file sharing. They often find it difficult to understand what actions are applied to their information and how they can reverse or undo these actions [14], pointing at a lack of intelligibility about the functioning of the data synchronization. There is a clear need for an additional control layer on top of these technical infrastructures that makes these underlying processes more visible to the end-user. Managing and accessing information across devices still poses significant configuration problems [14].

CONFIGURATION SPACE
As a conceptual background for describing the challenges and solutions for multi-device management, we introduce the three core concepts of *configuration work*, *activity configuration*, and *configuration space*.

Configuration work is defined as *the meta work required to find and set up all necessary resources needed to perform a specific task*. It is the overhead required to *setup*, *manage*, *understand* and *use* information, applications and services that are part of the ongoing interaction. Next, we define *activity configuration* as *a description of a work context (including files, applications and other meta information, coordination and communication tools) that is a reflection of the real ongoing activity*. This concept inherits from activity-based

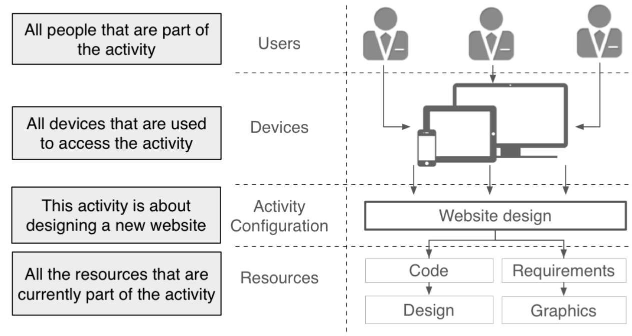


Figure 3. Activity configurations allow multiple users to access the encapsulated resources and services using multiple devices.

computing [2] and as illustrated in Figure 3, an activity configuration specifies which participants are part of the activity, what resources are used, and what devices are used as part of the activity. Compared to the traditional activity-based computing model [2] we introduce a new device layer. In addition to sharing or distributing an activity configuration between users, activities can be also *fragmented* across devices. Thus, a single activity configuration and its resources can span across several devices, and changes to its state are propagated and visualized on these attached devices. By explicitly using activities as fundamental first-class computational structures, such activity configurations can be (i) constructed, (ii) shared, and (iii) restored across devices. Thus, the notion of activity configuration is designed to reduce the amount of configuration work.

A *configuration space* is designed to support activity configurations and reduce configuration work across multiple devices and activities. A *configuration space* is defined as a *digitally augmented physical action space [33] that visualizes the activity of the user across all connected devices, using the surrounding space between the active devices*. Figure 1 illustrates a configuration space. The space is created using a specialized device (such as a projector, interactive surface, or a body worn projector) that mediates the interaction between other devices and their user interfaces. A configuration space can be either public or private and has three fundamental functions:

- **Device management** – Dynamically and visually create and manage device ecologies by coupling or decoupling devices. Based on the changing focus of the users, the space allows users to automatically or manually change the role of the device.
- **Activity management** – Create, copy, move, share, distribute, and fragment activity configurations across all devices in the ecology. The space allows for auto-configuration of devices using the configuration of previous activity states or other similar devices.
- **Interaction** – Configure devices by manipulating resources through interaction or by the physical properties of the configuration space. By leveraging the physical dimensions of the space or interaction techniques, users can pair devices, move resources between devices and re-configure activity configurations.

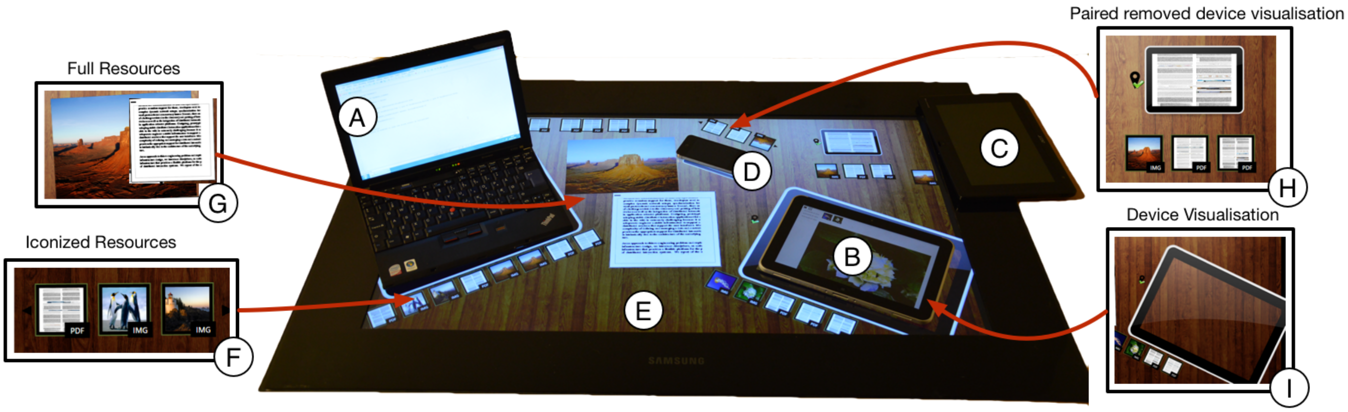


Figure 2. ***ActivitySpace*** **supports activity-centric resource management spanning across (A) laptops, (B-C) tablets and (D) phones using an (E) interactive desk as mediating configuration space. The configuration space visualizes all devices (H-I) and their allocated resources (F-G), that are part of the current activity. Additional devices can be added by placing them in the configuration space. Moving resources to a device is done by drag and drop. When the user suspends an activity, the entire space configuration on all linked devices is persisted. When the activity is resumed, the entire configuration is reestablished.**

ACTIVITYSPACE

ActivitySpace is an implementation of the *configuration space* concept. It is a distributed activity-centric information management system that allows users to create, manage and distribute applications, resources and services across several devices. Figure 2 shows *ActivitySpace*, which uses an interactive surface (e.g., a desk or meeting table) (Figure 2E) as a configuration space that allows users to interconnect and move information between devices on top of it. *ActivitySpace* consists of three parts: (i) a configuration space application that mediates the linkage to other clients; (ii) a number of platform specific clients for laptop, tablet, and phone devices; and (iii) a distributed activity-centric infrastructure that is used both for distribution of activities and resources, as well for discovery and pairing of devices.

Configuration Space

ActivitySpace is built around a personal interactive surface that works as the *configuration spaces* (Figure 2E). This space allows users to (i) easily add or remove devices to the ongoing work context, (ii) fragment resources across available devices, and (iii) save and restore such cross-device configurations when resuming and suspending activities.

Device Management

Activity-centric devices can run in three modes: (i) isolated, (ii) master, and (iii) slave. When set to *isolated*, devices shield all activities to become a single device activity system that does not allow any external connection to access the local set of activities. Whereas, if set to either of the remaining modes, devices participate in a distributed activity system. In *master* mode, a device allows attached devices to access its activity system making it distributed, which results in all activities part of the system being visualized on the devices. In *slave* mode, the device is attached to the activity system of a master device, which implies that the device has only access to resources assigned by the master device. When a new device is placed on the interactive space, the surface detects the device and adds a new visualization to indicate that it was added. If no other device is associated with the space, the newly detected device is marked as master device, meaning it has active control over other devices added to the space. If the device is not the first detected device on the space, it will similarly add a visualization to the space, but by default connect the device as a slave to the master device. Removing slave devices from the space will disconnect them from the master and cause their visualization in the space to be removed. If the master device is removed while there are connected slave devices, the first attached slave device will become the master. If no slave devices are connected, the space terminates the session. The device visualization includes a frame surrounding the physical device and two icons (Figure 2I). The first icon, visible only in case of connection failure, indicates whether the device is successfully connected to the activity system. The second one, located on the side of the device, is used to *pin* the device to the surface. Pinning a device allows users to physically remove the device from the space to allow for local mobility; in this condition, the frame surrounding the physical device becomes the placeholder representing the linked device. Finally, when multiple devices are added to the space, the resources connected to each specific device are visualized (Figure 2F). This mechanism supports both individual workflows in which all devices are owned by the same user, but also collaborative workflows where devices of other users are added to the activity system, thus providing them with access to the shared resources and activities.

Resource Management

Each time the configuration space is refreshed by linking a device as master or by switching activity from a master device already linked, the entire configuration space layout is saved and updated to the correct one. This allows users to create and switch between cross-device spatially organized activity configurations. Switching an activity on the master device causes all attached devices to change accordingly, thus, updating the attached resources visualized on the space. Creating a new activity clears all resources and provides users with a blank canvas, similar to a virtual desktop. By default each resource is shown in a iconized state. Users can use touch input to interact with the resources. The configuration spaces do not provide any occlusion handling when resources are in full scale mode. However, when iconized, resources can never be occluded by devices as they will automatically snap to either the device or dock on the side closest to the loca-

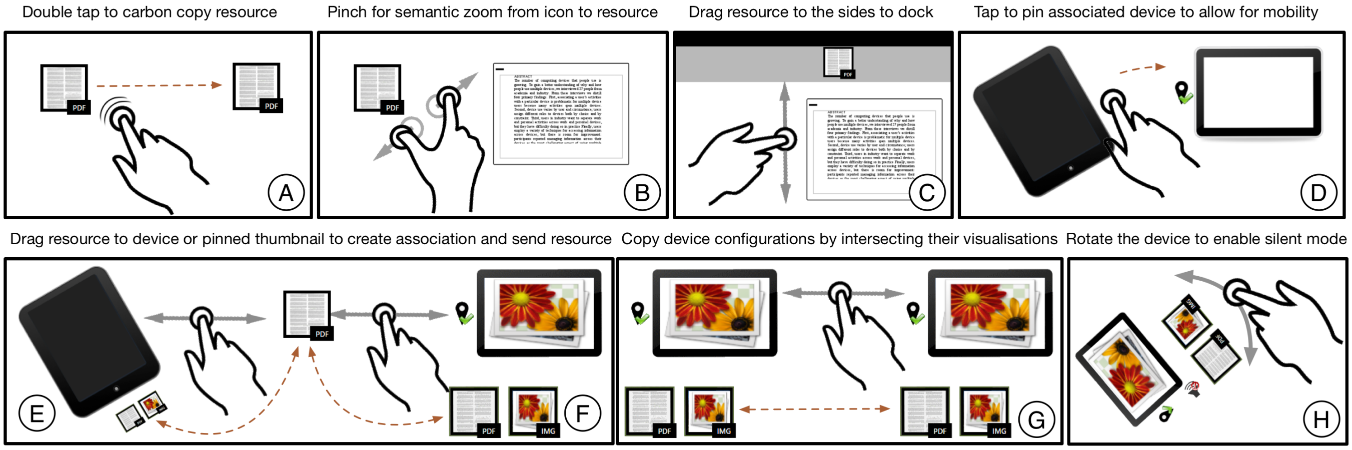


Figure 4. The configuration space provides a number of basic techniques to interact with resources and devices. User can copy (A), scale (B) or dock (C) resources on the space or drag them to another device (E). The space also allows users to pin their device (D), copy entire device configurations (G) and use spatial orientation to configure device properties (H).

tion of the resource (similar to [25]). This ensures that icons can be managed efficiently without moving the configuration problem to the physical space.

User Interaction

Figure 4 provides an overview of the interaction techniques supported in ActivitySpace, designed to facilitate visual and physical management of resources and devices on the space:

- **A : Copying** – To support the copying of resources (e.g., to share with another user), users can *double tap* the icon, causing the space to create a carbon copy of the resource. If the copied resource is in the space and not assigned to another device, it will be connected to the original icon with a visual line to indicate the relation between the nodes. This connection is removed either when one of the nodes is attached to a device by dragging it to the visualization, or by double tapping one of the icons to delete the copy.
- **B : Viewing** – By using the *pinch* gesture, users can semantically scale the resource from the iconized view to a full representation in a window (see Figure 2G). This window can be moved, rotated and scaled. Users can iconize a window by using the pinch gesture to scale down the touch window to less than 150 pixels, by using the minimize button or by double tapping the resource.
- **C : Docking** – Users can organize the space, by docking resources to the edges of the space. Dragging the resource (iconized or in a window) to the edge of the space will cause the space to render the resource as an icon and fix it to the side of the space. By dragging the docked icon back into the space, the previous state is restored.
- **D : Pinning** – Users can pin the device to the space by tapping the icon next to the visualization. A pinned device can be removed from the space without losing connection to the space.
- **E : Moving** – Icons can be dragged onto the space itself, thus detaching them from the devices. This allows users to utilize the entire configuration space to manage, compare or simply store resources.
- **F : Sending** – Icons can be dragged and dropped onto the visualizations of other devices. This will cause the underlying activity system to send the resource to that device, which in turn will show the full resource on the screen. The visualizations of the devices are updated to reflect the changes in the device configuration. To avoid accidentally sending the resource to a device on the space, they can only be sent to another device while in icon mode.
- **G : Reconfiguration** – To support fast and easy device reconfiguration, entire device configurations can be copied by bumping either the real devices or their visualizations together. The configuration is copied from the device that is moved first to the second device. This allows multiple users to quickly and easily copy an entire working context without manually dragging all resources to the space, simply to move them to another device.
- **H : Availability** – Devices (e.g., phones) can be put in *silent mode* by simply rotating them to a specific angle. The visualization updates accordingly and shows an icon that communicates the state of the phone. This approach allows users to very quickly reconfigure their availability without actually having to interact with the phone UI.

Activity-Centric Devices

Figure 5 shows the *activity-centric desktop interface* used on the laptop devices. This design leverages prior work in activity-centric computing for personal computers [3, 21] and provides users with an activity workspace supporting ad hoc configurations of windows, applications, and files. Ongoing activities are visualized on an activity bar, which mimics the normal Windows Taskbar and can be docked on any side of the screen. Clicking the buttons on the activity bar will cause the activity node to repopulate the desktop workspace with the windows and files associated with that activity. The bar is also used to create new activity workspaces and to modify activities. Files can be added to an activity by dragging and dropping them on the activity buttons in the activity bar.

Figure 6 shows the *mobile interface* used on tablets and phones that provides users with an activity-aware resource viewer and reading tool. The interface presents an overview of all resources related to an activity and allows users to annotate and modify the resources. Users can either switch activi-

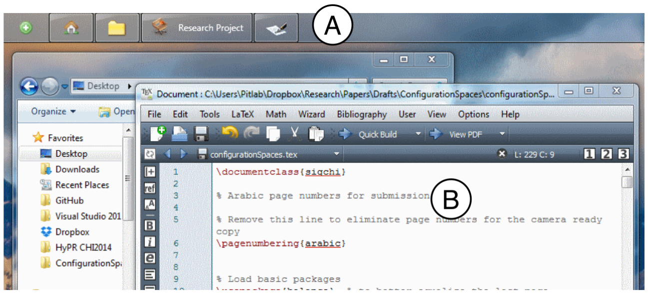


Figure 5. The activity-centric desktop application runs an activity manager allowing users to organize files and applications into an activity workspace (B). Activities are accessed and managed using the activity bar (A). Clicking an activity button on the bar will load the workspace of the associated activity.

ties using the activity sidebar on the tablet (master or isolated mode) or connect the tablet to the activity node running on the configuration space (slave mode).

Activity-centric Infrastructure

ActivitySpace is built using NooSphere [22], which is a distributed activity-centric infrastructure for management of activities, resources, devices, and users. It provides a set of basic technical services related to distributed activity configurations. The services provided range from distributed event management (using web sockets) to file and resource synchronization, from ad hoc broadcast and discovery (using Zeroconf) to a distributed context processor. The infrastructure allows the persistence of entities in the form of activities, resources linked to activities, users, and device data models. Additionally, the system supports the distribution, sharing, and fragmenting of activities across different connected devices.

Each device in *ActivitySpace* runs a specialized activity node, which is composed of an activity manager and an activity client. When running in master mode, the activity manager (AM) is used as a proxy to activities that are stored locally or in the cloud. The AM allows the device to share and distribute activities with other devices through a REST interface and web sockets. To allow other devices to connect to the AM, it runs a discovery and broadcast service (using Zeroconf). The activity client (AC) is used in slave mode and simply connects to another AM that is currently running. Using a similar discovery service, it searches for nearby AMs. Each device is augmented with a fiducial marker that uniquely identifies the device. The value encoded in the marker matches the identifier that is part of the information broadcasted by the device.

The interactive surface application runs a specialized activity node that is composed of several activity clients. When the surface applications detects a fiducial marker, it launches a discovery service to search for a device with a matching AM identifier. If no other devices are connected to the space, it loads all the activities of the AM of the detected device, and visualizes them on the space. If the AM of an other device is already connected to the space, the surface application commands the device into slave mode, and attaches the AC of that device to the AM of the previously detected master device. When the slave device is removed from the interactive space, the surface application disconnects the device from the

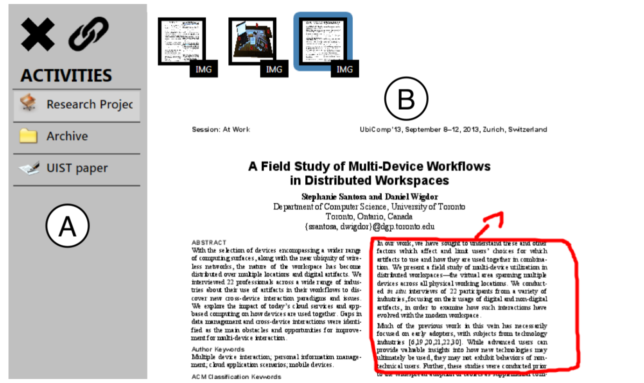


Figure 6. Mobile devices (tablets and smartphones) run an activity-aware resource visualizer and active reading tool (B). The activity bar (A) lists available activities and tools.

AM of the master device and resets the disconnected device back to its original mode. The surface application thus actively mediates the ad hoc peer to peer configuration between attached master and slave devices.

When multiple devices are connected through the interactive space, they are essentially all connected to the AM of the master devices. This means that the activities of that AM are shared and fragmented on the interactive space and slave devices. All devices (including the surface) use an event mechanism to send and receive information on (i) which devices are located on the space, (ii) which activity is currently selected in the AM of the master device and (iii) what resources are assigned to each device. All interactions with the interactive space are tunneled to the AM of the master device and propagated to all slave devices using the event system. The different cross-device activity configuration states (including location of devices, allocated resources, and spatial layout on the interactive space) are stored in the AM of the master device. The surface application can access these configurations to visualize and update them when users explicitly interact with the space or attached devices.

STUDY

To verify the usefulness of the configuration spaces concept and collect user feedback on the design of the *ActivitySpace* system, we conducted a scenario-based user evaluation [13]. The goals of this study were to (i) observe how participants would use the system for cross-device tasks, and (ii) elicit user input on the perceived usefulness of using mediating spaces for cross-device configuration work.

Study Setup

Nine users (two female and seven male, mean age = 30, σ = 4,85) from different backgrounds (such as clinical work, software development, business, and research) participated in the study. Participants rated themselves as generally experienced computer users ($\tilde{x}$= 4; *iqr*= 2 on 5-point Likert scale) and reported to be highly experienced with using multiple devices (average amount of devices = 6, σ = 1,74 including laptops, desktop computers, tablets, phone, and smart-TVs). The study was conducted in a controlled lab environment in which an interactive desk was deployed. The desk consisted of a Microsoft PixelSense built into a normal adjustable office desk. The other devices used in the experiment were a standard

Lenovo X100 laptop running Windows 7, two HP Elitepads running Windows 8, and an iPhone. All interactions with devices where logged and the experiment was videotaped.

Method

The study consisted of three phases. First, users were introduced to the general concept and functionality of the system. They were then asked to conduct a scenario using the think aloud method. The scenario focused on six key features of the system: (i) device coupling and decoupling, (ii) cross-device resource allocation, (iii) activity switching, (iv) multi-user interaction, (v) interruption management, and (vi) mobility. Participants were asked to complete both individual tasks, covering more basic functionality of the system, and collaborative tasks, focusing on the sharing of resources and activities. In the scenario, participants collected, compared and shared a number of example websites, logos and other data needed to build a new website for a company using available devices. After they successfully organized the required information in activities, they prepared a tablet with the information needed to give a presentation. After returning from the presentation, they continued to work on finding information across different activities, until interrupted by their boss who asked if the user could provide him with website designs and logos thus starting a collaborative session on the desk. Finally, after completing the scenario, participants were asked to complete a short survey, which was used as the basis for a semi-structured interview in which they were asked (i) to provide feedback on the usefulness of *ActivitySpace* and (ii) to explore potential use cases for configuration spaces.

Results

User Feedback

Figure 7 presents an overview of the results of the questionnaire on the usefulness of the different aspects of the system. Participants argued that *activities* provided them with a stable cross-device information management concept ($Q1$: $\tilde{x}$= 4; *iqr*= 1). During the scenarios, we observed how participants quickly became accustomed to using activities and even reasoned in activities. Switching between activities on all devices was considered easy ($Q2$: $\tilde{x}$= 4; *iqr*= 1) and most participants used both the tablets and the laptop computer to control the currently selected activity. They simply used the device that was most convenient. The consistent and persistent spatial configuration of devices and resources helped users to quickly switch between different work context without loosing overview:

– *"When you organize all files into activities, you avoid having too many files and clutter on the desk. You only have what you need or what you are working on." – P4*

We also observed different activity creation patterns. Some users would create a new activity for each specific sub-task right before they initiated the sub task, while others would create a number of empty activities up front. Participants also used a wide range of names and icons for the activities, as it helped them reflect better on the content of the activities. During the scenarios, some users would also update the name or icon to *"make it better reflect the work they were doing"* – P1. Because the activity structure is essentially an open ended configuration tool, participants used activities and switched between them in very different ways. As one user argued: *"flexibility is super important as not everyone thinks the same." – P8* Although the configuration space itself would actively display the name of the ongoing activity, very few users noticed this, as the spatial organization as well as the views on the devices were enough information to recall the activity. In general, most participants felt comfortable using activities across devices since they already achieved this to a certain degree in their usual device ecology using workarounds:

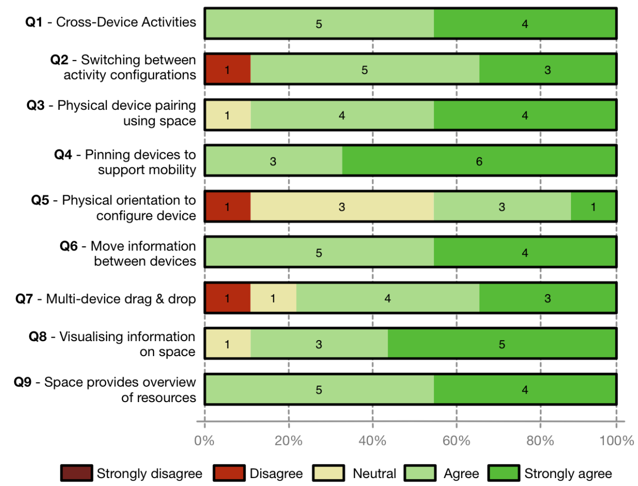


Figure 7. The results of the a 5-point Likert-scale questionnaire on the usefulness of the different parts of the system. The numbers in the bar represent the amount of participants.

– *"I guess that organizing information in activities is something everyone already tries to do, but with a lot of effort and workarounds." – P3*

Utilizing the configuration space to connect different devices to the same activity was considered very useful ($Q3$: $\tilde{x}$= 4; *iqr*= 1). Many users mentioned in the interviews that connecting devices to exchange one piece of information is often a tedious process that involves cloud storage or multiple interactions with devices. Using a physical connection between the mobile device and the space to add the device to the ongoing activity session was considered to be very useful and intuitive. Pinning the devices to the desk to allow for local mobility was also considered as a very useful feature ($Q4$: $\tilde{x}$= 5; *iqr*= 1). Participants generally mentioned during interviews that the device thumbnails helped in creating consistency between situations where the device was placed on the desk or when it was used in mid-air. Using the physical orientation to configure properties of the device was considered less important ($Q5$: $\tilde{x}$= 3; *iqr*= 1). Most participants considered this a *"nice feature"* – P2, but not really relevant to maintaining cross-device information overview.

Moving resources between different connected devices was perceived as very easy and useful ($Q6$: $\tilde{x}$= 4; *iqr*= 1). Participants appreciated the simplicity of dragging information across the space from one device to another ($Q7$: $\tilde{x}$= 4; *iqr*= 1). The externalization of resources on the space was considered as the best feature of the system. Because each of the devices visualized the associated resources in an iconized state, participants had a good overview on (i) which resources were part of the activity, and (ii) on which device the resources were allocated ($Q8$: $\tilde{x}$= 5; *iqr*= 1). This allowed them to efficiently *fragment* resources contained in an activity across different devices in use. Participants easily switched between the different resource viewing modes using the pinch gestures demonstrating the effectiveness of the modal interaction supported through semantic zooming. All participants agreed that that use of a configuration space provided them with a clear overview of all their activities, devices, and contained resources ($Q9$: $\tilde{x}$= 4; *iqr*= 1). The idea of using a physical space to make the connection between devices more visible was considered as very useful:

> – *"I really like the idea of using the empty space between devices to show what's on them. Most of my devices are currently already on or around my desk, so why not use this space." – P4*

Even during the relatively short scenarios, we observed how participants quickly took ownership over the desk. Conceptually, the configuration space is a public mediating infrastructure that requires a master device from a user to actually access that user's information. Participants liked this idea of a public space that can be used to *"do multi-device work"* – P2 and mentioned that, similar to normal tables and desks, most of them are public until one person claims ownership over it. In that case, they argued that the configuration space should be able to store local session information. The desk could for instance be used as a master log-in device that provides automatic authentication for all applications across devices. The configuration space concept was considered to be useful for both individual and collaborative work as users argued that the fundamental problem in both cases lies in providing easy and quick task and information exchange capabilities. However, the highly social character of the multi-user sharing model was received with mixed feelings. Since sharing essentially happens by one user allowing another user to place their device in their configuration space, some users argued that this might have some privacy implications related to what activities or what resources the visiting user can access.

Observations

While using devices and resources on the configuration space, we observed a number of distinct patterns of use, that occurred with most participants:

Pull and push devices – When working with information that was on both the device and on the desk, participants often moved devices around the configuration space to make more room for interacting with resources that were located on the space. Participants would consistently push devices to the back when they were no longer being used, and pull them back to the front of the space when needed.
Device on the edge – When some secondary devices (such as a mobile phone or tablet) were not in active use, participants would pin them to the desk, and place them on the edge of the configuration space outside of the tracked zone. Although the devices were not in use, participants generally kept them connected to the configuration space, "just in case".

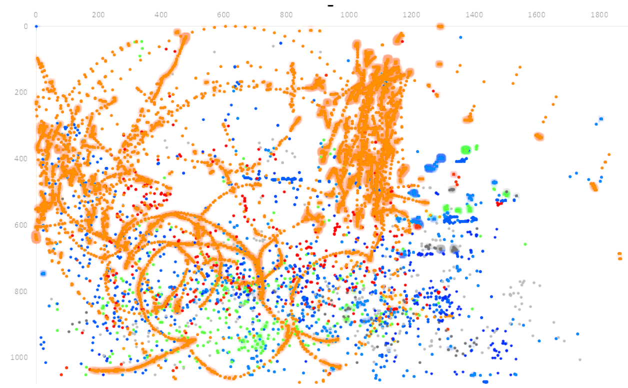


Figure 8. Touch interaction data of 9 users plotted on the desk space. The data shows circular movements of resources around visualisations (orange color). Other colors represent the touch input of participants which is primarily focussed in the front middle of the desk.

Implicit zones – During the multi-user scenarios, users would implicitly create zones in which each of the devices of the different users were located. The structure of these zones differed between users, but participants generally organized their devices in physical proximity to each other. The proximity of the devices as well as their spatial position were thus a helpful tool for users to distinguish between devices but also to get an overview of all the resources on the desk.
Maneuvering around devices – When moving resources or devices around in the space, users would often carefully maneuver them around other devices that were located on the space (orange plot in Figure 8), to avoid that a resource was accidentally associated with a wrong device.
Interaction zone – As seen in Figure 8, most interactions with resources and devices were done on the front middle of the desk, within the private zone of users. Users would move resources and devices into a focused interaction zone and essentially use the rest of space as a permanent peripheral display for configuration work or storage [20].
Devices as folders – Some users would drag and drop a number of resources to a specific device that was pinned to the space but not actively in use. This was done not to actually use them on the device, but to simply bundle them inside one physical device. Therefore, they used devices as a *physical folder* to organize a number of related resources into one manageable structure that can be moved around in the physical space.
Cross-device drag and drop – Because iconized resources can never be occluded by a device as they automatically snap to the overlaying device, we observed that long drag and drop operations across the configuration space would often result in several discrete drag and drop operations between intermediate devices. So rather than occluding the resource, the device would simply "absorb" and "pop up" the iconized resource at the bottom of the device.

DISCUSSION

The central goal of *ActivitySpace* is to provide users with a transparent platform for creating, maintaining, and sharing ad hoc device ecologies in which devices are visually connected on a *configuration space* using *activity* as a central computational concept.

As devices are increasingly becoming portals into a shared and collaborative ubiquitous information space, it has become apparent that devices can no longer be considered separate entities, but rather part of larger artifact ecology [24]. As one user concluded:

> – *"None of the things that we are currently using [in our company] is really working. We need to rethink how devices are connected and used." – P1*

Users essentially do not act *with* computing devices, but rather *through* computing devices. Bødker and Klokmose [8] therefore proposed to relate the notion of artifact ecologies [24] to human activities because *"the artifact ecology of an individual is highly dynamic"* [9] as it is constructed by the user through their activities. We build on this concept as we describe users' device ecologies as *ad hoc and dynamic interrelations between interconnected devices which are part of the same motive-oriented activity or task*. By connecting the activity of users explicitly to their devices, users are presented with a cross-device representation that moves away from the predominant application and document-centric paradigm.

ActivitySpace employs the concept of a configuration space to embrace this notion of *activity-centric device ecologies*. It supports visual and direct feedback on the connection of devices and the distribution of resources that are part of the same ongoing activity. By providing users with a physical space that has clear affordances and boundaries, users were able to easily manage multiple tasks containing multiple resources across different devices. The explicitness of placing devices in the configuration space provided users with a *stable concept* that helped them better understand and manage which of their devices are currently being used as part of their *active device ecology*. This point of view allowed users to appropriate individual devices as physical proxies for digital information [35] that could be included or excluded from their active device ecology.

ActivitySpace currently allows users to select one activity to be visualized on the configuration space and on the connected devices. Although individual devices can be decoupled from this shared activity view, the space can only visualize one activity: the one it is mediating. Participants however discussed that this could be greatly expanded by, e.g., using the space to compare a number of different activities at the same time. During the interviews, many participants essentially argued for moving more management or configuration tools to the desk, and step away from the use of the notebook as master device. One user proposed to have a cross-device task bar on the desk that could be used to switch tasks on different devices, but also give access to the sound and display properties of the device and even have a centralized notification mechanism. Participants also argued that visual feedback on the actual sending process of resources between devices, could be made even more explicit by using animations or other patterns that visualize the transition between devices in a more gradual way as described in the Gradual Engagement Pattern [27].

Sharing activities or specific resources is done using a social sharing model, in which users physically allow each other to place devices into their configuration space. Including the sharing of activity configuration into the fundamental concept improves some of the technical issues related to finding and agreeing on which tool or platform to use. However, it also creates a continuous public-private tension that is intrinsic to any interconnected artifact [32]. Allowing another user to enter one's private space is an explicit act of breaking the boundary between the self and other users [32] and is part of a continuous negotiation of intent [30]. A confirmation of this negotiation process observed during the evaluation is that each participant explicitly allowed the second user to access the configuration space, but only in a zone that was implicitly created through the spatial configuration of the devices of the owner. Although this highly social and spatial sharing model provides a stable concept for sharing information, it also opens discussion on what happens when the owner of the configuration space is not physically there. This privacy tension also explains why participants were so explicit about the purpose of configuration spaces as either public spaces—usable by anyone—or private spaces—clearly owned by one user.

ActivitySpace currently uses augmented tables and desks, using build-in interactive surfaces. This concept can however be expanded to other approaches using for instance top-mounted projectors to cover even larger spaces or support tracking of devices above and around the configuration space. Furthermore, mobile devices such as augmented reality glasses or body-worn projectors can support mobile configuration spaces. Finally, future work could explore in more depth how task management across devices can be unified in one centralized access point.

CONCLUSION

This paper introduced *ActivitySpace*, which is a distributed activity-centric information management system that allows users to create, manage and distribute applications, resources and services across different devices that are part of the same activity. *ActivitySpace* uses interactive surfaces as mediating configuration spaces that visualize the active device ecology of users. The scenario-based study demonstrated the usefulness of activity as a central concept for distributed interaction, and how the configuration space provided users with a stable concept for managing their device ecology. Finally, the study highlighted a number of usage patterns on how users appropriated the space for multi-device work.

ACKNOWLEDGMENTS

This work was supported by the EU Marie Curie Network iCareNet under grant number 264738. Thanks to Steven Jeuris and Aurelien Tabard for comments, suggestions, and feedback on an early version of this work.

REFERENCES

1. Bardram, J., Gueddana, S., Houben, S., and Nielsen, S. Reticularspaces: Activity-based computing support for physically distributed and collaborative smart spaces. In *Proc. of CHI'12*, ACM (2012), 2845–2854.
2. Bardram, J. E. Activity-based computing for medical work in hospitals. *TOCHI 16*, 2 (2009), 10.
3. Bardram, J. E., Bunde-Pedersen, J., and Soegaard, M. Support for activity-based computing in a personal computing operating system. In *Proc. of CHI'06*, ACM (2006), 211–220.

4. Bellucci, A., Malizia, A., and Aedo, I. Light on horizontal interactive surfaces: Input space for tabletop computing. *ACM Computing Surveys (CSUR) 46*, 3 (2014), 32.

5. Bi, X., Grossman, T., Matejka, J., and Fitzmaurice, G. Magic desk: bringing multi-touch surfaces into desktop work. In *Proc. of CHI'11*, ACM (2011), 2511–2520.

6. Biehl, J. T., and Bailey, B. P. Aris: an interface for application relocation in an interactive space. In *Proc. of GI'04*, Canadian Human-Computer Communications Society (2004), 107–116.

7. Biehl, J. T., Baker, W. T., Bailey, B. P., Tan, D. S., Inkpen, K. M., and Czerwinski, M. Impromptu: a new interaction framework for supporting collaboration in multiple display environments and its field evaluation for co-located software development. In *Proc. of CHI'08*, ACM (2008), 939–948.

8. Bødker, S., and Klokmose, C. N. The human–artifact model: An activity theoretical approach to artifact ecologies. *Human–Computer Interaction 26*, 4 (2011), 315–371.

9. Bødker, S., and Klokmose, C. N. Dynamics in artifact ecologies. In *Proc. of NordiCHI'12*, ACM (2012), 448–457.

10. Bragdon, A., DeLine, R., Hinckley, K., and Morris, M. R. Code space: touch+ air gesture hybrid interactions for supporting developer meetings. In *Proc. of ITS'11*, ACM (2011), 212–221.

11. Chang, T.-H., and Li, Y. Deep shot: a framework for migrating tasks across devices using mobile phone cameras. In *Proc. of CHI'11*, ACM (2011), 2163–2172.

12. Chen, N., Guimbretiere, F., and Sellen, A. Designing a multi-slate reading environment to support active reading activities. *TOCHI 19*, 3 (2012), 18.

13. Convertino, G., Neale, D. C., Hobby, L., Carroll, J. M., and Rosson, M. B. A laboratory method for studying activity awareness. In *Proc. of NordiCHI'04*, ACM (2004), 313–322.

14. Dearman, D., and Pierce, J. S. It's on my other computer!: computing with multiple devices. In *Proc. of CHI'08*, ACM (2008), 767–776.

15. Dietz, P., and Leigh, D. Diamondtouch: a multi-user touch technology. In *Proc. of UIST'01*, ACM (2001), 219–226.

16. Dragunov, A. N., Dietterich, T. G., Johnsrude, K., McLaughlin, M., Li, L., and Herlocker, J. L. Tasktracer: a desktop environment to support multi-tasking knowledge workers. In *Proc. of IUI'05*, ACM (2005), 75–82.

17. Gjerlufsen, T., Klokmose, C. N., Eagan, J., Pillias, C., and Beaudouin-Lafon, M. Shared substance: developing flexible multi-surface applications. In *Proc. of CHI'11*, ACM (2011), 3383–3392.

18. Hardy, R., and Rukzio, E. Touch & interact: touch-based interaction of mobile phones with displays. In *Proc. of MobileHCI'08*, ACM (2008), 245–254.

19. Hartmann, B., Morris, M. R., Benko, H., and Wilson, A. D. Augmenting interactive tables with mice & keyboards. In *Proc. of UIST'09*, ACM (2009), 149–152.

20. Houben, S., and Bardram, J. E. Activitydesk: multi-device configuration work using an interactive desk. In *In CHI'13 EA Abstracts*, ACM (2013), 721–726.

21. Houben, S., Bardram, J. E., Vermeulen, J., Luyten, K., and Coninx, K. Activity-centric support for ad hoc knowledge work: a case study of co-activity manager. In *Proc. of CHI'13*, ACM (2013), 2263–2272.

22. Houben, S., Nielsen, S., Esbensen, M., and Bardram, J. Noosphere: An activity-centric infrastructure for distributed interaction. In *Proc. of MUM '13*, ACM (2013).

23. Johanson, B., Hutchins, G., Winograd, T., and Stone, M. Pointright: experience with flexible input redirection in interactive workspaces. In *Proc. of UIST'02*, ACM (2002), 227–234.

24. 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*, ACM (2008), 201–210.

25. Khalilbeigi, M., Steimle, J., Riemann, J., Dezfuli, N., Mühlhäuser, M., and Hollan, J. D. Objectop: occlusion awareness of physical objects on interactive tabletops. In *Proc. of ITS'13*, ACM (2013), 255–264.

26. MacIntyre, B., Mynatt, E. D., Voida, S., Hansen, K. M., Tullio, J., and Corso, G. M. Support for multitasking and background awareness using interactive peripheral displays. In *Proc. of UIST'01*, ACM (2001), 41–50.

27. Marquardt, N., Ballendat, T., Boring, S., Greenberg, S., and Hinckley, K. Gradual engagement: facilitating information exchange between digital devices as a function of proximity. In *Proc. of ITS'12*, ACM (2012), 31–40.

28. Moran, T. P., and Zhai, S. *Beyond the desktop in seven dimensions*. The MIT Press, 2007, ch. 11, 335–354.

29. Muller, M. J., Geyer, W., Brownholtz, B., Wilcox, E., and Millen, D. R. One-hundred days in an activity-centric collaboration environment based on shared objects. In *Proc. of CHI'04*, ACM (2004), 375–382.

30. O'Hara, K. Interactivity and non-interactivity on tabletops. In *Proc. of CHI'10*, ACM (2010), 2611–2614.

31. Oulasvirta, A., and Sumari, L. Mobile kits and laptop trays: managing multiple devices in mobile information work. In *Proc. of CHI'07*, ACM (2007), 1127–1136.

32. Palen, L., and Dourish, P. Unpacking privacy for a networked world. In *Proc. of CHI'03*, ACM (2003), 129–136.

33. Pederson, T., Janlert, L.-E., and Surie, D. a situative space model for mobile mixed-reality computing. *Pervasive Computing, IEEE 10*, 4 (2011), 73–83.

34. Rekimoto, J. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *Proc. of UIST'97*, ACM (1997), 31–39.

35. Rekimoto, J., and Saitoh, M. Augmented surfaces: a spatially continuous work space for hybrid computing environments. In *Proc. of CHI'99*, ACM (1999), 378–385.

36. Santosa, S., and Wigdor, D. A field study of multi-device workflows in distributed workspaces. In *Proc. of UbiComp'13*, ACM (2013), 63–72.

37. Schmidt, D., Chehimi, F., Rukzio, E., and Gellersen, H. Phonetouch: a technique for direct phone interaction on surfaces. In *Proc. of UIST'10*, ACM (2010), 13–16.

38. Shen, C., Everitt, K., and Ryall, K. Ubitable: Impromptu face-to-face collaboration on horizontal interactive surfaces. In *Proc. of UbiComp'03*, Springer (2003), 281–288.

39. Shen, C., Vernier, F. D., Forlines, C., and Ringel, M. Diamondspin: an extensible toolkit for around-the-table interaction. In *Proc. of CHI'04*, ACM (2004), 167–174.

40. Sicard, L., Tabard, A., Hincapié-Ramos, J. D., and Bardram, J. E. Tide: Lightweight device composition for enhancing tabletop environments with smartphone applications. In *Human-Computer Interaction–INTERACT 2013*. Springer, 2013, 177–194.

41. Streitz, N. A., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., and Steinmetz, R. i-land: an interactive landscape for creativity and innovation. In *Proc. of CHI'97*, ACM (1999), 120–127.

42. Voida, S., and Mynatt, E. D. It feels better than filing: everyday work experiences in an activity-based computing system. In *Proc. of CHI'09*, ACM (2009), 259–268.

43. Weiser, M. The computer for the 21st century. *Scientific american 265*, 3 (1991), 94–104.

44. Wilson, A. D., and Sarin, R. Bluetable: connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In *Proc. of GI'07*, ACM (2007), 119–125.

45. Ziola, R., Kellar, M., and Inkpen, K. Deskjockey: exploiting passive surfaces to display peripheral information. In *Proc. of INTERACT 2007*. Springer, 2007, 447–460.

Paper 5: WatchConnect

Title of paper

WatchConnect: A Toolkit for Prototyping Smartwatch-Based Cross-Device Applications

Authors:

Steven Houben *and Nicolai Marquardt*

Published:

In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems ***(CHI '15)****. ACM, 2015.*

Abstract:

People increasingly use smartwatches in tandem with other devices such as smartphones, laptops or tablets. This allows for novel cross-device applications that use the watch as both input device and output display. However, despite the increasing availability of smartwatches, prototyping cross-device watch-centric applications remains a challenging task. Developers are limited in the applications they can explore as available toolkits provide only limited access to different types of input sensors for cross-device interactions. To address this problem, we introduce WatchConnect, a toolkit for rapidly prototyping cross-device applications and interaction techniques with smartwatches. The toolkit provides developers with (i) an extendable hardware platform that emulates a smartwatch, (ii) a UI framework that integrates with an existing UI builder, and (iii) a rich set of input and output events using a range of built-in sensor mappings. We demonstrate the versatility and design space of the toolkit with five interaction techniques and applications.

WATCHCONNECT: A Toolkit for Prototyping Smartwatch-Centric Cross-Device Applications

Steven Houben [1,2]
[1] Pervasive Interaction Technology Laboratory
IT University of Copenhagen
Rued Langgaardsvej 7, Copenhagen, DK
s.houben@ucl.ac.uk

Nicolai Marquardt [2]
[2] University College London
UCL Interaction Centre / ICRI Cities
Gower Street, London, UK
n.marquardt@ucl.ac.uk

ABSTRACT
People increasingly use smartwatches in tandem with other devices such as smartphones, laptops or tablets. This allows for novel cross-device applications that use the watch as both input device and output display. However, despite the increasing availability of smartwatches, prototyping cross-device watch-centric applications remains a challenging task. Developers are limited in the applications they can explore as available toolkits provide only limited access to different types of input sensors for cross-device interactions. To address this problem, we introduce *WatchConnect*, a toolkit for rapidly prototyping cross-device applications and interaction techniques with smartwatches. The toolkit provides developers with (i) an extendable hardware platform that emulates a smartwatch, (ii) a UI framework that integrates with an existing UI builder, and (iii) a rich set of input and output events using a range of built-in sensor mappings. We demonstrate the versatility and design space of the toolkit with five interaction techniques and applications.

Author Keywords
Smartwatch; Toolkit; Cross-Device Interaction; Rapid Prototyping; Gestural Interaction; Interface Design

ACM Classification Keywords
H.5.2. Information Interfaces. User Interfaces – input devices and strategies, prototyping.

INTRODUCTION
Smartwatches give people lightweight and immediate access to messages, notifications, and other digital data while on the go. While already powerful as standalone devices, the capabilities of smartwatches increase significantly when used in tandem with other devices that people carry, such as their phones or tablets, which allows for novel cross-device interaction techniques (e.g. [7,24]). However, so far there are only a relatively small number of explorations into *watch-centric, cross-device interaction techniques*. Building and exploring cross-device interaction techniques and applications is a difficult task, as most existing development kits have only limited support for input gesture recognition, different sensor hardware configurations, rapid interface designs, or cross-device connectivity and transfer of information.

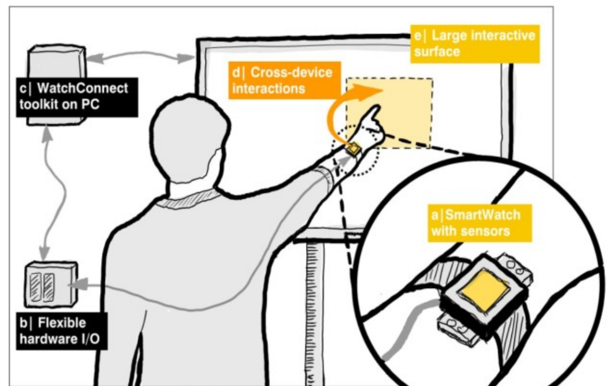


Figure 1. *WatchConnect* toolkit consists of (a) wired prototyping smartwatches with sensors through a (b) flexible and extendable hardware layer, (c) a software development platform providing user interface components and a rich set of input and output events and gestures, facilitating (d) cross-device interactions with (e) other interactive surfaces.

To bridge the gap between concept design and full implementation, we introduce *WatchConnect*, a rapid prototyping toolkit for watch-centric cross-device interaction techniques and applications (Figure 1). The toolkit provides (i) a modular and extendable hardware platform that emulates a smartwatch, (ii) a runtime system and user interface components that support quick prototyping of watch interfaces using an existing UI framework, and (iii) a rich set of input and output events and gestures using a range of built-in sensor mappings and simulators. The contribution of this paper is a *novel approach for rapidly prototyping and designing smartwatch-centric cross-device applications and interaction techniques, using simulated hardware and software building blocks.*

In this paper we first sample key related work and introduce the design of the *WatchConnect* toolkit. We proceed with the details of the architecture and components of the toolkit. Next, we demonstrate the versatility and generality of the toolkit by implementing five applications using only the basic building blocks of the toolkit. We conclude this paper with a discussion and reflection on the design and features of the toolkit, compared to other approaches.

CHI 2015, April 18 - 23 2015, Seoul, Republic of Korea

ACM 978-1-4503-3145-6/15/04…$15.00
http://dx.doi.org/10.1145/2702123.2702215

RELATED WORK

WatchConnect builds on work on interaction techniques for smartwatches, cross-device setups, and toolkit designs.

Smartwatch Interactions

Most smartwatches allow for *touch* input. Ashbrook et al. [1] explored interaction techniques for round touch-enabled watch faces. Facet [22] allows for multi-screen interactions by expanding the watch to multiple touch-enabled watch faces arranged as a bracelet. Later, Duet [7] introduced a set of cross-device interaction techniques using both the touch screen and sensors of the watch. TouchSense [18] expanded the touch bandwidth of a watch screen, by augmenting the human finger with an IMU. Finally, Mayer et al. [24] employed the touch screen of a watch to interact with objects in the environment. A number of other approaches moved touch interaction to the *bevel* and *band* of the watch face. Blasko et al. [4] support bidirectional strokes on the frame of the watch providing tactile feedback. Oakley et al. [29] expanded this idea to the side of the bevel providing high resolution capacitive input. Xiao et al. [38] moved away from a static bevel and introduced mechanical input such as panning, twisting, tilting and clicking the bevel. Watchit [33] is the first approach that moves touch interaction and scroll gestures to the wristband. More recently, Funk et al. [8] explored using the wristband for touch-enabled text entry. Finally, Abracadabra [12] is a system that supports *above* the device interaction using a magnetic input sensor.

Other systems expanded interaction with a smartwatch by using the arm or hand for gesture or touch input. One of the first explorations into smartwatches, was Gesturewrist [35], augmenting a watch with sensors to allow for hand gesture and arm posture recognition. Gesture Watch [19] augments a watch face with sensors for the detection of swipes gestures above and around the watch. Similarly, the Haptic Wristwatch [32] allows for detection of gestures such as covering the watch, turning the bevel, or swipe over the watch. AugmentedForearm [30] took this concept further, stretching the touch display of the watch across the entire forearm. Knibbe et al. [20] augmented the watch with proximity and acoustic sensors to detect hand postures and multi-finger interactions. Finally, Skin buttons [21] project touch-enabled interface elements on the skin. Other approaches include interaction with the back of a small display [3], and with small spatial aware displays such as Siftables [25].

Cross-Device Interaction Techniques

Cross-device interaction techniques have been explored in a wide range of other device configurations. Pick and Drop [34] introduced cross-device direct manipulation. Hinckley et al. [17] allow users to bump devices together into a single workspace, using synchronized gestures. Another approach is to stitch devices together to allow for cross-device pen input [16]. Hardy et al. [11] proposed to use the back of the phone to select and interact with information on a large display. Similarly, PhoneTouch [36] allows users to interact with an interactive surface, using their phones as a personal device to configure or change the interaction with the surface. Cross-device interaction techniques were described in function of proxemics in the *gradual engagement pattern* [23]. Only recently, systems explicitly used smartwatches for cross-device interaction. Duet [7] introduced a number of interaction techniques and gestures to support distributed interaction between a watch and smartphone. Mayer et al. [24] proposed "*user interfaces beaming*" to interact with objects that are in the focus of a head-mounted display. SleeD [40] uses a sleeve display for interaction techniques distributed between the sleeve and a large interactive wall display.

Toolkits and Programming Interfaces

In recent years, a number of novel cross-device interface design toolkits have been proposed to mitigate the engineering challenges in building distributed interfaces. HydraScope [14] supports multi-surface interfaces by transforming and synchronizing existing web-based applications. Conductor [10] is a prototyping framework that allows for the construction of cross-device applications and provides task-, session-, and information-management. Panelrama [39] is a web-based toolkit for DUIs that supports built-in UI synchronization across devices by allowing developers to specify the suitability of groups of UIs (or *panels*) that are used by an algorithm to automatically distribute panels across devices. XDStudio [27] is a GUI builder that supports interactive development of cross-device interfaces through the simulation of devices, or by actual on-device authoring. The Tandem Browsing Toolkit [15] is a proxy-based online multi-display application toolkit that provides developers with a declarative framework to define multi-device web pages. XDKinect [28] is a cross-device interface toolkit that uses a Kinect depth camera to mediate interaction between different devices. The toolkit allows for proxemic-aware interaction, body tracking and multi-modal input. Finally, PolyChrome [2] is a toolkit for multi-device collaborative applications that provide support for concurrency management. A small number of commercial application programming interfaces (APIs), such as the Pebble [41], Sony SDK [42] or Apple's WatchKit [43] are available for developers.

These toolkits and APIs, however, are designed for existing hardware platforms and interfaces and provide no support for novel hardware designs, custom sensor mappings or watch-specific cross-device interfaces. Although they provide means to synchronize UIs and events, using custom hardware or designing specific gestures and postures would still require substantial engineering. While still possible to build single smartwatch applications (as seen in the related work), the challenges to build those prevent rapid prototyping and experimentation [9]. Existing commercial watch APIs require proprietary hardware and lack support for rapid prototyping of cross-device applications. In contrast, *WatchConnect* provides holistic support for the entire prototyping cycle including (i) hardware design, abstraction and mapping, (ii) built-in machine learning and gesture recognition, (iii) distributed user interface and event systems, and (iv) a high level visual programming framework and tools.

INTERACTION SPACE

To summarize the challenges of supporting interaction between a watch and an interactive surface, we present an overview on the *interaction space* that emerges when connecting the input and output space of both the watch and surface.

Watch Input Space

Prior work shows that the sensors built into smartwatches provide three interaction spaces:

W1: On the watch interaction. A watch allows for direct interaction through physical contact with the device. Users can touch the screen of the watch [1,7,18,22,24], grab and interact with the bevel of the watch face [4,29,38] for discrete touch input, or interact with a touch-enabled wristband to provide continuous input [8,33]. Combining these different modalities into one watch design provides users with a very rich input device that allows for combinations of screen, bevel and strap input.

W2: Above the watch interaction. Users can perform gestures with the non-watch-arm in the three dimensional space above the watch. Although proximity sensors and depth cameras are becoming increasingly popular, only Abracadabra [12] currently supports above the watch interaction. However, a watch equipped with distance sensors or light sensors, that are frequently used to support mid-air gestures such as in SideSight [5], can provide both continuous and discrete input. This allows for a range of gestures above the watch such as covering the watch face, hovering and holding above the edges of the watch, zooming by moving the hand closer and away from the watch face or simply using the measured distance as discrete input.

W3: Interaction via internal sensing. Integrated watch sensors can provide data on the acceleration and orientation of the device that allow for a wide range of both implicit and explicit gestures. Implicit gestures can be used to, e.g., automatically turn the watch screen on or off depending on the orientation of the watch. As demonstrated by Duet [7], TouchSense [18] and GestureWrist [35], explicit gestures allow users to switch interaction modes or express hand posture and gestures. A high granularity of input allows one to use the watch as a game controller or to express different input forms with the watch hand. Similar to other interaction spaces, the integrated sensors support both continuous and discrete input.

Interactive Surface Input Space

When wearing a watch to interact with *another touch screen* – for example a tablet or a digital whiteboard – the setup has three basic input spaces (informed by [37]):

S1: Interaction Connector Point. Because the watch hand can be recognized using the built-in sensors (as demonstrated in Duet [7]), it can be used to identify the user and to connect a specific user session to the interactive display. Identifying the user behind a touch input, as done by Schmidt et al. [36] using a mobile phone, allows applications and interaction techniques to incorporate user specific functionality, to personalize the user interface or to use the input for authentication.

S2: Interaction Collision Plane. When touching the external touch screen with the watch hand, a two dimensional input space is created that is merged with the normal touch-based input space. The screen can differentiate between touches performed with the watch hand and non-watch hand. This allows applications and interaction techniques to consider bimanual input in which specific modalities or functionality is assigned to a specific hand. Furthermore, the built-in sensors allow the screen to detect touches from the watch hand with a higher degree of granularity, thus allowing for the detection of, e.g., back of the hand, knuckle or nail touches [7].

S3: Interaction Volume. The orientation and acceleration of the watch hand can be used for expressive input, adaptive user interfaces or even mid-air gestures. Furthermore, by combining three-dimensional spatial interaction with touches from the non-watch hand, applications and interaction techniques can support advanced scenarios. Examples include navigation in three-dimensional applications, game input, gestural interaction, and gradual transitions of UI elements between devices [23].

Joint Output Space

When using the watch and interactive surface, the combination of both displays creates an output space that can be used in three configurations:

O1: Output on interactive display. The output of the interaction technique or application is shown only on the display of the interactive screen, and not on the watch. This configuration can support scenarios in which the watch is used purely as an input sensor (such as, e.g., detecting how the watch hand is touching the screen [7]) or when user-specific personalized user interface elements are shown on the display [37] based on touch input.

O2: Output on watch display. The output of the interaction with the interactive display is only shown on the small watch display. This setup can be used to provide a private or contextual view (such as, e.g., a peephole metaphor on a static map) of the data shown on the interactive surface [37].

O3: Output distributed across displays. The output or feedback of the interaction between both devices is distributed or shared across both displays [37]. This configuration allows for scenarios in which both the interactive display and the watch display are updated to reflect or visualize cross-device interactions.

Temporal Synchronized Interaction

User actions in this interaction space combine input and output spaces of both devices. By performing temporally sequenced touches, postures and gestures, users can express input and interact with the dual setup. *Temporal interactions* provide users with a fine-grained distributed interaction framework. Interaction designers can combine touches, postures or gestures in any arbitrary sequence. *WatchConnect* is designed to support the prototyping of temporal interactions.

TOOLKIT

To mitigate the challenges in designing and prototyping watch-centric cross-device interaction, we present the *WatchConnect* toolkit. The major goal of the toolkit is to provide *a fast event-driven platform* for *rapid prototyping* of *watch-centric cross-device interaction techniques and applications*. The *WatchConnect* toolkit is composed of two parts: (i) a flexible and extendable hardware platform that emulates a smartwatch, and (ii) a software platform providing user interface components and a rich set of input and output events and gestures, based on default sensor mappings. The toolkit is integrated with an existing visual user interface design tool (WPF Visual Studio) to support a rich set of existing UI components and framework, and existing platforms for rapid hardware prototyping (Phidgets [44] and Arduino [45]). In this section, we provide an overview of the architecture and components of the toolkit.

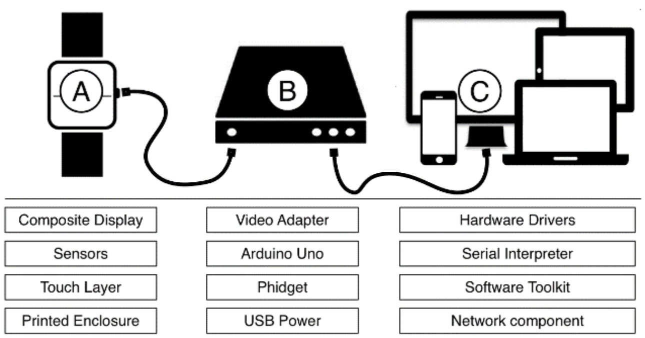


Figure 2. An overview of the *WatchConnect* toolkit.

Hardware

The *WatchConnect* toolkit is built around a wired *prototyping watch*, a smart watch emulator (Figure 2A and Figure 3) that is composed of a miniature display, a number of touch and motion sensors, and a microprocessor integrated into a form factor that *resembles a smart watch*. Using a physical cable, the *prototyping watch* is connected to the base station (Figure 2B) which converts and sends the data from the sensors and screen over a USB cable to the development computer (e.g., tablet or a large interactive surface) that runs the emulator software as well as the main toolkit (Figure 2C).

The default *prototyping watch* (Figure 3) is built around the Arduino platform and contains a light sensor, two infrared proximity sensors, an 8 channel capacitive touch sensor, a six-axis MEMS motion tracker (gyro + accelerometer), an RGB led, a flexible force sensing potentiometer and a 2 inch TFT display. The hardware components are soldered on a PCB, which slides into the 3D printed enclosure that is mounted on a wristband. Because of this setup, developers can easily extend the design with additional sensors, reconfigure the layout of the sensors or even redesign the existing watch hardware. Although the default watch uses Arduino, the toolkit also supports Phidgets to allow for fast plug and play prototyping (but somewhat bulkier components) without the need to write code for the hardware emulator.

The base station, which is connected to the development computer device using a USB and a VGA cable, consists of an Arduino microprocessor, a Phidgets interface kit, a USB power supply and VGA to component converter. All sensors are connected to either the Arduino or Phidget interface kit, which push the sensor readings over a serial protocol to the master device. The VGA converter converts the screen output from the development computer into a component signal, which is shown on the miniature display. To allow the software toolkit to analyze sensor data, a structured data exchange protocol is used which is composed of three parts: (i) a header that describes the sensor, (ii) the body that contains the sensor readings and (iii) the closing symbol that signifies the end of a package.

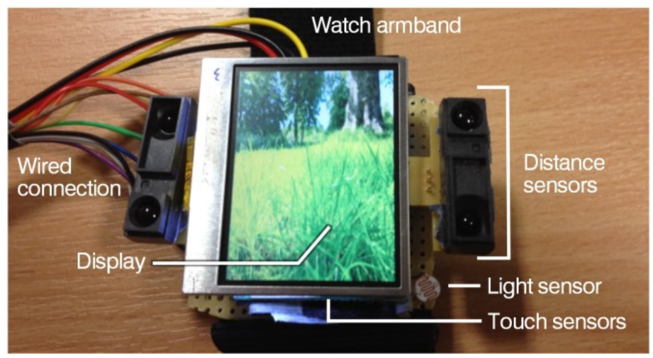


Figure 3. The Arduino-based watch probe with sensors.

Software

The toolkit's software architecture consists of six modules (Figure 4): (i) an *interface* library that includes a runtime and UI framework for the watch, (ii) an *input* library, providing touch, gesture and tracking input, (iii) a *hardware* and (iv) *processing* layer that abstracts the hardware and machine learning into events and gestures, (v) a *network* library that wraps REST and web socket services around the watch runtime and (vi) a *tools* library that provides applications to inspect and calibrate raw sensor data. In this section, we provide more details on the different modules.

Toolkit.Interface

The *WatchRuntime* is the central object of the toolkit in which all other toolkit components are merged into a single runtime environment that is used by developers to create a new watch application. The runtime, that can be configured and setup by using the *WatchConfiguration*, initiates all input, sensor management, processing and output into a watch window. When the hardware base station and watch probe are connected, the *Window Manager* of the runtime will push the watch window to the probe. If no hardware is connected, the runtime launches a native window to show the output. Watch applications can be designed using standard c# Windows Presentation Foundation (WPF) components in Visual Studio and the Expression Blend UI designer.

The only requirement for compatibility with *WatchConnect* is that watch applications are designed as user controls that inherit from the WatchVisual class, provided by the toolkit. Applications can be added and launched in the runtime by simply adding them as a new visual. Internally, the runtime manages all applications using a *WatchManager* that provides developers with a basic operating system-like environment to swap out watch applications. Through the *WatchRuntime,* the developer can access high level abstract gesture, touch and tracking events, which can easily be integrated into the interface design. Although we expect that most developers' needs reside in this high level abstraction space, we will later show how developers can make use of all lower level layers, right down to the hardware.

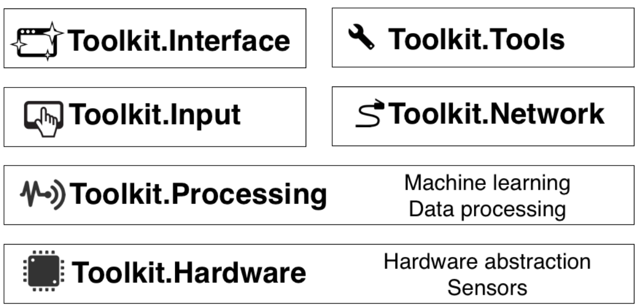


Figure 4: The architecture of the *WatchConnect* software.

Toolkit.Input

The input module provides three built-in input managers: a touch manager, gesture manager and tracker manager. First, the *TouchManager* encapsulates all "*on the device touch sensors*" - such as the *SlideTouch* device for the wristband, and a BevelTouch device - and presents the developer with high-level events including *TouchDown*, *TouchUp* and *TouchDoubleTap* events but also more complex and specialized events such as, e.g., *SliderDown*, *SliderUp* or *BevelMultiGrab*. Second, the *GestureManager* encapsulates "*above the device sensors*" – such as a light sensor and infrared proximity sensors, and tracks the internal state of the sensors using configurable thresholds and gesture detection algorithms to detect higher level gestures and postures. The abstract gesture events accessible in this manager include *SwipeLeft*, *SwipeRight*, *HoldLeft*, *HoldRight* and *Cover*. Finally, the *TrackerManager* encapsulates all the "*interaction via internal sensors*" such as the accelerometer, gyroscope and magnetometer. Similarly to the other managers, the *TrackerManager* monitors the internal state of the sensors and presents the developer with abstract high-level gestures or postures. These include an abstract IMU representation that includes the raw acceleration data, the world acceleration data, the angular motion, magnetic data and the yaw, pitch and roll. Using machine learning methods (defined in the *Toolkit.Processing* module), the manager can also detect gestures and postures that are defined by the developer who can provide training data and labels to the manager through the *WatchConfiguration*. This data can be collected using the data capture tool (provided in the *Toolkit.Tools* module). The gesture detection events provide users with the detected label as well as the probability and score of the detection. All three managers provide access to raw fused sensor data and can easily be extended by developers who can add new sensors, create new events or even add new managers (e.g., for "*on the skin sensors*"). Finally, each manager has a built-in simulator that allows developers to trigger events using simulated input such as a 3D controller or simulated data.

Toolkit.Hardware

The toolkit operates using an abstract HardwarePlatform, which can be an Arduino, Phidget or any other hardware platform that supports the *WatchConnect* protocol. The hardware module provides low level plug and play serial port management and allows managers to hook into the serial data loop to filter for specific data packets. Individual sensors are created and initiated in the managers, but use the packet definition to internally update their values. Although the toolkit supports a wide variety in sensors, there are four high-level abstract sensors: touch sensor, multi-touch sensor, proximity sensors and an IMU. These can represent a wide range of low level sensors ranging from flexible linear force resistant potentiometers to multi-channel capacitive sensors, various types of IMUs, and light and distance sensors. Although the managers provide high level events, developers can add custom lower level events directly to the sensor in order to listen or monitor changes in the internal values. Every sensor instance has an internal dynamic event mechanism that allows programmers to define events with a custom condition, which is checked and triggered from the internal value update function. This is achieved by allowing developers to inject methods into the execution body of the sensor. Finally, if new sensors are added to the setup, developers can add a new and custom hardware packet listener to the managers. This packet listener can be included in an existing manager, a newly defined manager or be used directly inside the existing watch application setup.

Toolkit.Processing

To support gesture, posture and pattern recognition, the Toolkit.Processing module provides a number of machine learning algorithms and data structures, that are built using the Accord framework and are integrated into the toolkit. The processing module includes a dynamic decision tree generator and a dynamic time warping (DTW) template engine that both use the training data and labels provided in the WatchConfiguration. The toolkit will use the training data to generate internal structures that are used by the managers to match recorded templates (e.g., for "*above the device*" proximity sensors) or monitor for gestures inside a time window.

Toolkit.Network

To allow multiple watches (connected to the same or multiple master computer devices) to interact with each other, the toolkit includes a network module that provides a websocket service that wraps the WatchRuntime and exposes all events over a real-time data connection. The module also includes *Bonjour Discovery* services to allow for zero-configuration networking support and broadcasting of watch addresses. It also provides a number of abstractions to distribute and share descriptions of the watch applications. The network module distributes watch data, meaning that each watch renders the data locally if new data is received from other watches.

Toolkit.Tools

To support developers in debugging and using the software framework, the toolkit includes a number of tools. First, the InputVisualizer provides developers with a number of visualizations that present the raw sensor data and allow for the testing of the machine learning and pattern matching data. Second, the DataRecorder provides a visual interface to record sensor data. Developers can select the data, sample rate and file location of the captured data. The recorder also allows developers to label the data as it is being recorded.

CROSS-DEVICE INTERACTION TECHNIQUES

To demonstrate the *functionality* and test the *feasibility* and *applicability* of the toolkit for the design of cross-device applications, we present five different interaction techniques implemented in realistic applications. The implementations of all applications and techniques use only the standard toolkit components, events and machine learning of the toolkit and do not include any specialized code or external tools. Table 1 provides an overview of the applications (with lines of code), and how the applications utilize the input and output of the interaction space. All applications were designed using a drag and drop editor for all UI elements, with minimal background code to link the UI to the underlying toolkit through high level objects and events. Although all applications are demonstrated on a laptop with interactive touchscreen, these techniques and applications are also usable and suitable for tablets and large horizontal or vertical surfaces. The purpose of these example applications is to demonstrate the types of advanced applications that can be constructed using only default components of the toolkit. Although these applications can be built using other methods (as demonstrated in [7,24,40]), these include custom hardware design, machine learning and other advanced computer science skills that many interaction designers do not have.

Application 1: Data Transfer

One of the core problems in multi-device information spaces is the fast, intuitive and easy transfer of files and resources across different devices [23]. A body of previous work (e.g., [23,34,36]) has explored how information can be seamlessly transferred across devices. These techniques can be expanded to smartwatches that have the potential to become *wearable mediating storage devices* that allow users to easily move their personal information to any display or device on hand. The *touch and swipe* technique allows users to connect their smartwatch to a display and use a mid-air swipe gesture to send information to the display. Users first touch the display with the watch hand to create a connection between the two devices. After the watch hand touch is recognized and the user touches an empty space, the user interface reveals a colored rectangle that is filled up over a period of two seconds. The color represents the active information on the watch, and the filling of the rectangle visualizes the time window in which the user can perform gestures to move the resource to the display. If the time window passes, the system dismisses the watch connection and treats the touch as a normal touch input. If the user performs a left to right swipe during the time window, the resource on the watch is sent to the display and shown as a touch-enabled resource (Figure 5A-B). To select which resources to send to the display, the user can use the wristband touch sensor to scroll between the different resources stored on the watch.

Application	Lines of Code	Watch: W1: Touch	Watch: W2: Above	Watch: W3: 3D Move	Screen: S1: Identify user	Screen: S2: Multi touch	Screen: S3: 3D Space	Output: O1: W feedback	Output: O2: S feedback	Output: O3: Distr. UI
Data transfer	164	•	•		•	•		•	•	•
Privacy	50			•	•			•	•	•
Navigation	98	•	•		•			•	•	•
Reading	132	•		•	•	•			•	•
UI distribution	47				•		•		•	

Table 1. Five example applications with their lines of code and how they use the interaction space.

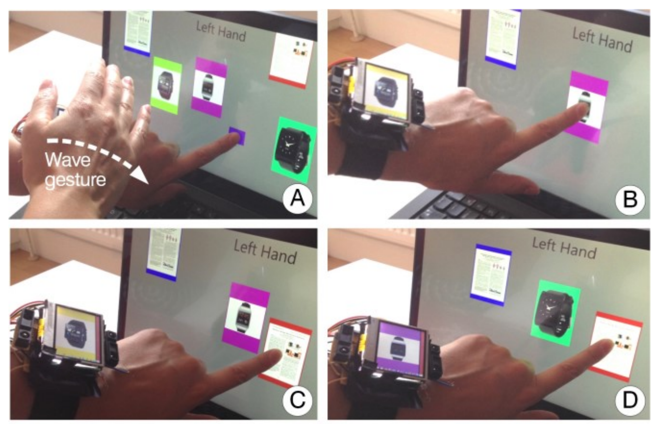


Figure 5. Users can perform gestures to move information between the display and the watch (A-B). The UI reveals part of the resource in the form of a color and shape (C-D).

If the user touches an existing resource on the display, the watch will update the UI to reveal that the watch can receive the resource, by showing a colored border on the right side of the watch (Figure 5 C-D). If the user performs a right-to-left swipe during the reveal time window, the resource is removed from the display and sent to the watch. When interacting with the resources on the surface, the UI can distinguish between left hand and right hand touches. As a consequence the UI only offers time windows to send information between devices, if a touch is linked to the watch hand.

The application leverages the entire software stack of the *WatchConnect* toolkit and was built in only 164 lines of code in a single class. It uses the built-in gesture recognizer to detect the watch hand. The UI elements are simply relocated between the watch runtime and the full screen application.

The different type of touch inputs (watch hand, non-watch hand) are channeled through events and coupled directly to the UI. The input layer on the watch automatically captures touch input on the bevel and updates the UI on the watch.

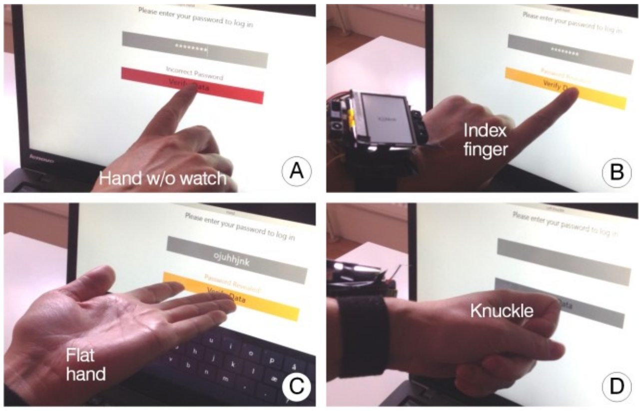


Figure 6. Users log in with the non-watch hand (A) or use the index finger or flat hand to show the password on the watch or screen (B-C). Users reset the password using the knuckle (D).

Application 2: Privacy and Password Access

The next technique facilitates access to highly private data such as passwords, bank account data or personal email. The *pose and touch* interaction technique provides users with a rich set of interactive capabilities to enter or correct a password field. Similar to the previous technique, the watch is paired to the display by touching the screen. However, in this case, the screen will monitor the posture of the hand at the moment of touching the screen. This means that the screen cannot only detect if the watch hand is touching the screen but also with which part of the hand (similarly to [7,13]). Touching the button with the non-watch hand validates the password and provides appropriate feedback (Figure 6A). The user can reveal the content of the hidden password field on the watch display, by touching the button with the index finger of the watch hand (Figure 6B), or on the touchscreen, by touching the button with the flat watch hand (Figure 6C). Users can reset the password field by touching the button with the knuckle of the watch hand (Figure 6D).

The application uses the gesture recognizer to distinguish between four different hand postures. Each posture is pushed to the UI as a different event, allowing the UI code to simply switch states and push the correct UI to the watch runtime or full screen application. This example was built in 50 lines of code and allows developers to focus only on the UI.

Application 3: Supporting Map Navigation

Interacting with maps often requires users to modify the view, find a location, or start route planning. Most maps currently provide little support for using additional devices to expand or distribute the view on the map. The *touch and push* interaction technique allows users to modify a custom secondary view on the display of the smartwatch, while using the interactive touchscreen for an overview of the general environment they want to explore. After touching the screen with the watch hand, the maps on both displays are synchronized (Figure 7A). The watch map has a default zoom level that is twice that of the main map. This allows users to quickly glance at the watch for more details as they explore the map. By touching the bevel of the watch, users can zoom in and out of the customized view, or toggle the watch map between a satellite view or the traditional map view (Figure 7C). When users interact with the map on the touchscreen, the map on the watch follows the movement, thus keeping both views synchronized. When users explore the customized view on the watch in more detail, they can synchronize the main map to that of the watch by using the *touch and swipe* gestures (Figure 7D). Finally, for selecting small targets, such as placing pushpins or route marks, the display of the watch can be used as a scope to zoom and find the exact location (Figure 7B). The user can touch the screen and press the left bevel of the watch to mark the point on the main map.

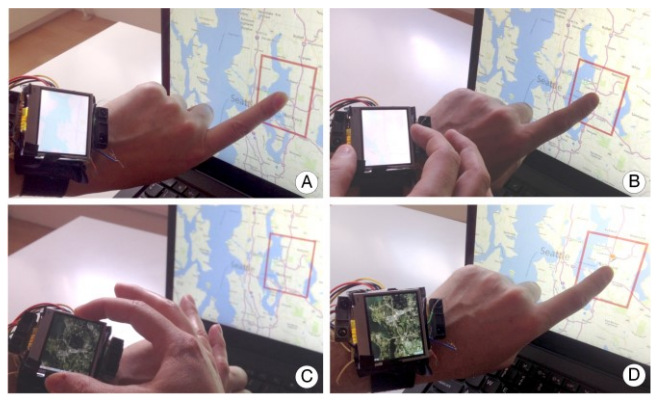


Figure 7. The watch screen shows a mini map (A), and allows users to zoom (B), change view (C) or mark locations (D).

This example was built in 98 lines of code, and utilizes the input layer to channel input from above and on the watch probe to the main interface on the surface. The application uses the temporal events to synchronize views between the watch runtime and main application, but integrates with a standard Bing maps component available in WPF / C#.

Application 4: Support Active Reading Applications

With the increasing availability of touch-enabled devices, active reading applications integrate new forms of touch-based interaction. The *gesture and touch* interaction techniques support a range of input techniques designed to create a fluid active reading application. In this application, the non-watch hand is used for *passive* browsing and reading, while the watch hand is used for *active* editing. Users can simply scroll through the text by performing on-screen swipe gestures using the non-watch hand (Figure 8A). By touching the bevel of the watch, users can browse through the menu items, thus, changing the selected option, which determines the effect of touching the screen with the watch hand (Figure 8B). The finger of the watch hand thus becomes *a reconfigurable instrument* that can be used for basic annotation with a black pen (Figure 8C), painting with a translucent brush (Figure 8D), marking text with a yellow marker (seen in Figure 8E) or as an eraser. Users can use the knuckle of the watch hand to select and copy text to a clipboard (Figure 8F).

This example was built in 132 lines of code, and again leverages the ML and processing features of the toolkit to augment a basic e-reader with advanced gestural interactions. The recognizer of the toolkit channels the recognized labels through events to the UI, which can simply be updated. Similar to all other examples, the UI itself is designed using drag and drop WPF C# components available in the Visual Studio IDE. *WatchConnect* simply connects the gestures and sensors of the watch to the already existing UI components.

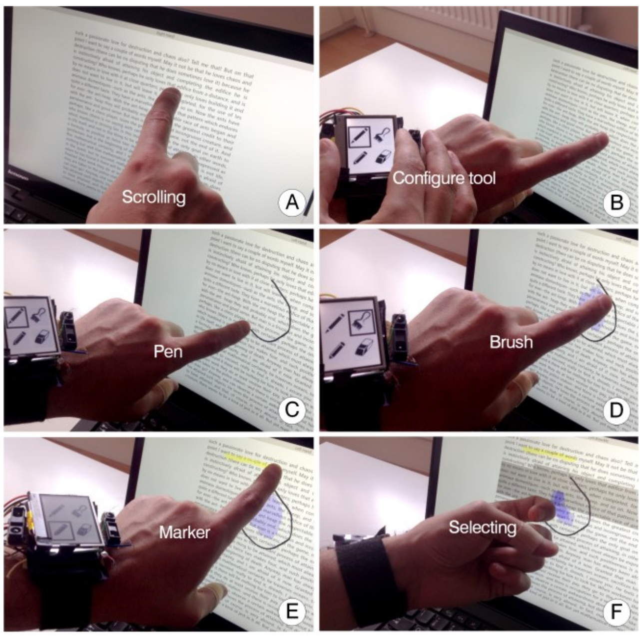


Figure 8. Users can browse text with the non-watch hand (A), configure the watch hand (B) into a pen (C), pencil (D), markers (E) or use the knuckle to select text (F).

Application 5: User Interface Beaming

One important research challenge in cross-device information spaces is how user interface elements can be seamlessly moved between different connected devices. Prior work has proposed the notion of "*user interface beaming*" [24] for mixed reality environments, or the flashlight metaphor [6] for transferring user interface elements from one device to another. Smartwatches can play a mediating role in *defining*, *exchanging* and *using* interface components or data. In this *touch and beam* technique, a UI element is initially only shown on a watch. After connecting the watch to an interactive surface by touching the display, the user interface is sent to that bigger display, to provide a bigger space for the output and utilize the potentially more advanced features provided by that device. E.g., an incoming phone call on a smart watch (Figure 9A) is simply transferred to a bigger display with better sound and camera by touching the display and connecting the watch. The hand acts like a flashlight that beams the interface on a larger canvas (Figure 9B), thus, increasing the interaction space for the user interfaces.

This example was built with 47 lines of code and uses the layout engine of the watch runtime to relay UI components based on synchronized event triggers. Designers do not need to define a multi-device context but can simply rely on the toolkit to move UI elements between the watch runtime and the main surface application.

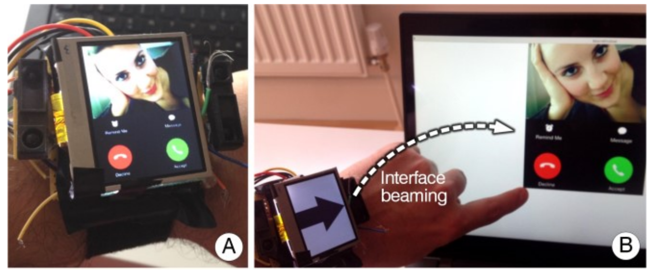


Figure 9. UI elements (A) can be beamed to the surface (B).

DISCUSSION

Designing, prototyping and testing cross-device interaction techniques with smartwatches is a complex task. To mitigate these challenges, we introduced *WatchConnect*, which uses a watch prototyping emulator to provide developers with a platform for the rapid design and prototyping of cross-device interaction techniques. Developers can create their own sensor and hardware configurations and use the software framework for easy and fast access to those hardware designs. In this section, we thematically compare *WatchConnect* to other approaches using Olsen's framework [31].

Problem Not Previously Solved

Commercial smartwatch APIs (such as [41,42,43]) provide limited support for existing hardware and single screen user interfaces. These APIs are designed to provide a *path of least resistance* towards specific UIs, but are not designed to explore novel interaction techniques and alternative designs. In contrast, *WatchConnect* allows designers to experiment, build and evaluate a range of different hardware designs, interaction techniques and gestural cross-device applications without any knowledge on distributed computing, hardware development and interfacing, data processing and sensor fusion, machine learning and networked setups. With the use of smartwatches and other mobile and wearable devices, providing tool support for designing UIs across an ecology of devices, becomes increasingly important and relevant.

Earlier cross-device UI toolkits – such as HydraScope [14], Conductor [10], or XDStudio [27] – lowered the threshold for developing applications spanning the ecology of devices. *WatchConnect* builds on top of these toolkits, and extends this work with a specialized support and focus on smartwatch specific interaction techniques, support for diverse hardware platforms, custom sensor mappings or creation of watch-centric gestural interactions. The toolkit also draws from previous work in watch-centric cross-device applications (such as Duet [7], UI Beaming [24] and SleeD [40]) to generalize these approaches and allow for rapid prototyping of complex cross-device interaction techniques using different display sizes and novel sensor input.

Reduce Solution Viscosity

Compared to other methods to develop watch-centric cross-device applications, *WatchConnect* dramatically reduces development viscosity [31] by providing a flexible architecture that allows for expressive leverage. The example applications demonstrate the range of scenarios that are supported by the toolkit. By moving all complex processes into high-level objects and events, designers can create complex gestural interactions with little overhead. However, designers with expert skills can leverage the toolkit and access, modify and create complex low level sensor mappings, custom hardware protocols and advanced machine learning approaches. Furthermore, the layered architecture allows for potential replacement of the UI layer with another existing cross-device toolkit (such as HydraScope [14], Conductor [10], or XDStudio [27]) to leverage existing multi-device features, while still using the watch-centric features of *WatchConnect*.

Empowering New Design Participants

Without adequate toolkit support, exploring watch-centric cross-device systems remained the domain of designers with highly specialized computer science skills. This is reflected in the very few watch-centric cross-device systems so far, and the number of simulation techniques used (e.g., using smartphones as watch proxies or relying on Wizard of Oz approaches). *WatchConnect* focuses in particular on making this emerging technology accessible to new, non-expert programmers. Complex applications and interaction techniques that support gestures, postures and multi-device synchronization can be designed in a short period of time without in-depth knowledge of distributed computing or machine learning. The toolkit lowers the threshold [26] for beginning the exploration of cross-device smartwatch applications, and it allows designers to focus their efforts on creative design solutions [9] for the actual cross-device user experience and interface design. In particular, the software abstracts sensor input from *above*, *on* and *in* the watch into abstract high-level events and objects for easier configuration and use. While an in-depth study of developers applying the toolkit in practice is part of our future work, we see *WatchConnect* as a fundamental step towards rapid iterative smartwatch prototyping, facilitating the exploration of novel cross-device behaviors.

Power in Combination

WatchConnect combines distributed UIs, machine learning, hardware management, data processing and simulation into one toolkit. Each building block is highly decoupled, allowing for the design of new layers or the inclusion of other approaches. The basic building blocks of *WatchConnect* can be used in complex temporal and spatial sequences that provide a power by combination [31]. *WatchConnect* integrates with C# / WPF to support a major and stable development platform that provides designer-level abstractions and a flexible and broad UI framework with numerous tools and libraries [9]. Once the design of the interaction technique or application transcends the prototyping phase and is verified and validated, designers can move the design to more permanent platforms, using standard SDKs and commercial hardware.

Generality

Using five example applications that include both novel and replications of state of the art interaction techniques, we demonstrate the versatility and generality of the toolkit, as well as the expressivity of the building block components [31]. The fundamental limitation but also strength of this toolkit is that it is based around a watch prototyping emulator and not a real – and wireless – watch. We argue that for the rapid prototyping and creative stage of the design process this is an acceptable trade-off. It brings the advantage that developers are not bound by existing hardware limitations or existing device designs, but can design and use their own setup to develop compelling and forward looking cross-device interaction techniques. We expect that in the near future more accessible smartwatch hardware platforms will emerge and future versions of the toolkit could support some of these smartwatches for prototyping. Furthermore, an in-depth testing of the toolkit and its expressive power with developers can provide us further insights into the prototyping process with smartwatch cross-device applications.

CONCLUSION

WatchConnect allows for rapid prototyping of smartwatch-centric cross-device applications and interaction techniques, using custom hardware designs and a software framework that removes complex machine learning, sensor fusion and hardware management into high level objects and events that are integrated with an existing drag and drop UI framework. The toolkit reduces development complexity and lowers the threshold for developers to design for complex device ecologies using a smartwatch as mediating instrument. The toolkit allows for future explorations of a wide range of novel multi-device interaction techniques, hardware designs and collaborative multi-surface environments. Future work includes integrating *WatchConnect* with other cross-device toolkits and smartwatch platforms to support existing frameworks and a wider set of setups and development platforms.

ACKNOWLEDGMENTS

This work was supported by the EU Marie Curie Network iCareNet under grant number 264738 and ICRI Cities. Thanks to Michael Nebeling, Sarah Gallacher and our anonymous reviewers for their feedback and helpful suggestions for the improvement of the manuscript.

REFERENCES

1. Ashbrook, D., Lyons, K., and Starner, T. An investigation into round touchscreen wristwatch interaction. *Proc. of ACM MobileHCI'08*.
2. Badam, S.K. and Elmqvist, N. PolyChrome: A Cross-Device Framework for Collaborative Web Visualization. *Proc. of ACM ITS'14*.
3. Baudisch, P. and Chu, G. Back-of-device interaction allows creating very small touch devices. *Proc. of ACM CHI'09*.
4. Blasko, G. and Feiner, S. An interaction system for watch computers using tactile guidance and bidirectional segmented strokes. *Proc. of IEEE ISWC 2004*.

5. Butler, A., Izadi, S., and Hodges, S. SideSight: multi-touch interaction around small devices. *Proc. of ACM UIST'08*.
6. Cao, X. and Balakrishnan, R. Interacting with dynamically defined information spaces using a handheld projector and a pen. *Proc. of ACM UIST'09*.
7. Chen, X., Grossman, T., Wigdor, D.J., and Fitzmaurice, G. Duet: exploring joint interactions on a smart phone and a smart watch. *Proc. of ACM CHI'14*.
8. Funk, M., Sahami, A., Henze, N., and Schmidt, A. Using a touch-sensitive wristband for text entry on smart watches. *ACM CHI'14 EA*.
9. Greenberg, S. Toolkits and interface creativity. *Multimedia Tools and Applications*, (2007).
10. Hamilton, P. and Wigdor, D.J. Conductor: enabling and understanding cross-device interaction. *Proc. of ACM CHI'14*.
11. Hardy, R. and Rukzio, E. Touch & interact: touch-based interaction of mobile phones with displays. *Proc. of ACM MobileHCI'08*.
12. Harrison, C. and Hudson, S.E. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. *Proc. of ACM UIST'09*.
13. Harrison, C., Schwarz, J., and Hudson, S.E. TapSense: enhancing finger interaction on touch surfaces. *Proc. of ACM UIST'11*.
14. Hartmann, B., Beaudouin-Lafon, M., and Mackay, W.E. HydraScope: creating multi-surface meta-applications through view synchronization and input multiplexing. *Proc. of ACM PerDis'13*.
15. Heikkinen, T., Goncalves, J., Kostakos, V., Elhart, I., and Ojala, T. Tandem Browsing Toolkit: Distributed Multi-Display Interfaces with Web Technologies. *Proc. of ACM PerDis'14*.
16. Hinckley, K., Ramos, G., Guimbretiere, F., Baudisch, P., and Smith, M. Stitching: pen gestures that span multiple displays. *Proc. of ACM AVI'04*.
17. Hinckley, K. Synchronous gestures for multiple persons and computers. *Proc. of ACM UIST'03*.
18. Huang, D.-Y., Tsai, M.-C., Tung, Y.-C., et al. TouchSense: expanding touchscreen input vocabulary using different areas of users' finger pads. *Proc. of ACM CHI'14*.
19. Kim, J., He, J., Lyons, K., and Starner, T. The gesture watch: A wireless contact-free gesture based wrist interface. *Proc. of IEEE ISWC'07,*.
20. Knibbe, J., Martinez Plasencia, D., Bainbridge, C., et al. Extending interaction for smart watches: enabling bimanual around device control. *ACM CHI'14 EA*.
21. Laput, G., Xiao, R., Chen, X., Hudson, S.E., and Harrison, C. Skin buttons: cheap, small, low-powered and clickable fixed-icon laser projectors. *Proc. of ACM UIST'14*.
22. Lyons, K., Nguyen, D., Ashbrook, D., and White, S. Facet: a multi-segment wrist worn system. *Proc. of ACM UIST'12*.
23. Marquardt, N., Ballendat, T., Boring, S., Greenberg, S., and Hinckley, K. Gradual engagement: facilitating information exchange between digital devices as a function of proximity. *Proc. of ACM ITS'12*.
24. Mayer, S. and Sörös, G. User Interface Beaming - Seamless Interaction with Smart Things using Personal Wearable Computers". *Proc. of IEEE BSN 2014*.
25. Merrill, D., Kalanithi, J., and Maes, P. Siftables: towards sensor network user interfaces. *Proc. of ACM TEI'07*.
26. Myers, B., Hudson, S.E., and Pausch, R. Past, present, and future of user interface software tools. *TOCHI '00*.
27. Nebeling, M., Mintsi, T., Husmann, M., and Norrie, M. Interactive development of cross-device user interfaces. *Proc. of ACM CHI'14*.
28. Nebeling, M., Teunissen, E., Husmann, M., and Norrie, M.C. XDKinect: development framework for cross-device interaction using kinect. *Proc. of ACM EICS'14*.
29. Oakley, I. and Lee, D. Interaction on the edge: offset sensing for small devices. *Proc. of ACM CHI'14*.
30. Olberding, S., Yeo, K.P., Nanayakkara, S., and Steimle, J. AugmentedForearm: exploring the design space of a display-enhanced forearm. *Proc. of ACM AH'13*.
31. Olsen Jr, D.R. Evaluating user interface systems research. *Proc. of ACM UIST'07*.
32. Pasquero, J., Stobbe, S.J., and Stonehouse, N. A haptic wristwatch for eyes-free interactions. *Proc. of ACM CHI'11*.
33. Perrault, S.T., Lecolinet, E., Eagan, J., and Guiard, Y. Watchit: simple gestures and eyes-free interaction for wristwatches and bracelets. *Proc. of ACM CHI'13*.
34. Rekimoto, J. Pick-and-drop: a direct manipulation technique for multiple computer environments. *Proc. of ACM UIST'97*.
35. Rekimoto, J. Gesturewrist and gesturepad: Unobtrusive wearable interaction devices. *Proc. of IEEE ISWC'01*.
36. Schmidt, D., Chehimi, F., Rukzio, E., and Gellersen, H. PhoneTouch: a technique for direct phone interaction on surfaces. *Proc. of ACM UIST'10*.
37. Schmidt, D., Seifert, J., Rukzio, E., and Gellersen, H. A cross-device interaction style for mobiles and surfaces. *Proc. of ACM DIS'12*.
38. Xiao, R., Laput, G., and Harrison, C. Expanding the input expressivity of smartwatches with mechanical pan, twist, tilt and click. *Proc. of ACM CHI'14*.
39. Yang, J. and Wigdor, D. Panelrama: enabling easy specification of cross-device web applications. *Proc. of ACM CHI'14*.
40. Von Zadow, U., Büschel, W., Langner, R., and Dachselt, Raimund. SleeD: Using a Sleeve Display to Interact with Touch-sensitive Display Walls. *Proc. of ACM ITS'14*.
41. Pebble. http://developer.getpebble.com.
42. Sony Watch. http://developer.sony.com/.
43. Apple Watch. http://apple.com/watch/.
44. Phidgets. http://phidgets.com.
45. Arduino. http://arduino.cc.

Paper 6: HyPR Device

Title of paper

HyPR Device: Mobile Support for Hybrid Patient Records

Authors:

Steven Houben, *Mads Frost and Jakob E. Bardram*

Published:

*In Proceedings of the 2014 ACM international conference on Interactive tabletops and surfaces **(ITS '14)**. ACM, 2014.*

Abstract:

The patient record is one of the central artifacts in medical work that is used to organize, communicate and coordinate important information related to patient care. In many hospitals a double record consisting of an electronic and paper part is maintained. However, this practice introduces a number of configuration problems related to finding, using and managing the paper and electronic patient record. This paper describes the exploration into the Hybrid Patient Record (HyPR) concept. Based on three design requirements derived from a field study, followed by a design study using a technology probe, we introduce the HyPR device, a device that merges the paper and electronic patient record into one system. We provide preliminary results on a clinical simulation with 8 clinicians resulting in a reflection on the functional, design and infrastructural requirements of Hybrid Patient Records. Our study suggests that the HyPR device decreases configuration work, supports mobility in clinical work and increases awareness on patient data.

HyPR Device: Mobile Support for Hybrid Patient Records

Steven Houben, Mads Frost, Jakob E. Bardram
The Pervasive Interaction Technology Laboratory
IT University of Copenhagen, Rued Langgaardsvej 7, DK-2300 Copenhagen, DK
{shou, madsf, bardram}@itu.dk

ABSTRACT

The patient record is one of the central artifacts in medical work that is used to organize, communicate and coordinate important information related to patient care. In many hospitals a double record consisting of an electronic and paper part is maintained. This practice introduces a number of configuration problems related to finding, using and aligning the paper and electronic patient record. In this paper, we describe the exploration into the Hybrid Patient Record (HyPR) concept. Based on design requirements derived from a field study, followed by a design study using a technology probe, we introduce the *HyPR Device*, a device that merges the paper and electronic patient record into one system. We provide results from a clinical simulation with eight clinicians and discuss the functional, design and infrastructural requirements of such hybrid patient records. Our study suggests that the HyPR device decreases configuration work, supports mobility in clinical work and increases awareness on patient data.

Author Keywords

Hybrid Patient Record; Electronic Health Record; HyPR; EHR; Nomadic Work; Hospitals

ACM Classification Keywords

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces; J.3 [**Computer Applications**]: Life and Medical Science.

INTRODUCTION

The patient record is one of the most important artifacts in medical work in hospitals as it is used as a central legal document to organize patient data, communicate relevant information with other clinicians and departments, and coordinate complex patient treatment procedures. In recent years, the Electronic Health Record (EHR) has been introduced in an effort to provide a higher level of quality in healthcare through a more efficient, safer and unified workflow. EHRs have a number of important advantages over traditional paper records, including a higher degree of security, simpler workflows, standardized documentation and more accurate and widely available access to patient data [26].

ITS 2014, November 16–19, 2014, Dresden, Germany.

ACM 978-1-4503-2587-5/14/11 ...$15.00.
http://dx.doi.org/10.1145/2669485.2669502

Figure 1. A Hybrid Patient Record (HyPR) device augments the paper patient record with color configuration and location tracking, while allowing clinicians to pair a tablet which shows the digital information associated with the paper record.

However, by attempting to replace paper records with an electronic counterpart, the affordances (such as handleability, manipulability and portability [15]) of paper-based interaction are removed. In many hospitals, the paper record is therefore still actively used as a central artifact for day to day work, despite the widespread deployment of EHR systems [25, 33]. These paper records are also frequently used for the storage of more informal documentation such as e.g., nursing notes or other working records, which again adds to their importance. Prior studies have even shown that paper helped some clinicians to be more efficient in their work [21]. Consequently, a typical setup in many hospitals is that the EHR system does not replace the paper-based record, but instead a *double record* consisting of both an electronic and paper part is maintained.

This double medical record introduces a number of configuration problems related to finding, using, and managing both the paper and electronic representation of the patient record. First, the usage of both electronic and paper records causes synchronization problems between both representations [27], forcing clinicians to deal with the paper and digital information simultaneous. Since digital information is often only available through desktop computers, it requires clinicians to sit at a desk when interacting with patient data. Second, since many hospitals require the use of a unique paper record, it is often transferred between different departments and wards as patients and clinicians move throughout the hospital. This causes paper records to be physically misplaced in the ward or even lost between departments resulting in clinical staff tracking down the record.

To mitigate these configuration problems, we introduce the Hybrid Patient Record (HyPR) device as shown in Figure 1.

The paper-based medical record can be augmented with the HyPR device, which supports notifications (color and sound), location tracking, and provides an easy way to link to the electronic patient record. This paper reports on the user-centered design and implementation of the HyPR device and presents four contributions: (i) we propose three design principles for hybrid patient records that are derived from a field study; (ii) we report on a design study, in which ten clinicians participated in a design walkthrough of a technology probe; (iii) we describe the design and technical implementation of a HyPR device and supporting infrastructure; and (iv) we analyze the preliminary results of a clinical simulation of the HyPR device.

RELATED WORK

A large body of research has explored the connection or linking of paper to digital data. One of the earliest approaches is DigitalDesk [29], an interactive desk that adds electronic features to physical paper and physical attributes to digital information using a top mounted camera and projector. Inspired by this work, the Paperlink [2] system attempted to scale this approach down by using a portable video pen. Based on this idea of a digitized pen (and commercial versions such as Anoto), other approaches such as the *Paper Augmented Digital Documents* (PADDs) [7], Paperproof [28] and PapierCraft [13] provide support for digital annotation of paper documents using pen gestures.

To increase support for mobility, several approaches proposed to use PDAs or phones as mediator between paper and digital information. Circa [9] proposes the "paper PDA" concept in which they use the *StickerLink* approach to link paper and digital information. A-book [16] overlays physical notes with a PDA and Pacer [14] uses a phone to link paper documents to digital versions based on visual features. Prism [24], S-Notebook [19], and ButterflyNet [31] further explored the design and impact of hybrid approaches in which paper notes and digital counterparts were linked together. Finally, a number of approaches explored the creation of broad hybrid workspaces, that included support for paper documents. Magictouch [18] is an early approach that uses RFID technology to detect the location of physical documents in a defined space. The Designers' Outpost [11] recognizes paper documents using a rear camera. IdeaVis [6] provides a hybrid brainstorm space by augmenting paper documents with interactive zones. Finally, Penbook [30] is a hybrid approach providing a touch screen together with a built-in projector integrated with a wireless pen to support handwriting for prescriptions or patient registration in hospitals.

A number of approaches have explored different ways to augment or enhance the medical record. Rodriguez et al. [20] demonstrated a location-aware information system for medical work, which estimates the clinicians' location to find available patient data and display it on a mobile device. Similarly, the MobileWard system [23] provides a context-aware mobile patient record system for hospital wards aimed at supporting autonomous adoption to the changing tasks or location of the nurses. A more radical approach to context-aware computing in hospitals is presented in the Activity-Based Computing project [3], which proposes a new paradigm for context-aware information access and collaboration in patient wards, using *activity* as a central construct. The augmented paper chart [32] augments a single paper chart with an Anoto interface, thus supporting seamless integration of traditional paper-based notes and digital storage. However, despite the fact that several studies point to the importance of the physical paper record, it has received remarkable little attention in the development of these new pervasive interactive patient record and information systems.

In general, prior work has primarily focused on ad hoc tracking and local linking of single paper documents to their electronic representation. But as pointed out by many studies of hospital work, medical workflow is highly nomadic and collaborative [3, 25]. The nomadic nature of clinical work puts forward a set of fundamental challenges in terms of locating and tracking artifacts, while the collaborative nature implies that support for exchanging and sharing artifacts, material, resources, and devices should be part of the system design. Compared to prior work, the core contribution of this paper is the physical/digital integration of the entire medical record in the nomadic and collaborative work setting of a hospital, thus complementing per-document approaches. The novelty of the HyPR device is thus its unique attempt to connect and align *the entire medical record* using a mediating sensor platform.

FIELD STUDY

To understand in depth how paper and electronic patient records are used, we conducted a field study. Over a period of two months, we studied five different medical departments, covering two patient bed wards, two surgical departments, and the emergency department. We performed task-centric, artifact-centric, and place-centric observations, contextual inquiries through shadowing of nurses, and post-hoc interviews.

The medical record

The hospital in this study uses one unique paper-based medical record for each admitted patient. It is a legal requirement that this patient record is present at the ward or department that is treating the patient. The paper record is made of a plastic binder with explicit color-coded sections for patient data, continuation (treatment history), nursing notes, various schemes and forms, observations, test results (e.g., blood tests and radiology examinations), and correspondence with other medical professionals. On the front, the binder has a label with the patient's name and ID written both in text as well as encoded in a bar code. On average the patient record is between 2 and 3 cm thick.

In parallel to the paper record, the hospital uses a number of specialized health information systems, such as radiology, medication, patient administration, and blood bank systems. Access to these systems have been collated in a portal, which is referred to as the electronic medical record. The paper and electronic medical records are used simultaneously in patient treatment and are equally important for medical work. Most information is duplicated in both records, whereas other information only exists in one or the other. This creates significant synchronization problems between the two versions of the records, which again leads to extraordinary work in manual updating, verification, and cross-referencing. For example, a lot of work is put into printing from the electronic medical record and storing print-outs in the paper records. This leads to significant problems of updating and replacing the printed documents in the paper record, when information changes in one of the electronic systems.

The different health information systems are primarily used to request or create new medical information, such as ordering blood tests at the hospital lab. The physical medical record, on the other hand, is primarily used to archive patient information. Because lab results e.g., need to be put into the paper-based record, the lab system is configured in such a way that when a lab result is ready, it is sent directly to the requesting ward's printer. In this way, the test results are physically presented and the printer becomes a coordinative artifact that signals when test results are ready. At a patient ward, there are typically up to 25 records of active patients. But since records from dismissed patients are stored at the ward, hundreds of archived records are at the department. Finding the right paper record is challenging as there is no visual differentiation between records; they are all stacked upon each other and scattered all over the ward in the nursing station, the secretary offices, and in the archiving room.

Patient Record in Nomadic Work

Medical work in hospitals is inherently nomadic [4], which implies that clinicians and the tools they use (including the patient record) move around inside wards, departments, and the entire hospital. The paper records are mostly used in offices, nursing stations, doctors' offices, and at the bedside of the patient. As mentioned earlier, it is a legal requirement that the record is present during medical treatment, which implies that the record always 'travels with the patient'. For example, when patients are send to other wards (e.g., for x-ray or surgery), the record is mounted in a special container on the side of the patient bed and travels with the patient to the receiving department. Moving the record around inside the hospital again causes it to get lost or misplaced both inside the ward and in other departments.

HYBRID PATIENT RECORD CONCEPT

Prior research on medical work (e.g., [5, 17]) and our field study have identified a range of challenges associated with handling medical records. At its core, these challenges are tied to clinicians' need to handle, align and coordinate physical and digital information simultaneously. One way of approaching this challenge is to digitize all information in medical work – a strategy that is being pursued in the creation of integrated electronic medical records (EMR) and hospital information systems (HIS). However, several studies (including [17, 25, 33]) show that despite the 'successfulness' of the deployment of EMRs, paper documentation, artifacts and records are still widely used in documentation as transitional artifact [5] or as redundant information source in medical work. As such, the findings from the medical domain back up findings from the office environment about the 'myth of the paperless office' [22]. Therefore, rather than designing for the 'paperless hospital', there is a need to design for the parallel management of both paper and electronic medical records, thereby creating a *hybrid* medical record. Inspired by our field study and prior work, we propose the following three principles for the design of patient records.

D1 Dual Use – Because the paper and electronic version of the record are almost always used simultaneously, setting up and removing the connection between the paper record and a device representing the electronic record should be instantly and easy. Both representations should be usable separately, without any changes to their original purpose or use. Since the paper record is used to identify the patient case, the hybrid record should use this patient context to load and visualize the correct data. To facilitate the usage of the double record, it should be integrated with existing practices, devices and technology.

D2 Recognizability – To support easy identification and recognition of a patient record (e.g., in a cluttered office space) the patient record should be able to relay and display various kinds of status and awareness information. Temporal visual and auditory cues (similar to the analogue affordance of e.g., sticky notes) should be supported to provide clinicians with an easy and fast configuration mechanism for self-reflection or coordination with other clinicians.

D3 Mobility – The patient record should support the nomadic workflow in hospitals, meaning that both the electronic and paper representation of patient data should be available in a portable and traceable form factor. To support clinicians in finding and managing the location of the record, the supporting infrastructure should support location tracking and remote access to the state of the paper record. Additionally, the location should be used to ease information retrieval.

To address and support dual use, recognizability and mobility in patient records, we propose the concept of a *Hybrid Patient Record* (HyPR). Conceptually, a HyPR setup consists of three parts: (i) the traditional paper patient record as used in hospitals today, (ii) the electronic record accessed from a tablet or phone, and (iii) a mediating platform that augments the paper record with a number of configurable properties and connects the paper record to the digital record on the tablet. The central purpose of this concept is to integrate the electronic patient record into the existing physical and mobile workflow of clinicians. By explicitly attaching digital information and notification systems to the existing paper record, the HyPR presents clinicians with a patient record that encapsulates existing practices but augments it with digital capabilities. In summary, the HyPR device allows for ad hoc integration of paper-based and digital information, authorized and fast access to digital information, customization of the record using the sensing platform, and traceability by location tracking.

DESIGN STUDY

To explore the feasibility of the HyPR device concept and to get a better understanding of the design and clinical implications of hybrid devices, we conducted a design study involving a group of clinicians from two different hospitals. The goal of this study was to get feedback on the design of a technology probe and use this as input for the design and implementation of the device. The study had two parts. First we introduced the concept of hybrid patient records to the clinicians in order to open up a discussion and brainstorm on the design dimensions and implications of the HyPR devices. Second, based on the use of a concrete prototype, we asked for detailed input on the perceived usefulness of the HyPR device in clinical work and the usefulness of its different features.

Technology Probe

Because it is often hard for clinicians to envision how they could benefit from technology, we performed a design walk-through on a fully working prototype. The technology probe was designed as an augmented hard-cover box with room for both the digital and physical paper version (Figure 2). To bridge the size mismatch between modern tablets and the paper record, the enclosure provides a dock for the tablet (Figure 2 B) and a slot for the paper record (Figure 2 C). The slot on the side allows for easy access and pushes the record together so it does not fall out while moving. The tablet dock is specifically designed so clinicians can securely mount their device, while still being able to use it. When interacting with both the paper and digital patient data, clinicians can simply remove the paper record from the slot and browse the paper and tablet data at the same time. The device is activated by inserting a patient record in the slot and mounting a tablet to the dock (Figure 2 A). The color and sound of the device can be controlled by using the application on the tablet. The enclosure thus creates a temporal connection between the paper record and tablet.

Study Setup

In total 10 clinicians (all female, mean age = 42, σ = 5,37) from two different wards participated in two separate design sessions. The first session included three clinicians from a surgical ward that the original field study discussed earlier, while the second session included 7 clinicians from the psychiatric ward of a different hospital that was not part of the prior field study. Participants included two doctors, a psychologist, a clinical specialist, a medical secretary and five nurses. All participants were highly experienced in day to day medical work in patient wards and rated themselves as average computer users ($\tilde{x}$= 3; *iqr*= 0). The design sessions were done *in situ* at the hospital ward of the participants.

Method

The design study consisted of three phases. First, participants were introduced to the concept of a hybrid patient record through a demonstration of the functionality of the technology probe. The introduction used a number of *scenarios* that were designed based on the field study discussed earlier and validated by the head nurse from the ward. After the introduction, a semi-structured interview and discussion session was initiated to allow the clinicians to provide feedback on the scenarios and the design of the technology probe. The data from the sessions were collected using audio and video recordings, note taking and pictures. After the semi-structured interviews, participants completed a 5-point Likert scale questionnaire, which was used to discuss the design and functionality of the technology probe.

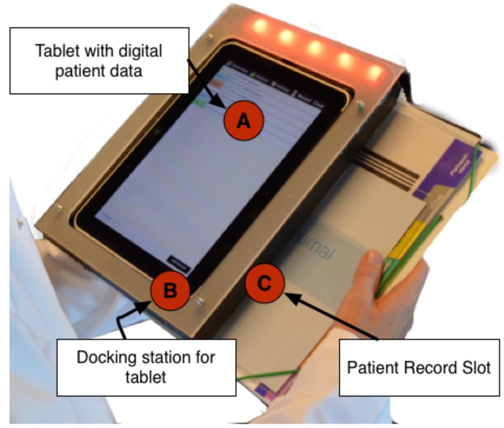


Figure 2. A fully working industrial prototype used as technology probe.

Results

In general, clinicians found that the HyPR device would be useful in clinical work. Although hospitals are trying to implement the vision of a 'paperless workflow', many clinicians realize that there are limitations to this vision, since much paper-based information does not exist in the digital world. This lack of one-to-one documentation between paper and digital information also greatly limits options to digitally augment individual records with e.g., Anoto technology. The basic functionality of being able to pair a paper-based and electronic record was considered as very useful. The contextualization of the visualized patient data, based on the paper record that was paired with the probe, was deemed as very important:

> – *"One of the key things of this device is the easiness of accessing patient data by getting rid of fixed computers and looking up patient data by simply placing the device on top of the paper record. That would save us a lot of time, and make the workflow a lot easier." – P3*

This instant and ad hoc pairing ability was especially valued for emergency cases in which patients are admitted with an acute problem:

> – *"Often when I have to treat acute patients – which I don't know in advance – this device would make it a lot easier to pair the paper record and the data in the electronic record. And I can do it anywhere and not only in front of my PC." – P8*

The ability to add color to the medical record was perceived as very useful. Clinicians mentioned that color is often used as a general coordination mechanism at the ward. For example, colored post-it notes are often used to indicate status information on patients. Tracking the device, thus locating the patient record was also considered very valuable. Clinicians responded that this would save them a lot of time and energy in finding the paper record, which can be stored anywhere, even *"under a pillow or in a drawer" (P1)*. As explained by one clinician:

> – *"Not only tracking the record in the system is useful, but also providing visible and audible feedback which makes the record easier to find when you know what*

room it is in – just like the key finder gadgets where you can whistle and it then makes a sound". – P2

Although the working area and input/output bandwidth is significantly lower than desktop computers, all clinicians preferred to use a tablet over a PC for all day to day medical work. In general, the inclusion of a portable tablet into the system setup was considered an improvement for clinical workflow.

However, there was a general consensus that the size of the system – in this case the docking station – should be designed to fit into the pocket of a standard white coat. Several clinicians argued that the mediating device (i.e the docking station for the tablet) should be much more closely integrated with the physical folder of the paper record, arguing that the current HyPR device should somehow be merged into the paper record:

– "The idea of having a device that communicates with the paper journal, and using the journal to retrieve information on the device is super. However, I think the device as a separate object makes it laborious, and does not really fit the current work practice. If the clinician had his own tablet, which could interact with the patient record by simply placing it on top, would make it a lot more useful". – P5

Finally, clinicians in general argued that since the patient record follows the patient throughout the hospital, the device should be usable in different types of wards and clinical conditions, such as in operating rooms or the x-ray department.

Summary

Clinicians first of all argued that most of the functionality provided by the mediating enclosure of the technology probe, such as *location tracking* and *visual cues*, would also be useful once the patient record is archived and thus no longer active at the ward. These requirements extend the concept of the HyPR device from an ad hoc temporal mediator for active patient cases to a permanent augmentation and deep physical integration with the paper record. Second, clinicians stressed the importance of the HyPR device's ability to cope with the medical environment. The device should e.g., be strong enough to survive being dropped; interaction with the device should be possible while wearing latex gloves; and clinicians should be able to sterilize the device. Furthermore, the device should be constructed from food-safe plastic. Finally, clinicians generally argued that having the tablet physically docked to the record would not support their work practices very well. The general consensus was that the HyPR device should be usable with different size tablets and even phones, and that pairing the tablet to the device should be faster.

HYBRID PATIENT RECORD DEVICE

Based on the design principles and the results from the design study, we constructed the *Hybrid Patient Record* (HyPR) device as shown in Figure 1, 3, and 4. The HyPR device and its underlying infrastructure (Figure 7) are designed to integrate paper-based and digital patient information into the nomadic workflow of clinicians. The HyPR device supports *dual use* by allowing for ad hoc pairing between the paper and digital information. And, it provides clinicians with a mechanism to dynamically change some of the physical properties (color and sound) associated with the record. Finally, the device is equipped with a location tracker to allow clinicians to easily find the paper record. The HyPR device works within a larger infrastructure that supports location tracking, device management, and access to the electronic medical systems.

Figure 3. Two clinicians interacting with a number of HyPR devices scattered in the patient ward.

Design

The HyPR (Figure 3) consists of three distinct parts: (i) a traditional paper record, (ii) a tablet used to access the electronic record and (iii) a HyPR device. The HyPR device is a rectangular plastic plate with the same width and height as the paper record. Compared to the technology probe, it does not have any layers as all the electronics are integrated in the plate. The paper record is attached to the plastic plate using metal clips to create a permanent connection to the device. In contrast to the technology probe in which an active patient record is loaded into the device by inserting it into the slot, this version of the HyPR device is a permanent augmentation of the record. This means that the HyPR device becomes an inherent and inseparable part of the paper record.

Functionality

The HyPR device provides three features: (i) *pairing* of the tablet and the paper record using proximity sensing, (ii) *light and sound* system that can be used to augment the record or notify other clinicians, and (iii) an integrated *location tracking* unit that allows clinicians to locate the record.

Clinicians can interact with the HyPR by placing a tablet on top of the paper record and HyPR device. By doing so, they pair the tablet to the record, causing the underlying infrastructure to fetch the digital patient information and push this to the active view of the tablet. Any changes made to the electronic record are immediately propagated through the infrastructure and synchronized with any other paired devices. This process essentially eliminates extra configuration work during e.g., ward rounds or during emergency situations in which manually fetching information would be too time consuming or inappropriate. The device is thus used as a proxy that provides clinicians access to the activity of the patient. Although the initial pairing process is done by proximity, both the tablet

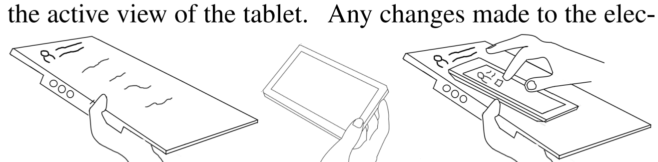


Figure 4. The color of the HyPR device can be configured to signal a wide range of things. For example, the colors can represent a specific nurse, patient status, or simply be used to highlight a patient record in an information dense environment.

and paper record can be used separately. When a user removes the tablet from the HyPR device, the data will remain coupled to the initial paper record until the user manually selects another patient, or pairs the tablet with another HyPR device. Furthermore, clinicians can also remotely connect to the record by selecting the patient case from the application. This allows multiple clinicians to work on the same patient case, while only having one physical journal. Only the paper record – not the tablet computer – is uniquely coupled to a HyPR device to ensure the infrastructure can correctly track and manage each record. (Supporting D1: Dual Use)

The HyPR device supports concurrent use and updates of both paper-based and digital information. For example, administration of medication in the medicine system can be done directly in the electronic medical record via the tablet computer. Similarly, ordering of lab tests can be done electronically by accessing the order-entry system. Simultaneously, paper-based information can be accessed from the paper-based record and ad hoc written notes can be added and stored temporarily in the physical folder. Moreover, electronic information – such as the lab results coming out of the printer – can be added in paper format to the folder. As such, the HyPR record supports blending paper-based and digital information in 'both directions'. (Supporting D1: Dual Use)

Once the device is paired, clinicians can change the physical properties of the HyPR record by changing its color scheme or identification sound. Figure 4 shows a number of different color configurations. These configurations can be used to relay status information. For example, a color can be associated with a specific nurse, thereby revealing who is the contact nurse for a specific patient. Or a color can represent a status change, by e.g., highlighting that there is a lab test result available for the patient. Moreover, sound and/or color can help locate records, which may be scattered all over the department. When the record is located in a cupboard or drawer, sound can be used to draw attention to the record. (Supporting D2: Recognizability)

The HyPR device supports nomadic medical work in several ways. First, in order to support location of medical records, the HyPR device is equipped with a location tag that broadcasts a unique value. This value is associated to a particular paper record, when the HyPR device registers the patient ID. Clinicians can look up the location of each patient record. Second, to minimize the burden of carrying both the augmented record and tablet, microsuction tape is attached to the front of the paper record to keep the tablet in place. Finally, the tracking capabilities of the HyPR device can be used to contextualize the patient's information. For example, if the HyPR device is taken to the patient's bed side (e.g., as part of a ward round), basic patient information and the latest entry in the record is shown, whereas the patient medicine treatment is shown if a nurse takes the HyPR record to the medicine room. (Supporting D3: Mobility)

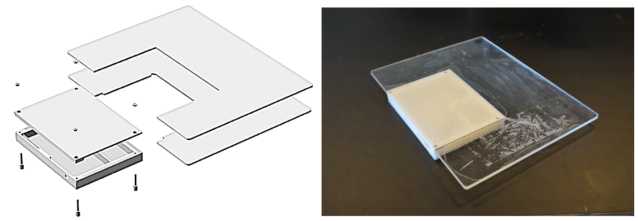


Figure 5. The physical parts of the HyPR device.

Technical Implementation

Figure 5 shows the design of the HyPR device. It consists of two different parts: (i) a rectangular plate of 2.5 mm food-safe transparent plastic, and (ii) an enclosure holding the electronics embedded into the side of the plate. Figure 6 shows the electronic architecture, which uses an Arduino ATmega168 chip 16 MHz crystal for basic processing; a RFID module with an antenna (125 kHz); a Texas Wifi CC 3000 module with antenna; an array of three high power RGB LEDs; a 2kHz range buzzer; an integrated rechargeable Volt battery pack with USB connector; a power switch; and a 35–45 kHz ultrasound tag with a dedicated 3V lithium battery.

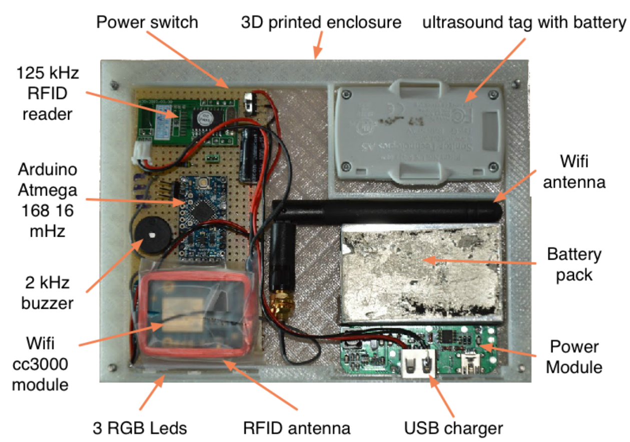


Figure 6. The electronics of the HyPR device.

To support two way communication between the tablet and the HyPR device, the firmware provides support for a custom protocol with a set of command messages. One set of messages allows the device to start a handshaking protocol when a tablet is paired and to send 'alive messages' that indicate that it is operating correctly. Other command messages allow the tablet to operate and configure the device's on-board buzzer and the RGB LED array based on e.g., user input or infrastructure changes. The RFID module continuously reads all nearby RFID tags and sends tag IDs to the Arduino board.

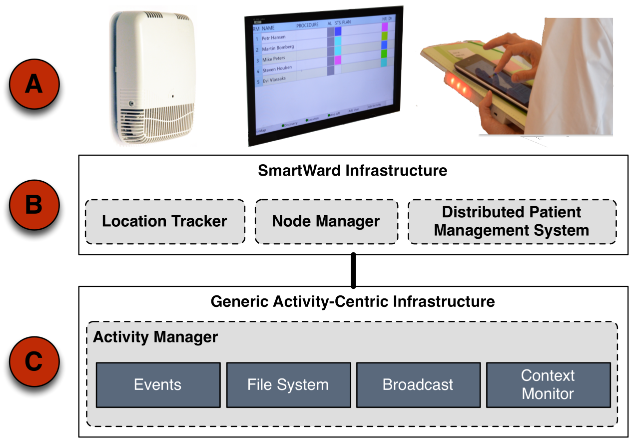


Figure 7. The *HyPR infrastructure* (B) is a distributed context-aware infrastructure designed to support multi-device location-aware collaborative workflows in patient wards. The infrastructure uses an ultrasound location tracker for location-aware services (A) and is built on top of a general purpose activity-centric infrastructure [10] (C).

When a new tag is detected, the device sends a message over Wifi to the infrastructure, which then pushes the data to the tablet. Similarly, any changes made on one of the paired tablets is send over Wifi to the HyPR device.

Infrastructure

Figure 7 shows the HyPR device infrastructure, which is an activity-aware patient management and information system designed to support multi-device location-aware collaborative workflows in patient wards. The infrastructure supports (i) large interactive screens for shared collaborative workspaces, (ii) tablet applications for mobile personalized tasks and detailed patient information, and (iii) desktop systems for integration with existing applications and services. The HyPR infrastructure is build on top of a generic distributed activity-centric infrastructure (detailed in [10]) that includes support for multi-device information management, context-awareness and ad hoc discovery and pairing of devices.

The infrastructure abstracts basic events, data, pairing, discovery and context services into a distributed *activity configuration*. These configurations connect all patient-related information resources, users and devices into one central reusable data model. This model is managed and distributed across all devices that are part of the same activity systems [10]. Devices such as tablets, pc computers, large displays are thus interconnected into one ad hoc distributed activity system, in which patient information is managed, synchronized and distributed as computational activity configurations. The hardware inside the HyPR device also connects to the infrastructure over Wifi and reports which tablet is detected. The infrastructure uses this information to push the right patient data to the paired tablet, or to update the properties of the HyPR device made through any of the connected devices.

Application

The HyPR application running on the tablet is a web-based stripped down electronic patient record that consists of a patient overview screen (Figure 8A) and a detailed patient record (Figure 8B). In the overview screen, all patients that are currently at the ward are listed with basic information including their name, medical procedure, assigned color and room number. Using this patient overview, clinicians can set to colored lights of a specific patient record to "blinking", thus asking for attention. Clinicians can also turn on the buzzing sound (which automatically stops after 15 seconds) of the record to quickly locate it when it is in a drawer or on a stack of other records.

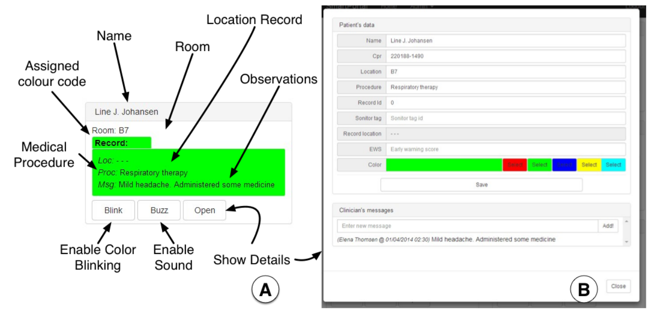


Figure 8. The details of the patient record.

The tablet is synchronized with the paper record through physical proximity. Placing the tablet on top of the paper record, automatically opens the detailed patient information of that patient to the tablet (Figure 8B). This view lists all detailed medical information and allows clinicians to add new medical data or messages. It can also be used to change the colored representation of the patient state. Changing this color in the details view updates the color on the HyPR device (Figure 4). When medical information is added remotely, through another tablet or computing device that is not physically paired to the paper record, the device's colored lights start blinking to signify an update. Once a clinician pairs the tablet, the new data is shown and the record stops blinking.

CLINICAL SIMULATION

To explore how clinicians would use the HyPR setup, we conducted a clinical simulation. Specifically, the study was set up to frame the use of the HyPR record within existing work practice as previously studied in the wards. In the medical domain, a clinical simulation is a frequently applied methodology used to train and educate clinicians in critical clinical scenarios, such as surgery, medicine prescription and administration, and emergency cases. It has proved very efficient and reliable for the initial phase of training and assessment of clinical staff [1]. Since the clinical simulation approach attempts to bring the dimension of clinical context into stronger focus, the method has lately been used also as a method for testing clinical systems with representative users doing representative tasks, in an ecological valid setting [12]. The goal of this simulation was to explore (i) the usefulness and usability of the HyPR device and (ii) the impact of HyPR devices on clinical work practices. Although a full description of the study is beyond the scope of this paper, we presents findings relevant to the system design of the HyPR device.

Study Setup

Over a period of 2 days, 8 clinicians (5 female + 3 male, mean age = 46, σ = 12,95) from three different wards (psychiatry, surgery, and emergency departments) participated in a clinical simulation. Participants included 5 doctors, 2 nurses and a psychologist. All clinicians were highly experienced in day to day medical work that involves managing and interacting with patient records, and rated themselves as experienced computer users ($\tilde{x}$= 4; *iqr*= 1). The clinical simulation was performed in a training–simulation ward that is identical to a full scale patient ward, and included both simulated patients (simulation dolls) as well as one human who was acting as a patient. The ward was organized and equipped as an oncology ward, but the scenarios were generic enough to be performed by clinicians with different clinical backgrounds. The study was conducted by two researchers, performing the roles of facilitator and observer.

Method

The study consisted of three phases. First, participants were introduced to the system and physical layout of the ward. After the introduction, participants were asked to complete three scenarios in pairs. The scenarios (that were based on observed situations from the field study) included (i) dual use, in which clinicians performed a ward round and calculated an early warning score (EWS) for four patients, (ii) recognizability and awareness, in which clinicians coordinated the arrival of a paper blood result, and (iii) mobility, in which clinicians searched for a lost record. Data from the scenario performances were captured by using video and audio recordings. Afterwards, participants completed a 5-point Likert scale questionnaire and an interview was conducted.

Results

Table 1 shows the results of the questionnaire on the usefulness of the final design and its specific features. The results of the questionnaire indicate that clinicians consider the final design of the HyPR device to be usable and useful in clinical work ($\tilde{x}$= 4,5; *iqr*= 1).

HyPR Usefulness (N=8)	*Min*	*Q1*	$\tilde{x}$	*Q3*	*Max*	*Iqr*
In general, the HyPR is useful	3	4	**4,5**	5	5	1
Pairing the PR and ER is useful	3	4	**4**	5	5	1
Color feature is useful	3	3,75	**4**	4,25	5	0,5
Tracking the PR is useful	4	4	**5**	5	5	1

Table 1. The results of the 5-point Likert scale questionnaire on the usefulness of the basic functions of a HyPR device. The table shows the minimum, maximum, median ($\tilde{x}$) and the inter quartile range (iqr) of the scores. PR:paper record; ER:electronic record.

Dual Use

The ability to simply place the tablet on top of the device to get instant access to digital data of a patient, was considered as very useful ($\tilde{x}$= 4; *iqr*= 1). During the ward round, almost all clinicians would immediately pair the device to the patient record, before actually checking the vital signs or talking to the patient. They thus preferred to configure and align both the electronic and paper record before commencing with assessing the patient. After this configuration, clinicians would often detach the tablet from the HyPR device. One clinician would typically hold the paper record to check the official early warning score (EWS) form, while the other clinician would check for messages on the tablet and add the EWS to the electronic record.

Although the feedback on the design of the HyPR device was more positive compared to the technology probe used during the design study, clinicians generally agreed that the device was still too heavy and too thick. In essence, they argued that for this device to be usable on a large scale, it has to be integrated in the paper record, thus being flat and flexible. There were also some issues with detecting the right HyPR device. When clinicians tried to pair the tablet with a HyPR device that was placed on a stack of other devices (e.g., Figure 4), the tablet would sometimes receive incorrect patient data. This opened discussion on security and privacy, as some clinicians mentioned that detailed access to patient data should be restricted to the assigned doctor and nurse. On the other hand, clinicians also saw the HyPR system as an opportunity to increase security. One suggestion was to actually physically lock the paper record to the HyPR enclosure until an authorized tablet is paired. This would ensure that only authorized browsing of the paper record would be possible.

Recognizability and Awareness

In general, most clinicians argued that using color was useful ($\tilde{x}$= 4; *iqr*=0,5) for coordination and communication between staff. During the scenarios, clinicians quickly adapted to using the color coding as part of the workflow. Although none of the clinicians felt that the color coding dictated a patient order for the ward round, most of them argued that having an extra layer of awareness on patient cases could improve coordination at the ward but also could help clinicians to reflect on their work. One doctor, for example, mentioned that the colors improved the structure of his round as they helped him prioritize patients. The blinking light feature when new critical messages were added to the electronic record received mixed responses. Some clinicians felt that using the colored lights on the record as a notification mechanism was very useful as they do not always carry a tablet when doing their work. Without the notification on the record, they felt that they might miss critical information. Other clinicians felt that the blinking colored lights were too distracting. Specially in cases where many records were in the same place, it could quickly escalate in a “Christmas tree”.

During the interviews, the clinicians suggested a number of use cases for the color coding. One theme of suggestions was based around coordination between clinicians. Examples such as triage, allocation of nurses and even to reflect the current state of the patient, were proposed as use cases for dynamic colors. A common argument was that one color did not provide enough granularity to communicate more complex information and communication streams. Most clinicians agreed that more colored light indicators could be added to support more applications. A second theme of suggestions was based around patient involvement. One example proposed by clinicians was to use the lights as a road map or guide for the patient, so they could keep track of the different steps in their procedure. However, some clinicians also argued that using these colored lights might worry or even frighten patients who might be unaware of the significance of the changing color. One clinician suggested that the device should include a ‘silent switch’, that would turn of the visual and auditory notifications.

Mobility

Tracking the patient record was considered as one of the most useful features of the system ($\tilde{x}$= 5; *iqr*=1). During the scenarios, most clinicians would follow a similar pattern in which they would first find the room of the patient and the location of the record. They would then proceed to the location of the record, and if the record was not immediately visible they would start the blinking light. If the record still could not be located, they would turn on sound. Most clinicians argued that this was an important feature as patient records get lost regularly. The color blinking feature was useful to find a record in stacks of other records, but clinicians would mostly use the sound to find the record inside a room. Although some of the records were inside the patient room, we observed that clinicians would still use the sound indicator, even if patients were sleeping in that room. Most clinicians agreed that finding the record was important enough to disturb a patient.

A main point of criticism on the current system was the sound of the buzzer. Some of the clinicians suggested that rather than using "another medical sounding sound", the HyPR device could use radically different sounds such as a singing bird, as this would sound less stressful or disturbing for patients. Clinicians also proposed to use the location tracking capabilities in a more integrated way. Rather than "simply" tracking the record, they suggested to set up more advanced functionality such as e.g., automatically check-in when the patient and record arrive at the ward.

DISCUSSION

Often, the term *paperless workspace* points to a vision of a completely digitized work environment in which paper is replaced by digital devices. However, an increasing body of evidence indicates that despite increased digitization, paper is still an important resource in accomplishing everyday work and collaboration. This is true for office environments [22] but also for medical work in hospitals [17, 25, 33]. The central focus of the HyPR device prototype is to explore the functional and clinical design of an augmented hybrid medical record bridging across both the physical and digital records. Our design study and preliminary evaluation show that clinicians generally agreed that such a hybrid record would significantly improve the existing workflow. As such, the HyPR device could be viewed as part of a solution to two long standing problems in nomadic clinical work [3]: configuration and mobility work.

The HyPR device provides clinicians with a tool to synchronize and merge the paper and contextual digital representation of patient data, which significantly reduces *configuration work*, i.e. the amount of work required to setup a working context for a specific patient. The augmented record becomes an entry point into the digital patient record. Many clinicians argued that automatically loading patient data on the tablet when placed on top of the HyPR device, would significantly reduce this configuration work. The location tracking features of the HyPR device provide clinicians with a spatial coordination tool designed to help reduce *mobility work*. Using the wall-based displays, clinicians can look up and track the physical location of the device throughout the ward and the hospital, and the interface of the tablet provides *location awareness* cues on the location of the patient record.

The HyPR device is designed to support the highly collaborative workflow in hospitals. It supports *user multiplicity* by allowing multiple tablets to be connected to the same HyPR device. This allows multiple clinicians to work simultaneously on the same patient case – some using the paper record and some using the digital counter-part. This feature mimics the way that paper-based records are often shared among clinicians in colocated collaboration (e.g., by the bed side or during a medical conference).

One of the central limitations of the current approach, however, is the deliberate absence of digital support for separate paper documents and forms. Although the current design does not exclude the integration of Anoto or similar pens to automatically digitize written notes and forms, this would require a substantial change of existing hospital work practices. The paper forms and electronic records simply do not align one to one, thus posing fundamental questions on how these tools can be integrated and how they would effect work practices. Furthermore, our study showed that support for paper documents should in particular also incorporate support for handling legacy documentation for both legal and practical reasons. The HyPR provides support to *align*, not to *integrate*, information from the paper and digital records into one system. As such, the HyPR concept allows for a fluent and gradual approach to digitizing the entire medical work.

The current HyPR design still places a large emphasis on the use of a physical paper record, which is augmented for easy connections to the digital workflow. The form factor of the HyPR device leverages the shape of the paper record. This implies that at the cost of the weight of the back plate and electronics, the HyPR provides affordances (such as *flexibility*, *markability*, *portability*, and *accessibility* [8]) that are very similar to those of the paper record by itself. This allows clinicians to manipulate and use the HyPR in the exact same way as the paper record, thus embracing a number of existing practices. As with many electronic devices, the battery life of the device often limits its full potential. The current design of the HyPR allows for up to 6 hours of continuous use but includes a standard USB connector for easy recharging. However, for real long term deployment, this would not be a workable solution. Although battery use can be greatly optimized, digitizing mobile patient records will require the careful design of charging strategies that are mobile and fit into the existing workflows of clinicians that use the record.

The notion of a Hybrid Patient Record opens up a number of interesting questions for future work. HyPR devices could be augmented to support complex multi-device interactions including interactive whiteboards or desktop computers. Additionally, the role and design of printers can be re-thought: a printer could e.g., only print patient results when the record is physically moved into the nurse's station. Finally, the physical design of the HyPR device could consider a smaller and more flexible form factor embedded into the patient record binder.

CONCLUSION

In this paper, we introduced the novel concept of a Hybrid Patient Record (HyPR). Based on a field and design study, we presented the design and implementation of a HyPR device that supports (i) dual use, by allowing the pairing of the paper and digital information, (ii) recognizability, by allowing for dynamic color and sound coding of the record, and (iii) mobility, by using a portable form factor and location tracking. We presented initial feedback from a clinical simulation indicating that the HyPR device decreases configuration work, supports mobility in clinical work, and increases awareness on patient data.

ACKNOWLEDGMENTS

This work was supported by the EU Marie Curie Network iCareNet under grant number 264738. Thanks to Mathias Schmidt, Thomas Snaidero, and Vaidas Sirtautas for helping with the implementation of the HyPR device. Thanks to Shahram Jalalinia for help on the technology probe and Sanne Jensen for helping conducting the medical simulation.

REFERENCES

1. K. Ahmed, M. Jawad, M. Abboudi, A. Gavazzi, A. Darzi, T. Athanasiou, J. Vale, M. S. Khan, and P. Dasgupta. Effectiveness of procedural simulation in urology: a systematic review. *The Journal of urology*, 186(1):26–34, 2011.
2. T. Arai, D. Aust, and S. E. Hudson. Paperlink: a technique for hyperlinking from real paper to electronic content. In *Proc. of CHI '97*, pages 327–334. ACM.
3. J. E. Bardram. Activity-based computing for medical work in hospitals. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 16(2):10, 2009.
4. J. E. Bardram and C. Bossen. Mobility work: The spatial dimension of collaboration at a hospital. *Comput. Supported Coop. Work*, 14(2):131–160, Apr. 2005.
5. Y. Chen. Documenting transitional information in emr. In *Proc. of CHI '10*, pages 1787–1796. ACM.
6. F. Geyer, J. Budzinski, and H. Reiterer. Ideavis: a hybrid workspace and interactive visualization for paper-based collaborative sketching sessions. In *Proc. of NordiCHI '12*, pages 331–340. ACM.
7. F. Guimbretière. Paper augmented digital documents. In *Proc. of UIST '03*, pages 51–60. ACM.
8. R. H. Harper, K. P. O'Hara, A. J. Sellen, and D. J. Duthie. Toward the paperless hospital? *British Journal of Anaesthesia*, 78(6):762–767, 1997.
9. J. M. Heiner, S. E. Hudson, and K. Tanaka. Linking and messaging from real paper in the paper pda. In *Proc. of UIST '99*, pages 179–186. ACM.
10. S. Houben, S. Nielsen, M. Esbensen, and J. Bardram. Noosphere: An activity-centric infrastructure for distributed interaction. In *Proc. of MUM '13*, pages 100–110. ACM, 2013.
11. S. R. Klemmer, M. W. Newman, R. Farrell, M. Bilezikjian, and J. A. Landay. The designers' outpost: a tangible interface for collaborative web site. In *Proc. of UIST '01*, pages 1–10. ACM.
12. A. Kushniruk, C. Nohr, S. Jensen, and E. Borycki. From usability testing to clinical simulations: Bringing context into the design and evaluation of usable and safe health information technologies. *Yearb Med Inform*, 78:85, 2013.
13. C. Liao, F. Guimbretière, and K. Hinckley. Papiercraft: a command system for interactive paper. In *Proc. of UIST '05*, pages 241–244. ACM.
14. C. Liao, Q. Liu, B. Liew, and L. Wilcox. Pacer: fine-grained interactive paper via camera-touch hybrid gestures on a cell phone. In *Proc. of CHI '01*, pages 2441–2450. ACM.
15. P. Luff and C. Heath. Mobility in collaboration. In *Proc. of CSCW '98*, pages 305–314. ACM.
16. W. E. Mackay, G. Pothier, C. Letondal, K. Bøegh, and H. E. Sørensen. The missing link: augmenting biology laboratory notebooks. In *Proc. of UIST '02*, pages 41–50. ACM.
17. S. Y. Park and Y. Chen. Adaptation as design: Learning from an emr deployment study. In *Proc. of CHI '12*, pages 2097–2106. ACM.
18. T. Pederson. Magic touch: A simple object location tracking system enabling the development of physical-virtual artefacts in office environments. *Personal and Ubiquitous Computing*, 5(1):54–57, 2001.
19. T. Pietrzak, S. Malacria, and É. Lecolinet. S-notebook: augmenting mobile devices with interactive paper for data management. In *Proc. of AVI '12*, pages 733–736. ACM.
20. M. D. Rodriguez, J. Favela, E. A. Martínez, and M. A. Muñoz. Location-aware access to hospital information and services. *Information Technology in Biomedicine, IEEE Transactions on*, 8(4):448–455, 2004.
21. J. J. Saleem, A. L. Russ, C. F. Justice, H. Hagg, P. R. Ebright, P. A. Woodbridge, and B. N. Doebbeling. Exploring the persistence of paper with the electronic health record. *International journal of medical informatics*, 78(9):618–628, 2009.
22. A. J. Sellen and R. Harper. *The myth of the paperless office*. The MIT Press, 2003.
23. B. Skov and T. Høegh. Supporting information access in a hospital ward by a context-aware mobile electronic patient record. *Personal and Ubiquitous Computing*, 10(4):205–214, 2006.
24. A. Tabard, W. E. Mackay, and E. Eastmond. From individual to collaborative: the evolution of prism, a hybrid laboratory notebook. In *proc. of CSCW '08*, pages 569–578. ACM.
25. C. Tang and S. Carpendale. Evaluating the deployment of a mobile technology in a hospital ward. In *Proc. of CSCW '08*, pages 205–214. ACM.
26. P. C. Tang, J. S. Ash, D. W. Bates, J. M. Overhage, and D. Z. Sands. Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption. *Journal of the American Medical Informatics Association*, 13(2):121–126, 2006.
27. R. H. Trigg, J. Blomberg, and L. Suchman. Moving document collections online: The evolution of a shared repository. In *Proc. of ECSCW '99*, pages 331–350. Springer.
28. N. Weibel, A. Ispas, B. Signer, and M. C. Norrie. Paperproof: a paper-digital proof-editing system. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems*, pages 2349–2354. ACM, 2008.
29. P. Wellner. The digitaldesk calculator: tangible manipulation on a desk top display. In *Proc. of UIST '91*, pages 27–33. ACM.
30. C. Winkler, J. Seifert, C. Reinartz, P. Krahmer, and E. Rukzio. Penbook: Bringing pen+paper interaction to a tablet device to facilitate paper-based workflows in the hospital domain. In *Proc. of ITS '13*, pages 283–286. ACM, 2013.
31. R. Yeh, C. Liao, S. Klemmer, F. Guimbretière, B. Lee, B. Kakaradov, J. Stamberger, and A. Paepcke. Butterflynet: a mobile capture and access system for field biology research. In *Proc. of CHI '06*, pages 571–580. ACM.
32. M. S. Zamarripa, V. M. Gonzalez, and J. Favela. The augmented patient chart: seamless integration of physical and digital artifacts for hospital work. In *Universal Access in Human-Computer Interaction. Applications and Services*, pages 1006–1015. Springer, 2007.
33. X. Zhou, M. S. Ackerman, and K. Zheng. I just don't know why it's gone: maintaining informal information use in inpatient care. In *Proc. of CHI '09*, pages 2061–2070. ACM.

Paper 7: HyPR Study

Title of paper

Collaborative Affordances of Hybrid Patient Record Technologies in Medical Work

Authors:

Steven Houben, *Mads Frost and Jakob E. Bardram*

Published:

In Proceedings of the 2015 Conference on Computer-Supported Cooperative Work and Social Computing ***(CSCW'15)****. ACM, 2015.*

Abstract:

The medical record is a central artifact used to organize, communicate and coordinate information related to patient care. Despite recent deployments of electronic health records (EHR), paper medical records are still widely used because of the affordances of paper. Although a number of approaches explored the integration of paper and digital technology, there are still a wide range of open issues in the design of technologies that integrate digital and paper-based medical records. This paper studies the use of one such novel technology, called the Hybrid Patient Record (HyPR), that is designed to digitally augment a paper medical record. We report on two studies: a field study in which we describe the benefits and challenges of using a combination of electronic and paper-based medical records in a large university hospital and a deployment study in which we analyze how 8 clinicians used the HyPR in a medical simulation. Based on these empirical studies, this paper introduces and discusses the concept of collaborative affordances*, which describes a set of properties of the medical record that foster collaborative collocated work.*

Collaborative Affordances of Hybrid Patient Record Technologies in Medical Work

Steven Houben, Mads Frost, Jakob E. Bardram
The Pervasive Interaction Technology Laboratory
IT University of Copenhagen, Rued Langgaardsvej 7, DK-2300 Copenhagen, DK
{shou,madsf,bardram}@itu.dk

ABSTRACT
The medical record is a central artifact used to organize, communicate and coordinate information related to patient care. Despite recent deployments of electronic health records (EHR), paper medical records are still widely used because of the affordances of paper. Although a number of approaches explored the integration of paper and digital technology, there are still a wide range of open issues in the design of technologies that integrate digital and paper-based medical records. This paper studies the use of one such novel technology, called the Hybrid Patient Record (HyPR), that is designed to digitally augment a paper medical record. We report on two studies: a field study in which we describe the benefits and challenges of using a combination of electronic and paper-based medical records in a large university hospital and a deployment study in which we analyze how 8 clinicians used the HyPR in a medical simulation. Based on these empirical studies, this paper introduces and discusses the concept of *collaborative affordances*, which describes a set of properties of the medical record that foster collaborative collocated work.

Author Keywords
Hybrid Patient Record; Collaborative Affordance; Hospitals; Electronic Health Record; EHR; Mixed Reality Interaction

ACM Classification Keywords
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces; J.3 [**Computer Applications**]: Life and Medical Science.

INTRODUCTION
Information management in medical work in hospitals is centered around medical records: a legal document containing detailed information about the patient's treatment. Medical records are organizational and coordinative artifacts that are used to share, communicate and manage complex treatment procedures [7]. To improve efficiency and quality in care, the Western world is investing significant resources in digitizing healthcare with a special focus on creating an integrated Electronic Health Record (EHR) [16]. The EHR offers a number of fundamental advantages over the Paper Medical Record (PMR) related to quality of health care, efficiency in use and a higher level of patient safety [11].

Despite this ongoing trend, hospitals still use PMRs as part of the daily workflow [5, 12, 37]. The significance and affordances of paper have been drawn together in Sellen and Harper's book — *The Myth of the Paperless Office* [36] — and are also highlighted in reflections on PMRs in clinical work [16]. Studies show that paper makes clinicians more efficient at their work [34] and that the use of paper forms increases significantly after the introduction of an EHR [35]. In fact, there is little to no evidence about the actual effectiveness of some of the new digitized workflows [8, 43] or EHRs [19]. Moreover, the EHR often operates as a passive information repository and is therefore often supplemented with a PMR which holds more informal documentation, such as ad hoc notes, as part of a *working record* [15]. Furthermore, PMRs frequently function as *transitional artifacts* [12] that mediate the information flow between day to day work in the hospital and the EHR, while also providing redundancy of information [10, 14]. As a consequence, clinicians *"... continue to maintain a hybrid documentation environment"* [13][p. 160] and a typical setup in many hospitals is that the EHR system does not replace the PMR, but a *double record* consisting of both a paper and electronic part is maintained. This double medical record, however, introduces a number of configuration and coordination problems related to finding, using, updating, communicating and managing both records.

To address the ubiquity of paper in workplaces like hospitals, a number of technologies that integrate paper and digital technology have been proposed. Paperlink [2], or the commercial solution Anoto, provides a digital pen as a synchronization mechanism between written documents and digital storage of that data. Building on this idea, other examples such as the Paper Augmented Digital Documents (PADDs) [20], Paperproof [39] and PapierCraft [27] provide support for digital annotation of paper documents using pen input. Other approaches focus on the medical record. Penbook [40] supports capturing handwritten prescriptions by providing a hybrid setup using a touch screen and projector equipped with a digital pen. NOSTOS [3] supports data capture in emergency

CSCW 2015, March 14–18, 2015, Vancouver, BC, Canada.

ACM 978-1-4503-2922-4/15/03 ...$15.00.
http://dx.doi.org/10.1145/2675133.2675164

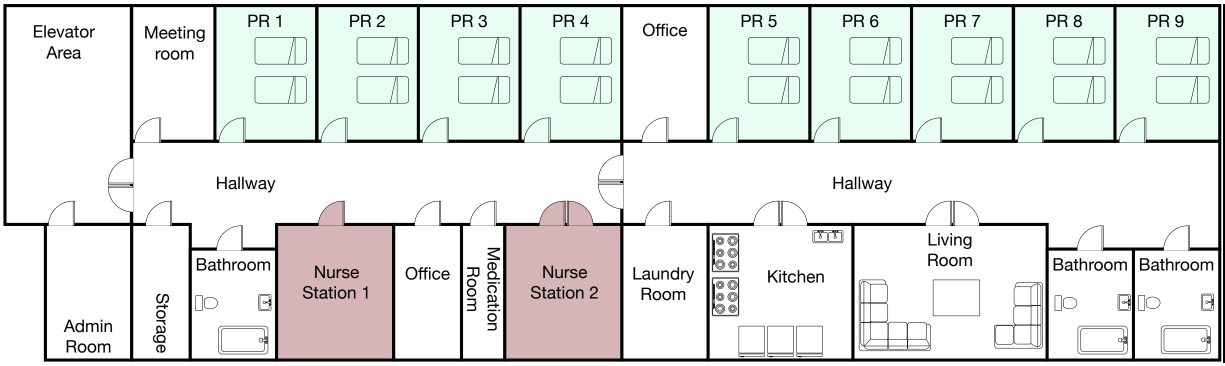


Figure 1. The physical layout of one of the wards from our field study. A typical ward consists of an administration desk, a number of patient rooms (PR), stations for the nurses, offices for the doctors, storage and medication rooms for medical equipment and finally bath- and living rooms for patients.

rooms and combines digital pens, wall displays and a digital desk to augment and enhance the PMR and form with digital patient information. Finally, the Augmented Paper Chart [41] provides seamless integration of paper notes with digital storage by using the Anoto pen.

This paper investigates one specific approach, called the Hybrid Patient Record (HyPR) (Figure 7) [23], for integrating the paper-based and electronic medical record. The HyPR device is attached to the PMR (like a notepad clip) and augments it with a notification system (color and sound), location tracking and ad hoc integration to a tablet that provides access to contextual relevant electronic patient data. By supporting various mechanisms to integrate the PMR and the EHR, the HyPR is designed to be a transitional artifact [12] that helps clinicians to gradually introduce digital tools in their use of medical records.

In this paper we make three contributions. First, we provide results from a detailed field study on the collaborative alignment and integration of PMR and EHR in different medical departments of a university hospital. This study verifies prior findings on collaboration in hospitals, but also contributes to the body of knowledge on the relationship between paper-based and electronic medical records. Second, we describe a study of the HyPR in a clinical simulation environment where 8 clinicians used the HyPR devices over two days performing scenarios originating from the field study. This study provides detailed insights into the use of hybrid or mixed reality technologies in collocated clinical work. Finally, we introduce and discuss the concept of '*collaborative affordances*', which is used to understand what features of a tool (be it paper-based or electronic) foster and support collocated collaboration. We analyze the collaborative affordances of the existing paper-based medical records and describe how they translate to the hybrid technology. Collaborative affordances thus extend the set of paper affordances identified by Sellen & Harper [36].

STUDY OF DOUBLE RECORD KEEPING

To thoroughly understand the nature of double record keeping, we conducted a field study on the use of the PMR and EHR in a large university hospital. The objective of the study was twofold; first, to obtain insights into the mechanisms clinicians adopt to collaboratively align and configure the two medical records in daily work and second, to prepare for the clinical simulation study of the HyPR technology.

Setting

The study took place in a university teaching hospital with about 3,000 employees providing care for a municipality of about 400,000 people in greater Copenhagen, Denmark. The study involved five connected medical departments, covering two patient bed wards, two surgical departments and the emergency department. All five departments are located in the same building and work in close collaboration with each other. Patients treated in the surgical or emergency departments are sent to the bed wards for recovery and post-op care. Each of the bed wards admit 20 to 30 patients and employ about 15 staff members including doctors, nurses and administrative personnel. The bed wards share the same architecture and consist of a set of patient-related rooms, including patient rooms, living area, bathrooms and a set of rooms used by doctors, nurses and secretaries including the meeting room, nurses stations, medication room, ward offices and the administrative room (Figure 1).

Method

The field study applied participant observations, contextual inquiries and interviews. Observations included task-centric, artifact-centric, place-centric and person-centric observations of work in all the wards and departments. Task-centric observations provided an understanding of the tasks and activities performed in the different wards and departments. Artifact-centric observations studied the use of paper-based artifacts including the PMRs; the different medical information systems used including the EHR; other computing devices, such as digital whiteboards, mobile PDA devices, traditional desktop computers; specialized medical equipment and monitors; and other physical artifacts like whiteboards, carts and medical equipment. Place-centric observations studied the flow of work in and between departments, wards, meeting rooms and patient rooms. Person-centric observation comprised of contextual inquiries of nurses and doctors for one day followed by a post-hoc interview to get a more detailed understanding of the work in each department. In total there were 7 shadowing sessions, 5 follow-up interviews and 10 days of observation material (images and notes). The data were collected and recorded using photographs, audio tapes and extensive note taking, and were analyzed into reports, diagrams and workflow charts. To conclude the study, we conducted a follow-up workshop after the observations in which our findings were presented and verified.

The Medical Record Workflow

At the hospital, the medical record consists of a unique PMR. It is a legal requirement that this record is at all times present at the ward that is currently treating the patient. Although the content of the PMR varies between different departments, the record itself is standardized within the entire hospital. The record consists of a plastic cover that is marked with color-coded sections for different types of documentation (Figure 2). Documentation includes basic patient data, the narrative treatment record (called the 'continuation'), nursing documentation, various schemes and forms, observations, test results (e.g., radiology examinations) and messages from other medical professionals. Each record carries a label that uniquely identifies the patient by stating name and social security number both in text and encoded in a barcode. This label is attached to the front of the record. Normally, the PMR is between 2 and 3 cm thick, but the size of a record can take extreme proportions. In one case, the record of a cancer patient had to be distributed over several physical folders because of the large amount of documents accumulated over a long treatment period. At a patient ward, there are typically up to 25 active PMRs in use.

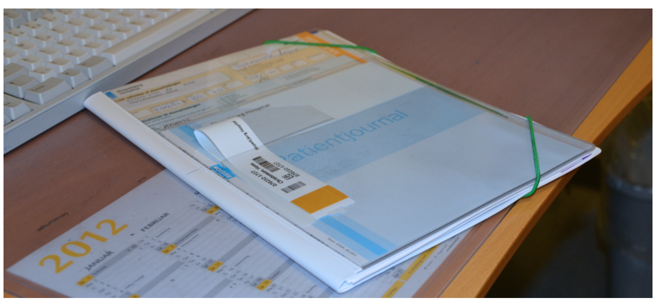


Figure 2. The standard PMR consists of a plastic folder that is labeled with the name and ID of the patient. The record holds all patient documentation and provides separate color-coded sections, e.g., for nurse notes, treatment history or other forms and observations.

Next to the PMR, the hospital provides a set of Health Information Systems (HISs), such as a Radiology Information System (RIS), an Electronic Medication System (EMS), a Patient Administration System (PAS), a Blood Bank System (BBS) and many others. Clinicians can access these applications through a system portal, which collates all applications into one interface, that is referred to as the EHR. Both the paper and digital information are often used simultaneously and are of equal importance. But in order to have electronically stored information 'ready-at-hand', information like lab results and radiology examinations are printed and added to the physical PMR.

Figure 3 shows the four main processes involving the medical record. The color code indicates whether only the PMR (blue), only the EHR (green) or both records (red) are used for that part of the process. The patient is referred to the hospital (e.g., for surgery) by the general practitioner (GP), who fills in an online form. This form is printed by the administration of the hospital when setting up an appointment for the patient. The form is stamped and approved by a doctor and sent

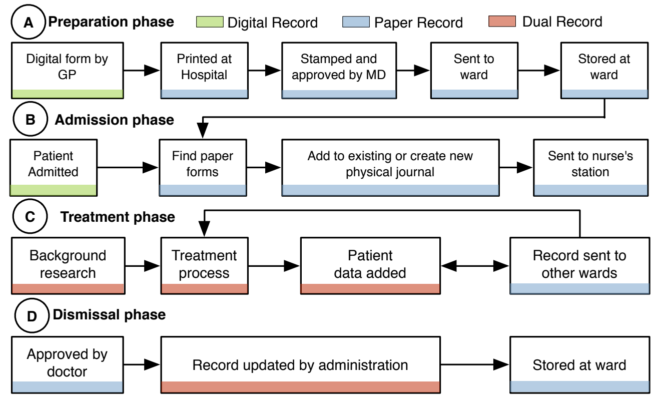


Figure 3. The life cycle of the PMR consists of a (A) preparation phase, (B) admission phase, (C) treatment phase and (D) dismissal phase (GP: general practitioner; MD: medical doctor).

to the ward responsible for the patient (Figure 3A – *preparation phase*). In the hospital, the general workflow surrounding the PMR is primarily managed by the ward secretaries and the nurses. When a patient is admitted to the hospital, the ward secretary locates the PMR. Most patients are readmitted to the same department and this 'home' department hence physically stores the PMR in the storage room. However, if a patient was previously treated at another department, locating the PMR can be a rather cumbersome process. Once located, the referral letter (e.g., from the GP) is added to the PMR. If a new patient with no PMR is admitted to the ward, one is created. The record is then sent to the nursing station the day before (or on the morning) the patient arrives. During the morning conference between the doctors and nurses, the record is used to prepare the arrival of the patient and to plan the treatment (Figure 3B – *admission phase*). Once the daily treatment and care of the patient has ended (Figure 3C – *treatment phase*), the medical continuation is updated by a ward secretary while nurses update the nursing record, the medicine scheme and add relevant examination results to the record (Figure 3D – *dismissal phase*). Once the patient is discharged from the department, the PMR is finalized and stored at the ward. This implies that hundreds of archived records are at the department.

FINDINGS

The field study on the use of (double) medical records provided three main findings related to: (i) establishing workflows, awareness and coordination; (ii) micro- and macro-mobility of medical records; and (iii) how medical record sub-artifacts are collated and aligned. These findings relate well to those from other studies of the use of (electronic) medical records in hospital environments, but also add a detailed insight into how medical professionals handle such double record keeping.

Workflow, Awareness and Coordination

Several studies (e.g., [12, 33, 37]) have shown that the PMR serves as a key coordination mechanism between clinicians, which is confirmed in our study. The PMR is used as a coordination mechanism between nurses and doctors inside the patient ward where the doctor can give orders concerning the patient via the PMR and in turn, the nurses note information in the record that helps the physician decide what to do next for the patient. In our study, we found that the PMR is actively used during coordination of patient treatment and care during the morning conference. The content of the record including the latest examination results is key in deciding on treatment and care as well as the allocation of doctors and nurses to the patient. As such, the record is essential in coordinating treatment within and between departments, which is reflected in the fact that the record is always required by law to follow the patient to other departments.

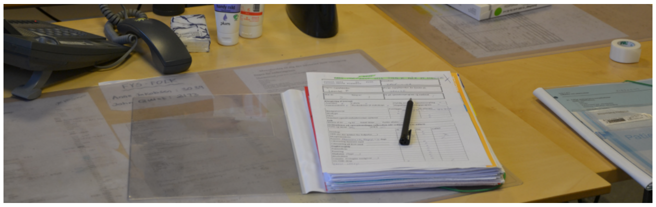


Figure 4. The placement of a PMR (e.g., spatial orientation or opening the record) is used to signal information to other clinicians.

The PMR is also key in a more subtle coordination inside the ward as its physical form helps to achieve local coordination and awareness. The physical placement of records often reveals status information and can be used for signaling and for drawing attention to important matters. For example, the PMRs shown in figure 4 are deliberately placed on the desk by a nurse to signal to the doctor that the paper forms in the record should be inspected and validated. Similarly, the PMR would often be placed in the bed of the patient inside a patient room, visible from the hallway. This is a signal to the porter that this patient is ready for pick-up and can be moved, e.g., to surgery. This phenomenon of signaling through document placement was also observed by Bång et al. [3] who noted that if, e.g., a physician wanted a laboratory test, the request was attached to the folder that was then placed at a designated spot on the desk. Hence, our study confirms that the processes of signaling and monitoring are used in medical work to constitute mutual workplace awareness [6]. However, an interesting addition to prior research is that in this case, not only humans but also computational devices work as active signaling actors. For example, printing lab results directly on the ward's printer is a signal from the lab to the ward. In general, the spatial arrangement of documents bestows local importance on the workplace and provides an overview and deposition of information that facilitates memory recall and the tracking of work processes. The placement of a record inside the ward, the nurses station, the filing cabinet or storage room convey information about their role in the overall workflow.

Micro- and Macro-mobility

Medical work in hospitals is inherently nomadic [4], which implies that clinicians and the tools they use move around inside wards, departments and the entire hospital. This mobility includes the PMR, which moves between different departments, wards and locations within the ward [32]. As such, the mobility affordance of PMRs fits medical work well, as they are easy to move around inside a hospital. Our study showed that the PMR is primarily moved around inside the patient ward and is mostly used in the ward offices, nursing stations, doctor's offices and at the bedside of the patient. However, since it is a legal requirement that the record is present during medical treatment, the record always 'travels with the patient'. For example, when patients are sent to other departments (e.g., for x-ray or surgery), the record is mounted in a special container on the side of the patient's bed and travels with the patient to the receiving department. This flexible mobility affordance of the PMR, however, also introduced significant challenges since it often was lost or misplaced both inside the ward and in other departments. Significant time was spent looking for records, especially by the ward secretaries. Hence, a core challenge to a paper-based workflow is that sometimes the physical record is misplaced or not available. As one of the nurses explained:

> *"Doing background research on a patient can be difficult when the patient does not belong to our ward. Then the PMR is elsewhere and the digital data might not be up to date. All we can do is wait for the patient and improvise."*

But also finding records inside the ward offers its own set of challenges. Finding the right PMR is difficult because there is no visual differentiation between different records; they are all stacked upon each other and scattered all over the ward in the nurses station, the secretary offices and in the storage room.

In addition to mobility on a macro level, micro-mobility of a PMR plays a central role in medical work [28]. Micro-mobility is *the way in which an artifact can by mobilized and manipulated for various purposes around a relatively circumscribed, or 'at hand', domain*. For example, during the ward round a physician and nurse jointly worked on a PMR by standing next to each other reading the record. They handed over parts of the record to each other, pointing out specific results. Furthermore, the record was often broken open and the individual records, results, forms and graphs were spread out on, e.g., a desk for better overview. This micro-mobility affordance of a PMR hence supports clinicians in achieving an overview of the medical situation at hand [9].

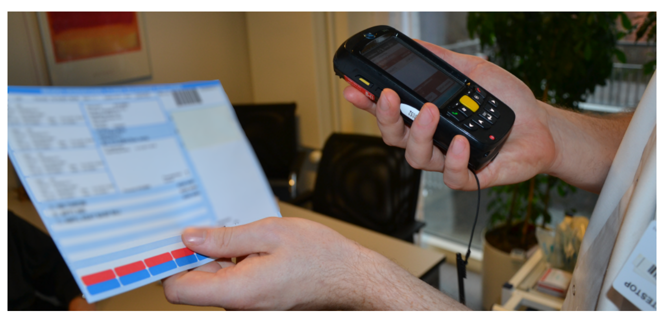


Figure 5. A nurse completing a form while using a PDA to scan the vacutainer after taking a blood sample.

Moreover, as seen in figure 5, nurses often 'break open' the record by only taking parts of the record with them, while interacting with patients. In this specific case, a nurse has brought the record and placed it on the table before taking a blood test from the patient, scanning the tube with the PDA and adding the form to the PMR. In current work practices, this kind of micro-mobility alignment between paper and digital information is often cumbersome. Additionally, since the

PDA only provides limited access to the patient data (in this case only the blood work), the clinician will have to return to the nurses station to manually add any data to the electronic record. The mobile device as such is only used to ensure the blood test was taken from the correct patient.

Artifact Collation and Alignment

Core to medical overview and decision making is the collation and alignment of information from many sources. This includes both the many different paper forms and records in the PMR, as well as the information located in the EHR. Hence, significant effort was put into collation and alignment of medical information for several sources to get a comprehensive overview of a patient's medical state. As seen in figure 6, both the paper and digital information are of equal importance and are thus often used simultaneous. This implies that while some information is duplicated in both records, other information only exists in one of both.

Figure 6. Clinicians are using both the PMR and EHR to coordinate information.

Managing the dual record introduces a number of configuration challenges related to managing, synchronizing, communicating and cross-referencing both versions of the record. Current work practices still include printing a significant amount of information, which is then stored in the PMR. Furthermore, most of the digital applications can only be used to request or add new medical data (such as blood test or MRI). The results of these requested tests are often still sent by paper trough internal mail or by sending the results to the printer located at the requesting ward. This places a large and important coordinative role on the printer, which essentially operates as a communication and awareness mechanism. It also implies that although a lot of time and effort is invested in printing, often these printouts are quickly outdated compared to the digital record, or even get lost throughout the printing process. There was a general recognition that aiming for a completely 'paperless hospital' would be naive. Moreover, information on large whiteboards were also constantly updated to align information across several records and other coordinative artifacts. The need for collation and alignment of information across such a 'web of coordinative artifacts' is known to be essential for the general flow and coordination of medical work [5].

HYBRID MEDICAL RECORD TECHNOLOGY

As argued above, there is a significant set of configuration challenges related to managing, synchronizing, communicating and cross-referencing both versions of the medical record.

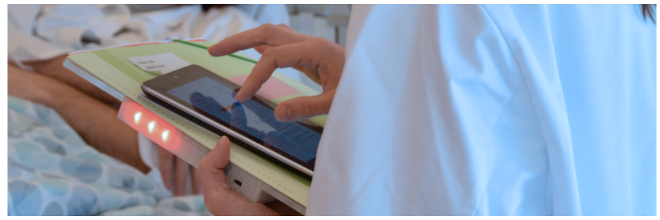


Figure 7. The HyPR device augments the PMR with a notification system (color and sound), location tracking and near-field communication that uniquely identify the medical record, which allows clinicians to pair a tablet with the HyPR to display the digital information associated with the present PMR.

In order to provide a technical design to mitigate these problems, we have previously proposed an augmented hybrid patient record or HyPR device (Figure 7) [23]. The HyPR device is a small sensing and computing platform that supports notification through colored light and sound, location tracking and unique identification using near-field communication (NFC). The HyPR device is designed to be 'clipped on' to the paper-based record as an augmentation of the plastic folder used now. Hence, one HyPR device should be deployed for each PMR in use in a hospital department.

As shown in Figure 7, the HyPR device is attached to the PMR and, by placing a mobile device like a tablet computer or a smart phone on top of the record, relevant electronic information from the EHR is loaded and displayed. The HyPR embeds a unique id, which is communicated to the mobile device using NFC and Wifi and is used to retrieve relevant patient information in the EHR. On the side of the HyPR device, an array of LED lights can be used for signaling purposes and the device also has a small buzzer for sound notifications. These notification features can be used to convey status information about a patient, such as when the patient is ready for the ward round or should be prepared for surgery. The HyPR also embeds a location tracking tag that allows for tracking the location of the physical paper record. This allows clinicians to retrieve the location of a record when missing. Additionally, they can turn on the buzzer or set the colored lights to blink, to visually or auditory locate the HyPR.

The overall design goal of the HyPR approach is to create a transitional artifact allowing clinicians to easily move between paper-based and digital records. Clinicians thus benefit from both the portability and flexibility of paper-based records as well as the easy access and information processing capabilities of electronic medical records. As such, the goal is to reduce the amount of configuration work required to use and setup this dual record. Specifically, the HyPR approach supports: easy and fast *configuration* of the dual record by allowing clinicians to connect the paper and digital information, *coordination* by providing a notification system equipped with dynamic colored lights and sound and *mobility* by including location tracking capabilities to the portable device. The HyPR approach supports flexible and dynamic configuration of paper and digital information, which allow for a gradual transition between paper and digital. For example, paper-based forms can be digitized and stored in the EHR or digital material can be printed and stored in the PMR, all of which can be handled by the HyPR approach.

HYBRID MEDICAL RECORD STUDY

Previous studies of the HyPR device have only provided preliminary insights into its overall usability and usefulness [23]. Therefore, there is a need to understand the details of how this technology supports collaborative work in hospital departments, in particular, how it helps alleviating configuration challenges related to managing, synchronizing, communicating and cross-referencing both versions of the medical record. The study investigated to what degree the HyPR supports and potentially enhances existing clinical practices.

Since doing a field deployment of this type of technology is technically, legally and organizationally unfeasible, we conducted a clinical simulation in a separate 1:1 simulation environment. In the medical domain, a clinical simulation is a methodology frequently applied to train and educate clinicians in critical clinical scenarios, such as surgery, medicine prescription and administration and emergency cases. It has proved very efficient and reliable for the initial phase of training and assessment of clinical staff [1]. However, since the clinical simulation approach attempts to bring the dimension of clinical context into stronger focus, the method has lately been used also as a method for testing clinical systems with representative users doing representative tasks, in an ecologically valid setting [24]. The main source for this medical simulation was the data and insights obtained from the original hospital field study, which provided input for the physical setup, the scenarios and the configuration of the technology.

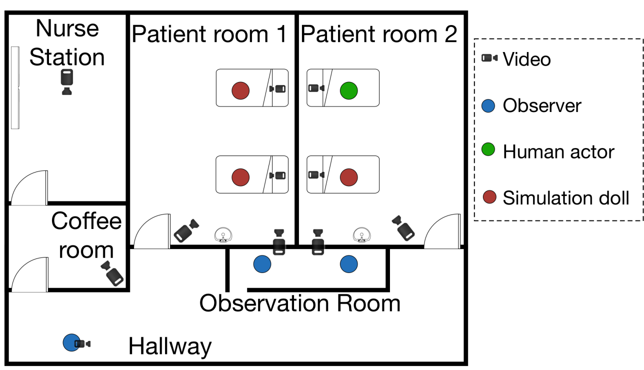


Figure 8. The simulation setup was comprised of a medical ward with five zones including two patient rooms, a nurse station, a coffee room and a hallway. The simulation facility is equipped with hidden cameras and an observation room behind a one-way mirror. The simulation included three simulation dolls (patients) and one human acting as a real patient.

Setting and Participants

The study was conducted at a 1:1 clinical simulation and training facility in a large hospital. This simulation facility supports the simulation of different hospital departments ranging from patient wards to surgical department and emergency ones. For our study, we set up the facility to be identical to a fully equipped patient ward with two patient bed rooms. Figure 8 shows the layout of the setup consisting of five zones: two patient rooms, a nurse station, a coffee room and the hallway. One human actor performed as a patient in a bed in room 2 (Figure 7, green dot). The other patient beds were equipped with simulation dolls, each connected to a monitor displaying the vital signs of the 'patient' (such as heart rate, saturation, blood pressure, temperature, etc). The setup included artifacts such as a traditional whiteboard with patients' data, desks in the nursing station with a stationary computer and nursing carts with medical equipment.

During the two-day simulation, 8 senior clinicians with different specialties (such as surgery, psychiatry and intensive care) participated in the experiment. Participants included 5 doctors, 2 nurses and a psychologist. The entire simulation was recorded using video and audio as well as extensive note taking and observations from inside the observation room through a one-way mirror.

Method

The study applied a scenario-based evaluation of the HyPR approach. The scenarios were drawn directly from the initial field study and revolved around interacting with the patients (both the actor and simulation dolls) to assess the patient case, update the status in the EHR and add or remove all necessary documents to the PMR. Scenarios included:

S1 *Ward Round* – Clinicians were asked to perform a ward round to assess the situation of four patients. By examining the patients and monitoring vitals signs on the monitor, they had to calculate an Early Warning Score (EWS) to describe their current status.

S2 *Blood Result* – Clinicians were asked to order a blood test result while working on the case of patient P1. After receiving the results they had to visit the patient, re-calculate the EWS and discuss the situation with the patient.

S3 *Lost Record* – Clinicians were asked to find a number of PMR, which, after a shift change, were not at their usual place. For this scenario, they could employ information on the patient's location, the last treating doctor, the current treatment procedure and the location of the HyPR.

After welcoming the participants, a brief introduction was delivered on the concept, the system and the physical layout of the ward. Participants were then asked to perform the three scenarios above. We did not provide any detailed instructions on how to perform the scenarios, which patients to look at first or how to use the system. Because we were interested in how clinicians would leverage their existing practices while using the HyPR setup, the scenarios were deliberately open ended: no explicit instructions or training on the system was given to them and the facilitator only intervened to solve technical issues. Because the initial field study showed that most medical work is highly collaborative, involving both doctors and nurses, the scenarios were conducted in pairs of two clinicians from the same department.

Artifact and Technology Setup

A list of patient cases with realistic names, backgrounds, social security number and medical background was compiled for the study. The PMRs used in the simulation contained real blood tests, EWS forms, admission forms, doctor and nurse notes and other medical information. The whiteboard placed in the nurse station listed all the patients with room number, treatment plan, responsible doctor and nurse and admission date. Four HyPR devices and three Nexus 7 tablet computers were used. We equiped the simulation facility with the Sonitor[1] ultrasound location tracking system in all rooms, excluding the hallway.

Since there was no open access to the medical information system in the hospital, we implemented a simple Activity-Centric EHR application to be used in the simulation. This application contained all patient cases with a set of medical entries such as blood test results, continuation records and nursing notes. The adaptable web application runs on phones, tablets and desktop computer and supports two views: (i) an overview screen and (ii) a patient details screen. The overview screen lists all patients currently active on the ward and provides basic information, such as name, assigned color, room number and ongoing medical procedures on each patient (Figure 9A). Using the 'Blink' button causes the colored lights of the HyPR to start blinking (Figure 7), asking for attention until a tablet is paired. The 'Buzz' button can be used to turn on a buzzing sound to quickly find a record. The buzzing sound is automatically stopped after 15 seconds.

Figure 9. The details of the patient record.

By placing the tablet on the HyPR device, the patient information (Figure 9B) is displayed. This screen allows clinicians to inspect prior observations and update medical information (e.g., EWS), add a message, or change the status color of the patient. Changing this color in the EHR will cause the color of the HyPR device to change accordingly (see Figure 7). The EHR on the tablet computer can also be used while not paired with an HyPR device. In this case, if new data (medical observations or a message) for a patient is entered into the EHR application, the patient's HyPR device (attached to the patient's PMR) will start to blink. In this way, the physical record signals changes in the digital record. Once a tablet is paired with the HyPR record, the patient information is displayed and the device stops blinking. All patient information is thus organized in 'patient activities' which are visualized in the overview screen and can be quickly and easily accessed through physical or remote interaction.

FINDINGS

The objective of the simulation was to study how work practices observed in the field study would translate to the HyPR. As such, we were interested in relating the observations of the HyPR approach to current work practices at a medical ward. The discussion of the simulation findings are therefore framed in the three main findings from the field study on the use of medical records in the hospital; (i) establishing workflows, awareness and coordination; (ii) micro- and macro-mobility of medical records; and, (iii) how medical record sub-artifacts are collated and aligned.

Workflow, Awareness and Coordination

Since the HyPR encloses the PMR, it was used as a coordination mechanism in exactly the same way as seen during the field studies. Clinicians would use the record for information sharing, during clinical conferences and as the key coordination mechanisms between doctors and nurses during, e.g., the ward round. It was, however, interesting to observe how the features and functionality of the HyPR sparked new workflows, awareness and coordination practices.

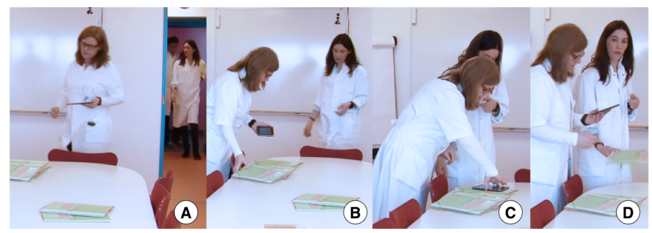


Figure 10. Clinicians glance at the PMRs (A) and notice that a colored light is blinking, indicating a new message (B). They pair a tablet computer with the record to read the message on the tablet (C) and finally consult the paper forms for more details (D).

First, the colored lights of the HyPR triggered a number of novel workflow coordination mechanisms. The color was quickly appropriated as part of the externalization of the EWS. This meant that clinicians would use the color to organize and structure their ward round. One doctor, for example, mentioned that the colors helped him prioritize his patients during the ward round by using blue, green and red colors to indicate low, normal and acute patient cases. Colors were also used to reveal workflow status information. For example, during one of the ward round scenarios, a nurse had already appropriated the use of colored light as a method to keep track of the workflow:

> *"This [patient record] is green and it is not blinking, so he is fine." – P7*

Colors were also used for revealing if new content was added to the PMR. Figure 10 shows a video fragment of two nurses picking up a new message via the HyPR. First, the two nurses pass by the nurse station and glance at a number of PMRs in active use (Figure 10A). One nurse notices a blinking colored light indicating that new content has been added to the record (such as a new observation, message, or lab test result) (Figure 10B). The clinicians approach the record and, by placing the tablet on top of the record, they are able to read the updates (Figure 10C). They realize that they need more detailed information and align it with the paper documentation (Figure 10D) to construct a shared overview of all patient data.

Second, placement of the HyPR combined with the color light was also used for deliberately signaling status information.

[1] http://www.sonitor.com

Placing PMRs in the patient's bed was a signaling mechanism often observed during the field study. The simulation study showed that this work practice was continued and enhanced using the HyPR. We observed that clinicians carefully considered location and orientation when placing the HyPR. In the patient rooms, for example, clinicians would often position the records in such a way that the lights were visible from the hallway. This mechanism was adopted so that clinicians could easily glance inside the room and check if the colored light was changed or if a new message was received. They considered the colored lights to be an important collaborative affordance that helped them share and externalize the status of the patient in a fast and efficient way.

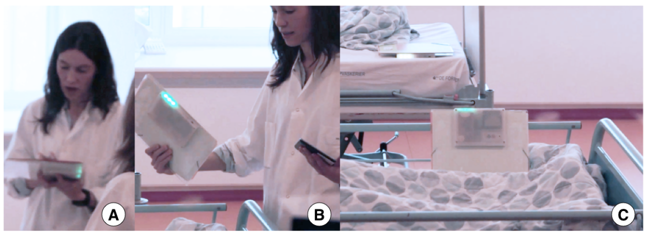


Figure 11. The clinician uses the record (A), checks if the colored light matches her assessment (B) and places both records in such a way that they are visible from the hallway (C).

Figure 11 shows a video fragment of a nurse using the HyPR for signaling. After finishing assessing the patient case (Figure 11A), the clinician double checks the color to see if it matches her assessment color (Figure 11B) and then positions the record to allow for visibility from the hallway (Figure 11C). Interestingly, this positioning was done differently depending on the location of the bed in the patient room:

> *"Then I should place it so one can see the light.. now it lies across his legs, then I should probably move it.. there!" – P8*

As seen in Figure 11C, the record on the second bed in the background is positioned differently than the one on the bed in the foreground. This was done because the bed in the foreground was close to the wall, implying it would not be visible from the hallway if it were flat on the bed. With the current positioning both records where visible from the hallway.

Micro- and Macro-mobility

The HyPR device is designed to maintain the macro-mobility affordance of regular PMRs. The medical simulation study clearly showed that a HyPR was carried around in the simulation facility just like regular PMRs are carried around. This can be seen in the video fragments in Figures 11, 14 and 16. However, as observed during the field studies, there are a number of problems related to mobility in clinical work, including the problem of lost PMRs and the lack of support for accessing the EHR while roaming the ward or hospital. The HyPR was designed to mitigate these problems by supporting location tracking of the HyPR and by allowing access to the EHR via a tablet that is paired with the PMR.

During the simulation, clinicians very quickly appropriated these features and we observed a number of recurring patterns in the mobile use of the HyPR. Figure 12 shows a video fragment of two clinicians searching for a lost PMR. The search strategy was very similar for most clinicians. First, they use the tablet computer to look up the location of the HyPR PMR. Since location tracking is room-based, they enter the room and look for the record. If it is not immediately visible, they use the tablet to turn on a blinking pattern on the record (Figure 12A). This approach only works if the PMR is in plain sight, e.g., on a desk or on a stack of other records, but not if the record is in a cupboard or drawer. Clinicians therefore turn on the buzzing sound and divide the mobility work between both of them. One clinician holds the tablet and repeatedly presses the buzz button while the other clinician tries to identify the location of the record with the instructions of the clinician that is holding the tablet and who has a better idea on where the sound is coming from (Figure 12B). After finding the tablet, it is still blinking from the initial search attempt, thus helping them verify that they have found the correct PMR (Figure 12C).

Figure 12. Clinicians use the color blinking (A) and buzzing feature (B) to identity the location of the record (C).

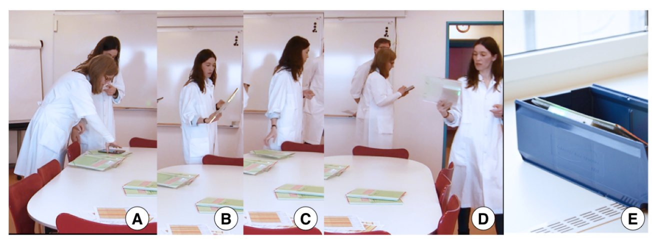


Figure 13. Clinicians inspect a pile of PMRs (A), select the one that they need (B) and position it strategically on the table while discussing it (C). Afterwards the doctor takes it (D) and places it in the out-tray with the colored lights clearly visible (E)

The field study also revealed extensive micro-mobility of PMRs; the spatial orientation and positioning of the record often carry a meaning as it symbolizes and reflects work processes. During the simulation, we observed similar micro-mobility use patterns with the HyPR as depicted in the video fragment in Figure 13. Clinicians inspect a pile of records and spatially categorize them based on their importance (Figure 13A). After selecting and inspecting a particular PMR (Figure 13B), the doctor places the record on the corner of the table (Figure 13C) to indicate that this is the record they are currently using. After discussing the patient case, they realize that this patient is no longer at the ward and the record is therefore no longer needed at the ward, but has to be sent to another ward (Figure 13D). The doctor then places the record in the out-tray with the colored lights facing upwards and clearly visible (Figure 13E).

Artifact Collation and Alignment

The original field study revealed significant work associated with aligning and collating PMR with EHR. A first observation from the simulation study reveals that record alignment and collation no longer takes place at the desk in the nurses station (as seen in Figure 6), but can be done by clinicians while roaming around inside the entire ward. All clinicians followed a very similar strategy in aligning the PMR and EHR. Figure 14 shows a video fragment showcasing the new approach available through the HyPR, performed by the patient's bedside. The clinicians first pair the paper and digital record to get an overview on the patient case (Figure 14A). They then jointly inspect both the paper and digital information and explicitly check if any new observations were added to the digital or PMR (Figure 14B). After discussing the case with the patient, they add a new observation to the digital record and place the paper forms back in the record (Figure 14C).

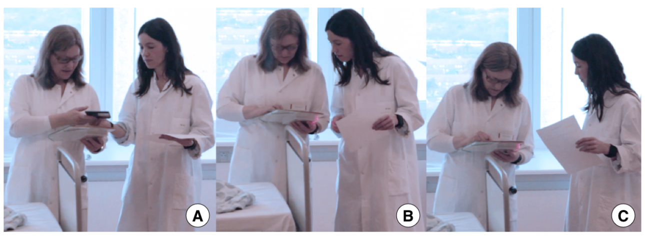


Figure 14. Clinicians first align the paper and digital record (A), then inspect both types of documentation (B) and finally update the documentation in both records (C).

While interacting with a patient, clinicians would divide the PMR between both of them. One clinicians would thus hold the PMR and inspect all the printed blood tests and paper forms, while the other clinicians would consult the paired tablet for the EHR. However, while moving to another patient or room, one clinician would carry both the HyPR and tablet, while the other clinician would focus on the patient. One clinician is thus often "left behind". Busy adding information to the electronic record using the tablet, the other clinician moves to the next patient. Interestingly, we also observed that although clinicians had easier access to both records, they sometimes were reluctant to use all its features. At one time a nurse, for example, whispered to the doctor:

> *"I am very much against that we should be writing it in here. We should do this at the nurses station." – P4*

She essentially disliked using the system in front of patients, as she argued that it detached the contact between the clinician and the patient.

During the EWS assessments of patients, we often observed that clinicians would spatially organize information that was needed to better understand the case. The video fragment in Figure 15 shows two clinicians doing a ward round while using the patient bed to organize and collate all the paper forms stored in the PMR and in the tablet computer. They first collate the prior blood results paper, EWS forms and digital messages (Figure 15A). Based on this information, they discuss

Figure 15. Two clinicians are using the patient bed to spatially organize both the paper and digital record of the patient (A-B), before adding content to the digital record (C).

the patient case and compare the previous data to the live information on the monitor (Figure 15B). Finally, they complete a new EWS form and add the form to the PMR while at the same time recording the EWS score to the digital record (Figure 15C).

When a physical surface was not available for spatial organization, clinicians would perform the collation in a much more mobile setting. The video fragment in Figure 16 depicts two clinicians that are collating medical forms, blood tests and digital record entries. While talking to the patient, the doctor (Figure 16A — on the right) is studying previous data. Based on the vital signs on the monitor, he requests the other clinician (Figure 16B — on the left) to update the electronic record with his assessment. He also asks the other clinician to update the color of the record to match his assessment (Figure 16C). Before leaving the patient room, the doctor checks the HyPR to see if the color has been updated based on his assessment of the patient (Figure 16D).

Figure 16. During the ward round, two clinicians are collating the paper and digital record (A-B) of the patient while standing at the bed assessing the patient case (C-D).

DISCUSSION

Prior studies [5, 12, 37] as well as this paper show that paper documentation remains to play an important central role as it is persistently and pervasively used during medical work in hospitals. This intensive use of paper documentation seems to be independent from the degree to which EHRs are integrated as paper simply makes clinicians more efficient in parts of their work [34]. Based on Gibson's theory of affordances [18], Sellen and Harper have argued that paper in general possesses a set of affordances, that makes it especially efficient in use [36]. These affordances include the ability to quickly navigate through documents, read across multiple documents at once, mark up a document while reading and interweave reading and writing. Looking more specifically to the medical domain, Harper et al. [21] point to the affordances of *flexibility*, *markability*, *portability* and *accessibility* of the anesthesia record that makes it easy to fill out, share and use during surgery.

Collaborative Affordance - *Affords the ability to*
Mobility and Portability
Physically carry, share and use the record in different places
Collocated Access
Simultaneous and collocated read and update the record
Shared Overview
Collectively create an overview of the content on the record
Mutual Awareness
Signal and monitor information between users

Table 1. The Collaborative Affordances of PMR and HyPR.

Based on the two studies reported in this paper, we argue that the core benefit of a paper-based medical record does not solely lie in its core basic affordances, but that a set of *collaborative affordances* exist that support clinicians in coordinating work. Rather than applying the original definition of affordance provided by Gibson [18], we rely on Norman's interpretation for human-computer interaction[2], in which affordance refers to those action possibilities that are *readily perceivable* by an actor [29]. In this definition, affordances depend not only on the physical capabilities of an actor, but also on the actor's goals, plans, values, beliefs and past experiences. Gaver describes that the perception of affordances is "*embedded in the observer's culture, social setting, experience and intentions*" [17]. Although affordances exist as a configuration of physical properties, its perceptible meaning is often dependent on the social strata and can thus change or differ between environments or social setting. Within the framework of distributed cognition, Zhang and Patel [42] refer to this dependency to culture as *cognitive affordance*. Using Activity Theory, Kaptelinin and Nardi reconceptualize affordances in a social-culture background describing them as *mediating actions* [25]. Similarely, Vyas et al. [38] describe that *affordances in interaction* exist between the user and environment, emerges from activities and practices and are therefore socially and culturally constructed. Finally, Kreijns and Kirschner [26] introduced *social affordances* as properties of collaborative environments "*which act as social-contextual facilitators relevant for the learners social interactions.*"

Based on these prior interpretations of the social role of affordances, we describe *collaborative affordances* as 'physical' properties that afford — make possible different actions for a person perceiving the object — collaborative perceptions and actions within a specific social context. Collaborative affordances do not replace the normal affordances of paper, but rather contextualize them in a social structure. Table 1 lists four basic collaborative affordances of the medical record which were observed in both studies. As technology comes closer to paper (as in the HyPR approach), questions arise on how these collaborative affordances translate to technology. Using the field study and the clinical simulation as case studies, we describe below how the four collaborative affordances described in Table 1 were observed in the initial study and how they translate into the HyPR approach.

[2]Although Norman nowadays prefers the terms perceived affordance [30] or signifier [31] over affordance.

Mobility & Portability

Portability, the ability to carry, maneuver and navigate, is an important affordance of paper [36], hence, also of the PMR [21]. From a collaborative stance, the portability and mobility of PMRs is a central reason for its success in mediating cooperative medical work. As argued by Østerlund [32], a PMR serves as a *portable place* in the sense that it can move across space and time but retain the indexical structure which points out relevant participants, places and times. This collaborative affordance allows several clinicians to use the record on the move as they continuously perform care activities for many patients across multiple locations. Such macro-mobility inside and across patient wards was also found in both of the presented studies: PMR as well as HyPR were carried around and used during care activities (e.g., ward rounds) and this portability of the record helped clinicians to jointly accomplish their work. Although the HyPR in its current state is relatively heavy and bulky due to the sensor platform, the technology essentially incorporates support for mobility as there was less mobility work [4] required to configure the work setup. The mobility and portability affordance of the HyPR is much closer to that of a PMR, specially if compared to other approaches that attempt to include mobility use for the EHR (e.g., Computers on Wheels (COW) [37]). By using the PMR as a contextual surface that auto-configures the paired tablet, HyPR mitigates the high cost of information access [37] and physical [21] challenges of handling digitized medical information at the bedside.

Collocated Access

Paper has a high degree of configurability as it affords simultaneous access through reading and writing [36]. These affordances are key to PMRs, since *working records* [15] and *transitional artifact*s [12] are used by clinicians as a coordinative reflective tool to bridge the gap between day to day work in the hospital and managing the EHR. In the course of the working shift, clinicians keep these documents to continuously gather information on the move and gradually transfer them back to the official record [32]. Our studies of the use of PMRs and the HyPR approach emphasize the collaborative nature of such simultaneous access to medical records. Records were often used in a collocated setting in which typically a pair of a nurse and a doctor would break open the record and simultaneously inspect and access the documents and forms. Examples of situations in which collocated access of the medical record is evident include the ward conference situation and the use of the record at the patient's bedside during a ward round (Figure 14). Such situations are examples of *standard operation configurations* [4] in a hospital, which are spatial setups fostering easy cooperation because of a common knowledge and agreement as to how to use and navigate the artifacts involved. A core requirement for medical records is that they embed this collaborative affordance of collocated access, which enables them to be part of such standard operation configurations. Using HyPR provides clinicians with a high degree of plasticity to import a digital device into standard operation configurations. Furthermore, by using a tablet computer with a size that fits into the clinicians' white coat pockets, the tablet can easily be brought in and out of the configuration. Although the sensors of the HyPR device are clearly visible to the clinicians, the technology was considered to be transparent as clinicians did not mention the mediating hardware at all, but rather talked about the HyPR as a 'smart paper-based medical record'.

Shared Overview

One of the most prominent affordance of paper is that it supports quick and flexible navigation and simultaneous access of multiple documents [36]. This is extensively used in a clinical setting, in which the PMR is key in obtaining an overview of the treatment and care of a patient [9]. Our studies of both the PMR and HyPR showed that this creation of an overview is primarily a collaborative effort, as the records afforded the creation of a shared overview by aligning documents. In the case of the PMR, we observed that clinicians would collate and align medical information from several sources (both PMR and digitally, as shown in Figure 6) to get a comprehensive overview of a patient's medical state. Moreover, during the study of the HyPR record, we often observed that clinicians would spatially organize information that was needed to better understand the case. For example, the video fragment in Figure 15 shows two clinicians doing a ward round while using the patient bed to organize and collate all the paper forms stored in the PMR and in the tablet computer. The micro-mobility associated with this specific operation configuration, thus includes digital devices that can essentially be handled similar to another paper artifact while providing a portal into the EHR. As concluded by Bossen and Jensen:

> *"Collaborative overview requires a display that can be shared by two or more people in the many ad hoc conversations that take place, for example a display that can be carried and handed around like paper (like the printed lists), providing essential overview, but also making it possible to go into specifics"* [9][p. 11].

How the device contextualizes, integrates and visualizes large, complex and specialized EHRs to achieve *seamless integration* remains an important open question [14]. Nevertheless, HyPR can be used as a long term *stable concept* for a gradual movement towards a higher degree of integration and a less paper-centric hospital.

Mutual Awareness

Paper is not only easy to annotate and manipulate, but also provides an intrinsic historical account on these actions or changes [36]. Physical records are extensively used in achieving workplace awareness [6] in a hospital setting, as also evident in our studies of the PMR and HyPR. For example, the PMRs shown in figure 4 are deliberately placed on this desk to signal the hand-over from the nurse to the doctor. And, while using the HyPR record, clinicians positioned the HyPR in various ways (e.g., in the patient bed) to signal a status change, as seen in the video fragment in Figure 11. Similarly, clinicians were able to monitor places to pick up awareness information on status changes. For example, monitoring the printer in the ward office for lab results, as well as the HyPR records in the out-tray for status changes (Figure 13). Mutual awareness is a collaborative affordance as the medical record should allow the creation and perception of awareness information through the physical artifact. The physical properties of a PMR allow for placement in different places and positions — something that the EHR does not. HyPR did not remove any of the original collaborative affordances of the PMR, but rather supports and amplifies existing ones. The colored lights, for example, were used as *signifiers* [31] that allowed clinicians to externalize work practices into signals that helped them to optimize and prioritize interaction with patients. As such, medical records — PMR, EHR or HyPR — should be designed with the affordance of mutual awareness in mind, thus, providing clinicians with tools to *configure awareness* [22].

CONCLUSION

Medical records are key in coordinating treatment and care of patients in modern hospitals. Historically, they were paper-based, but due to the increased digitizing of medical information, more and more patient information is stored in different medical information systems. This creates a situation in which clinicians need to maintain and use a double record consisting of both a paper-based and electronic part. A detailed field study of the use of such a double record in a large university hospital revealed that a paper-based medical record is key in the subtle coordination inside the ward as its physical form helps to both achieve local coordination and awareness, as well as facilitates micro- and macro-mobility. These results echo previous findings, but our study highlighted that these paper-based affordances are not transferred to the electronic medical record used in the hospital. Specifically, our study showed that managing the dual record introduces a number of configuration challenges related to managing, synchronizing, communicating and cross-referencing both versions of the record. Different technologies for bridging the gab between paper-based and digital records have been proposed and we did a detailed study of one particular technology called the HyPR device. The study was conducted in a simulated medical ward environment with 8 clinicians performing a set of scenarios. Although the study was limited to 8 clinicians, it already showed that the HyPR approach has the potential to function as a transitional artifact that helps integrate and synchronize paper-based and digital information while maintaining some of the benefits from both the paper-based and digital records. Based on these two studies we introduced the concept of collaborative affordances, which denotes a set of properties of physical devices and artifacts that supports collaboration. These collaborative affordance include: mobility and portability, collocated access, shared overview and mutual awareness. The concept of collaborative affordances can be used in the analysis and design of collaborative technologies.

ACKNOWLEDGMENTS

This work was supported by the EU Marie Curie Network iCareNet grant number 264738. Special thanks to Sanne Jensen for facilitating the medical simulation study. Mathias Schmidt, Thomas Snaidero and Vaidas Sirtautas helped implement and test the HyPR technology. We thank our anonymous reviewers, Evi Vlassaks and Paolo Tell for their feedback and helpful suggestions for the improvement of the manuscript.

REFERENCES

1. Ahmed, K., Jawad, M., Abboudi, M., Gavazzi, A., Darzi, A., Athanasiou, T., Vale, J., Khan, M. S., and Dasgupta, P. Effectiveness of procedural simulation in urology: a systematic review. *The Journal of urology 186*, 1 (2011), 26–34.
2. Arai, T., Aust, D., and Hudson, S. E. Paperlink: a technique for hyperlinking from real paper to electronic content. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, ACM (1997), 327–334.
3. Bång, M., Larsson, A., and Eriksson, H. Nostos: A paper–based ubiquitous computing healthcare environment to support data capture and collaboration. In *AMIA Annual Symposium Proceedings*, vol. 2003, American Medical Informatics Association (2003), 46.
4. Bardram, J. E., and Bossen, C. Mobility work: The spatial dimension of collaboration at a hospital. *Computer Supported Cooperative Work (CSCW) 14*, 2 (2005), 131–160.
5. Bardram, J. E., and Bossen, C. A web of coordinative artifacts: collaborative work at a hospital ward. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, ACM (2005), 168–176.
6. Bardram, J. E., and Hansen, T. R. Context-based workplace awareness. *Computer Supported Cooperative Work (CSCW) 19*, 2 (2010), 105–138.
7. Berg, M. Accumulating and coordinating: occasions for information technologies in medical work. *Computer Supported Cooperative Work (CSCW) 8*, 4 (1999), 373–401.
8. Berger, R. G., and Kichak, J. Computerized physician order entry: helpful or harmful? *Journal of the American Medical Informatics Association 11*, 2 (2004), 100–103.
9. Bossen, C., and Jensen, L. G. How physicians' achieve overview': a case-based study in a hospital ward. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, ACM (2014), 257–268.
10. Cabitza, F., Sarini, M., Simone, C., and Telaro, M. When once is not enough: the role of redundancy in a hospital ward setting. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, ACM (2005), 158–167.
11. Chantler, C., Clarke, T., and Granger, R. Information technology in the english national health service. *JAMA 296*, 18 (2006), 2255–2258.
12. Chen, Y. Documenting transitional information in emr. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2010), 1787–1796.
13. Dykstra, R. H., Ash, J. S., Campbell, E., Sittig, D. F., Guappone, K., Carpenter, J., Richardson, J., Wright, A., and McMullen, C. Persistent paper: the myth of "going paperless". In *AMIA Annual Symposium Proceedings*, vol. 2009, American Medical Informatics Association (2009), 158.
14. Ellingsen, G., and Monteiro, E. A patchwork planet integration and cooperation in hospitals. *Computer Supported Cooperative Work (CSCW) 12*, 1 (2003), 71–95.
15. Fitzpatrick, G. Integrated care and the working record. *Health Informatics Journal 10*, 4 (2004), 291–302.
16. Fitzpatrick, G., and Ellingsen, G. A review of 25 years of cscw research in healthcare: contributions, challenges and future agendas. *Computer Supported Cooperative Work (CSCW) 22*, 4-6 (2013), 609–665.
17. Gaver, W. W. Technology affordances. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (1991), 79–84.
18. Gibson, J. J. The theory of affordances. *Hilldale, USA* (1977).
19. Greenhalgh, T., Potts, H. W., Wong, G., Bark, P., and Swinglehurst, D. Tensions and paradoxes in electronic patient record research: A systematic literature review using the meta-narrative method. *Milbank Quarterly 87*, 4 (2009), 729–788.
20. Guimbretière, F. Paper augmented digital documents. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, ACM (2003), 51–60.
21. Harper, R., O'Hara, K., Sellen, A., and Duthie, D. Toward the paperless hospital? *British journal of anaesthesia 78*, 6 (1997), 762–767.
22. Heath, C., Svensson, M. S., Hindmarsh, J., Luff, P., and Vom Lehn, D. Configuring awareness. *Computer Supported Cooperative Work (CSCW) 11*, 3-4 (2002), 317–347.
23. Houben, S., Frost, M., and Bardram, J. Hypr device: Mobile support for hybrid patient records. In *Proceedings of the 2014 ACM international conference on Interactive tabletops and surfaces*, ACM (2014).
24. Jaspers, M. W. A comparison of usability methods for testing interactive health technologies: methodological aspects and empirical evidence. *International journal of medical informatics 78*, 5 (2009), 340–353.
25. Kaptelinin, V., and Nardi, B. Affordances in hci: toward a mediated action perspective. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2012), 967–976.
26. Kreijns, K., and Kirschner, P. A. The social affordances of computer-supported collaborative learning environments. In *Frontiers in Education Conference, 2001. 31st Annual*, vol. 1, IEEE (2001), T1F–12.

27. Liao, C., Guimbretière, F., and Hinckley, K. Papiercraft: a command system for interactive paper. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, ACM (2005), 241–244.

28. Luff, P., and Heath, C. Mobility in collaboration. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work*, ACM (1998), 305–314.

29. Norman, D. A. *The psychology of everyday things*. Basic books, 1988.

30. Norman, D. A. Affordance, conventions, and design. *interactions 6*, 3 (1999), 38–43.

31. Norman, D. A. The way i see it signifiers, not affordances. *interactions 15*, 6 (2008), 18–19.

32. Østerlund, C. S. Documents in place: demarcating places for collaboration in healthcare settings. *Computer Supported Cooperative Work (CSCW) 17*, 2-3 (2008), 195–225.

33. Park, S. Y., and Chen, Y. Adaptation as design: Learning from an emr deployment study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2012), 2097–2106.

34. Saleem, J. J., Russ, A. L., Justice, C. F., Hagg, H., Ebright, P. R., Woodbridge, P. A., and Doebbeling, B. N. Exploring the persistence of paper with the electronic health record. *International journal of medical informatics 78*, 9 (2009), 618–628.

35. Sands, D. Z., Rind, D. M., Vieira, C., and Safran, C. Can a large institution go paperless? *Studies in health technology and informatics 52* (1997), 60–63.

36. Sellen, A. J., and Harper, R. H. *The myth of the paperless office*. MIT press, 2002.

37. Tang, C., and Carpendale, S. Evaluating the deployment of a mobile technology in a hospital ward. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, ACM (2008), 205–214.

38. Vyas, D., Chisalita, C. M., and Van Der Veer, G. C. Affordance in interaction. In *Proceedings of the 13th Eurpoean conference on Cognitive ergonomics: trust and control in complex socio-technical systems*, ACM (2006), 92–99.

39. Weibel, N., Ispas, A., Signer, B., and Norrie, M. C. Paperproof: a paper-digital proof-editing system. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems*, ACM (2008), 2349–2354.

40. Winkler, C., Seifert, J., Reinartz, C., Krahmer, P., and Rukzio, E. Penbook: Bringing pen+ paper interaction to a tablet device to facilitate paper-based workflows in the hospital domain. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces*, ACM (2013), 283–286.

41. Zamarripa, M. S., Gonzalez, V. M., and Favela, J. The augmented patient chart: Seamless integration of physical and digital artifacts for hospital work. In *Universal Access in Human-Computer Interaction. Applications and Services*. Springer, 2007, 1006–1015.

42. Zhang, J., and Patel, V. L. Distributed cognition, representation, and affordance. *Pragmatics & Cognition 14*, 2 (2006), 333–341.

43. Zhou, X., Ackerman, M. S., and Zheng, K. I just don't know why it's gone: maintaining informal information use in inpatient care. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2009), 2061–2070.

Appendices

Co-Author Statement

I hereby declare that I am aware that the work in the paper:

Title of paper
Activity-Centric Support for Ad Hoc Knowledge Work – A Case Study of co-Activity Manager

Authors:
Steven Houben, *Jakob E. Bardram, Jo Vermeulen, Kris Luyten and Karin Coninx*

Published:
In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems ***(CHI '13)****. ACM, New York, NY, USA, 2263-2272.*

of which I am a co-author, will form part of the PhD dissertation by **Steven Houben** who made a:

X	Major Contribution
	Proportional Contribution
	Minor Contribution

Name: JO VERMEULEN

Date: 06/10/2014

Signature: [signature]

Co-Author Statement

I hereby declare that I am aware that the work in the paper:

Title of paper
Activity-Centric Support for Ad Hoc Knowledge Work – A Case Study of co-Activity Manager

Authors:
Steven Houben, *Jakob E. Bardram, Jo Vermeulen, Kris Luyten and Karin Coninx*

Published:
In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems ***(CHI '13)****. ACM, New York, NY, USA, 2263-2272.*

of which I am a co-author, will form part of the PhD dissertation by **Steven Houben** who made a:

X	Major Contribution
	Proportional Contribution
	Minor Contribution

Name: Kris Luyten

Date: 06/10/2014

Signature:

Co-Author Statement

I hereby declare that I am aware that the work in the paper:

Title of paper
Activity-Centric Support for Ad Hoc Knowledge Work – A Case Study of co-Activity Manager

Authors:
Steven Houben, *Jakob E. Bardram, Jo Vermeulen, Kris Luyten and Karin Coninx*

Published:
In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems ***(CHI '13)****. ACM, New York, NY, USA, 2263-2272.*

of which I am a co-author, will form part of the PhD dissertation by **Steven Houben** who made a:

X	Major Contribution
	Proportional Contribution
	Minor Contribution

Name: CONINX KARIN

Date: 6/10/'14

Signature: Coninx Karin

Co-Author Statement

I hereby declare that I am aware that the work in the paper:

Title of paper
Activity-Centric Support for Ad Hoc Knowledge Work – A Case Study of co-Activity Manager

Authors:
Steven Houben, *Jakob E. Bardram, Jo Vermeulen, Kris Luyten and Karin Coninx*

Published:
In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems ***(CHI '13)****. ACM, New York, NY, USA, 2263-2272.*

of which I am a co-author, will form part of the PhD dissertation by **Steven Houben** who made a:

X	Major Contribution
	Proportional Contribution
	Minor Contribution

Name: Jakob Bardram

Date: 8/10 2014

Signature:

Co-Author Statement

I hereby declare that I am aware that the work in the paper:

Title of paper
NooSphere: An Activity-Centric Infrastructure for Distributed Interaction

Authors:
Steven Houben, *Søren Nielsen, Morten Esbensen and Jakob E. Bardram*

Published:
*In Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia **(MUM '13)**. ACM, New York, NY, USA, Article 13.*

of which I am a co-author, will form part of the PhD dissertation by **Steven Houben** who made a:

X	Major Contribution
	Proportional Contribution
	Minor Contribution

Name: Søren Nielsen

Date: 06-10-2014

Signature:

Co-Author Statement

I hereby declare that I am aware that the work in the paper:

Title of paper
NooSphere: An Activity-Centric Infrastructure for Distributed Interaction

Authors:
Steven Houben, *Søren Nielsen, Morten Esbensen and Jakob E. Bardram*

Published:
*In Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia **(MUM '13)**. ACM, New York, NY, USA, Article 13.*

of which I am a co-author, will form part of the PhD dissertation by **Steven Houben** who made a:

X	Major Contribution
	Proportional Contribution
	Minor Contribution

Name: Morten Esbensen

Date: 06/10/2014

Signature: 

Co-Author Statement

I hereby declare that I am aware that the work in the paper:

Title of paper
NooSphere: An Activity-Centric Infrastructure for Distributed Interaction

Authors:
Steven Houben, *Søren Nielsen, Morten Esbensen and Jakob E. Bardram*

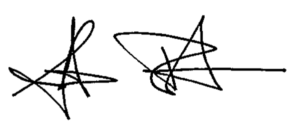
Published:
*In Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia **(MUM '13)**. ACM, New York, NY, USA, Article 13.*

of which I am a co-author, will form part of the PhD dissertation by **Steven Houben** who made a:

X	Major Contribution
	Proportional Contribution
	Minor Contribution

Name: Jakob Bardram

Date: 7/10 2014

Signature: 

Co-Author Statement

I hereby declare that I am aware that the work in the paper:

Title of paper
ActivityDesk: Multi-Device Configuration Work using an Interactive Desk

Authors:
Steven Houben *and Jakob E. Bardram*

Published:
In CHI '13 Extended Abstracts on Human Factors in Computing Systems **(CHI EA '13)**. *ACM, New York,NY, USA, 721-726.*

of which I am a co-author, will form part of the PhD dissertation by **Steven Houben** who made a:

X	Major Contribution
	Proportional Contribution
	Minor Contribution

Name: Jakob Bardram

Date: 7/10 2014

Signature:

Co-Author Statement

I hereby declare that I am aware that the work in the paper:

Title of paper
ActivitySpace: Managing Device Ecologies in an Activity-Centric Configuration Space

Authors:
Steven Houben, *Paolo Tell and Jakob E. Bardram*

Published:
In Proceedings of the 2014 ACM international conference on Interactive tabletops and surfaces **(*ITS '14*)**. *ACM, 2014.*

of which I am a co-author, will form part of the PhD dissertation by **Steven Houben** who made a:

X	Major Contribution
	Proportional Contribution
	Minor Contribution

Name: Jakob Bardram

Date: 9/10 2014

Signature:

Co-Author Statement

I hereby declare that I am aware that the work in the paper:

Title of paper
ActivitySpace: Managing Device Ecologies in an Activity-Centric Configuration Space

Authors:
Steven Houben, *Paolo Tell and Jakob E. Bardram*

Published:
In Proceedings of the 2014 ACM international conference on Interactive tabletops and surfaces ***(ITS '14)****. ACM, 2014.*

of which I am a co-author, will form part of the PhD dissertation by **Steven Houben** who made a:

X	Major Contribution
	Proportional Contribution
	Minor Contribution

Name: Paolo Tell

Date: 06. oct. 2014

Signature:

Co-Author Statement

I hereby declare that I am aware that the work in the paper:

Title of paper
WatchConnect: A Toolkit for Prototyping Smartwatch-Based Cross-Device Applications

Authors:
Steven Houben *and Nicolai Marquardt*

Published:
*Submitted for review to the SIGCHI Conference on Human Factors in Computing Systems **(CHI '15)**.*

of which I am a co-author, will form part of the PhD dissertation by **Steven Houben** who made a:

X	Major Contribution
	Proportional Contribution
	Minor Contribution

Name: Nicolai Marquardt

Date: November 22nd, 2014

Signature:

Co-Author Statement

I hereby declare that I am aware that the work in the paper:

Title of paper
HyPR Device: Mobile Support for Hybrid Patient Records

Authors:
Steven Houben, *Mads Frost and Jakob E. Bardram*

Published:
In Proceedings of the 2014 ACM international conference on Interactive tabletops and surfaces ***ITS '14****. ACM, 2014.*

of which I am a co-author, will form part of the PhD dissertation by **Steven Houben** who made a:

X	Major Contribution
	Proportional Contribution
	Minor Contribution

Name: Jakob Bardram

Date: 9/10 2014

Signature:

Co-Author Statement

I hereby declare that I am aware that the work in the paper:

Title of paper
HyPR Device: Mobile Support for Hybrid Patient Records

Authors:
Steven Houben, *Mads Frost and Jakob E. Bardram*

Published:
In Proceedings of the 2014 ACM international conference on Interactive tabletops and surfaces ***ITS '14****. ACM, 2014.*

of which I am a co-author, will form part of the PhD dissertation by **Steven Houben** who made a:

X	Major Contribution
	Proportional Contribution
	Minor Contribution

Name: Mads Frost

Date: 9/10-2014

Signature:

Co-Author Statement

I hereby declare that I am aware that the work in the paper:

Title of paper
Collaborative Affordances of Hybrid Patient Record Technologies in Medical Work

Authors:
Steven Houben, *Mads Frost and Jakob E. Bardram*

Published:
To appear in the proceedings of the 2015 Conference on Computer-Supported Cooperative Work and Social Computing ***CSCW '15****. ACM, 2015.*

of which I am a co-author, will form part of the PhD dissertation by **Steven Houben** who made a:

X	Major Contribution
	Proportional Contribution
	Minor Contribution

Name: Jakob Bardram

Date: 7/10 2014

Signature:

Co-Author Statement

I hereby declare that I am aware that the work in the paper:

Title of paper
Collaborative Affordances of Hybrid Patient Record Technologies in Medical Work

Authors:
Steven Houben, *Mads Frost and Jakob E. Bardram*

Published:
To appear in the proceedings of the 2015 Conference on Computer-Supported Cooperative Work and Social Computing ***CSCW '15****. ACM, 2015.*

of which I am a co-author, will form part of the PhD dissertation by **Steven Houben** who made a:

X	Major Contribution
	Proportional Contribution
	Minor Contribution

Name: Mads Frost

Date: 9/10-2014

Signature: