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

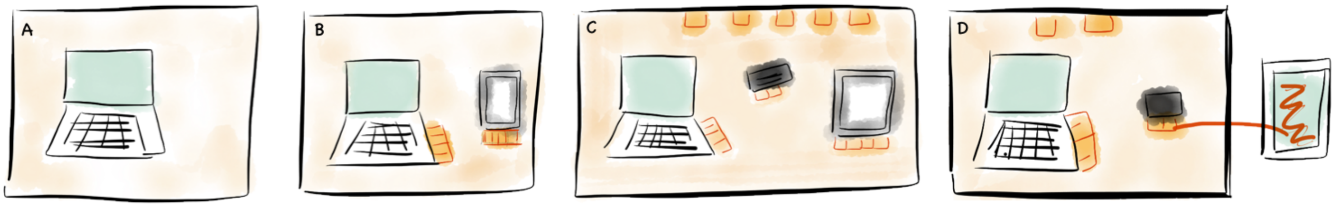


Figure 1. The configuration space is activated by placing the master device in the space (A). All other devices that are added to the space become part of the same activity space (B). The space visualizes the resource attached to each device (B). Resources can be moved between devices or stored in the 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 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.

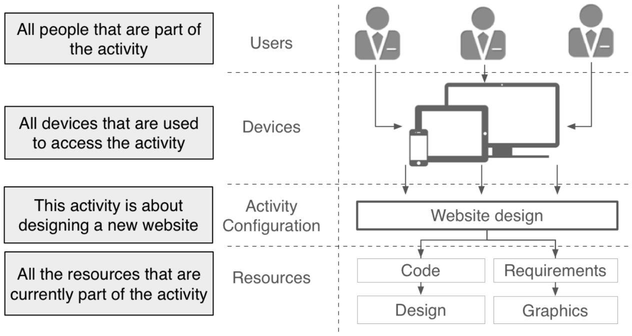

All people that are part of the activity
Users
All devices that are used to access the activity
Devices
This activity is about designing a new website
Activity Configuration
Website design
All the resources that are currently part of the activity
Resources
Code
Requirements
Design
Graphics

Figure 3. Activity configurations allow multiple users to access the encapsulated resources and services using multiple devices.

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.

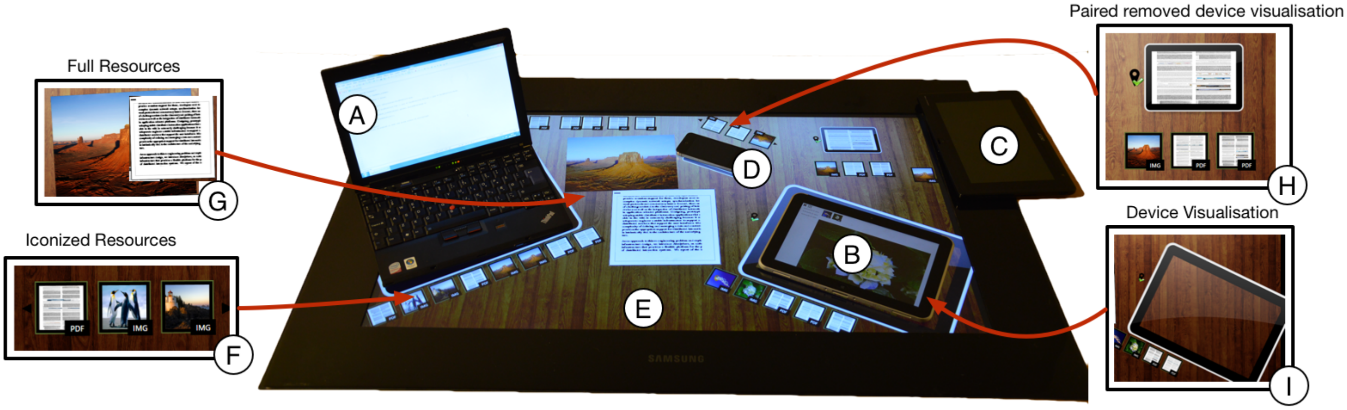


Figure 2. ***ActivitySpace*** **supports activity-centric resource management spanning across (A) laptops, (B-C) tablets and (D) phones using an (E) interactive desk as mediating configuration space. The configuration space visualizes all devices (H-I) and their allocated resources (F-G), that are part 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

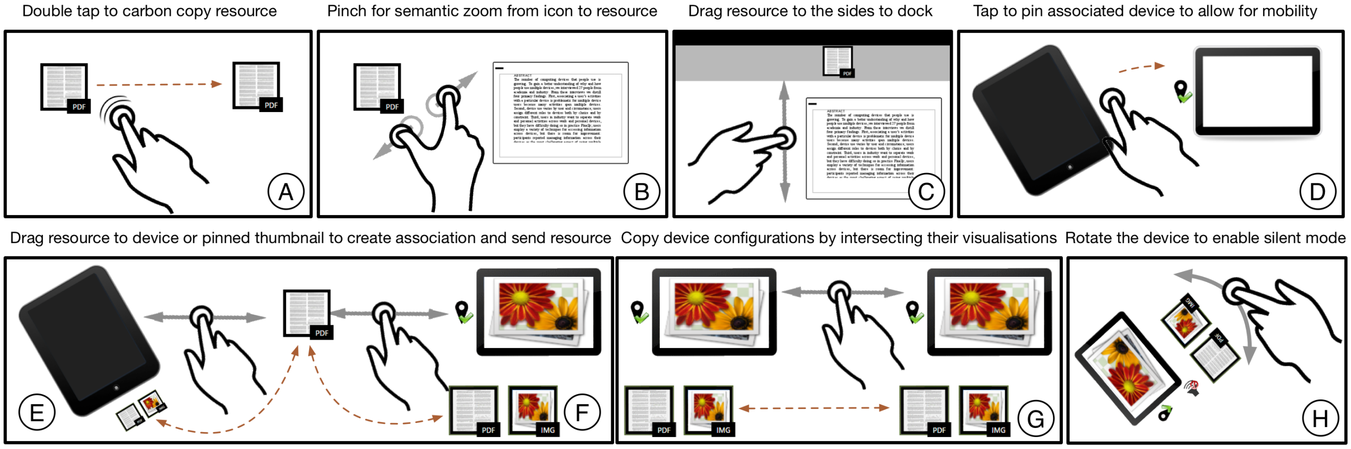
ActivitySpace is built around a personal interactive surface that works as the *configuration spaces* (Figure 2E). This space allows users to (i) easily add or remove devices to the ongoing work context, (ii) fragment resources across available devices, and (iii) save and restore such cross-device configurations 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-


Double tap to carbon copy resource

Pinch for semantic zoom from icon to resource

Drag resource to the sides to dock

Tap to pin associated device to allow for mobility

Drag resource to device or pinned thumbnail to create association and send resource

Copy device configurations by intersecting their visualisations

Rotate the device to enable silent mode

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.

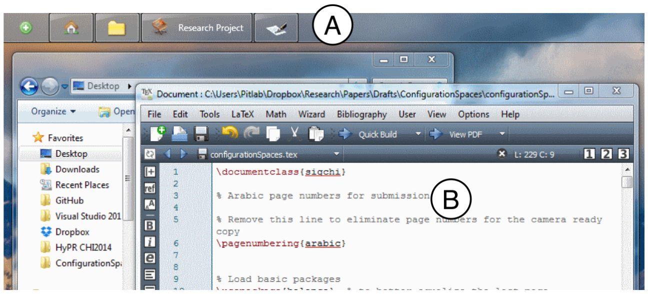
G : Reconfiguration – To support fast and easy device reconfiguration, entire device configurations can be copied by bumping either the real devices or their visualizations together. The configuration is copied from the device that is moved first to the second device. This allows multiple users to quickly 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-



Document : C:\Users\Pitlab\Dropbox\Research\Papers\Drafts\ConfigurationSpaces\configurationSp...

```
\documentclass{sigchi}

% Arabic page numbers for submission

% Remove this line to eliminate page numbers for the camera ready
copy
\pagenumbering{arabic}


% Load basic packages
```

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.

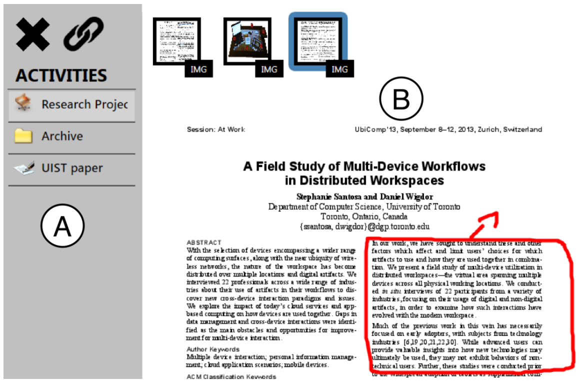
ties using the activity sidebar on the tablet (master or isolated mode) or connect the tablet to the activity node running on the configuration space (slave mode).

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 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.



Session: At Work

UbiComp'13, September 8–12, 2013, Zurich, Switzerland

A Field Study of Multi-Device Workflows in Distributed Workspaces

Stephanie Santosa and Daniel Wigdor
Department of Computer Science, University of Toronto
Toronto, Ontario, Canada
{ssantosa, dwigdor}@dgp.toronto.edu

ABSTRACT

With the selection of devices encompassing a wider range of computing surfaces, along with the near ubiquity of wireless networks, the nature of the workspace has become distributed over multiple locations and digital artifacts. We interviewed 22 professionals across a wide range of industries about their use of artifacts in their workflows to discover new cross-device interaction paradigms and issues. We explore the impact of today's cloud services and app-based computing on how devices are used together. Gaps in data management and cross-device interactions were identified as the main obstacles and opportunities for improvement for multi-device interaction.

Author Keywords

Multiple device interaction; personal information management; cloud application scenarios; mobile devices.

ACM Classification Keywords

In our work, we have sought to understand these and other factors which affect and limit users' choices for which artifacts to use and how they are used together in combination. We present a field study of multi-device utilization in distributed workspaces—the virtual area spanning multiple devices across all physical working locations. We conducted *in situ* interviews of 22 participants from a variety of industries, focusing on their usage of digital and non-digital artifacts, in order to examine how such interactions have evolved with the modern workspace.

Much of the previous work in this vein has necessarily focused on early adopters, with subjects from technology industries [6,19,20,21,22,30]. While advanced users can provide valuable insights into how new technologies may ultimately be used, they may not exhibit behaviors of non-technical users. Further, these studies were conducted prior

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.

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:

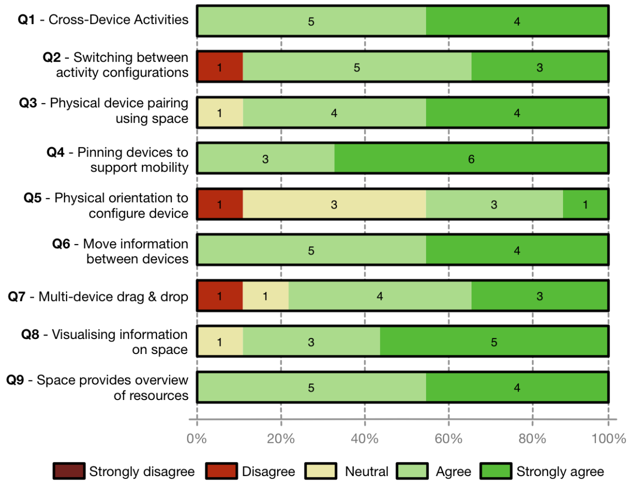


Figure 7. The results of the a 5-point Likert-scale questionnaire on the usefulness of the different parts of the system. The numbers in the bar represent the amount of participants.

> – *"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".

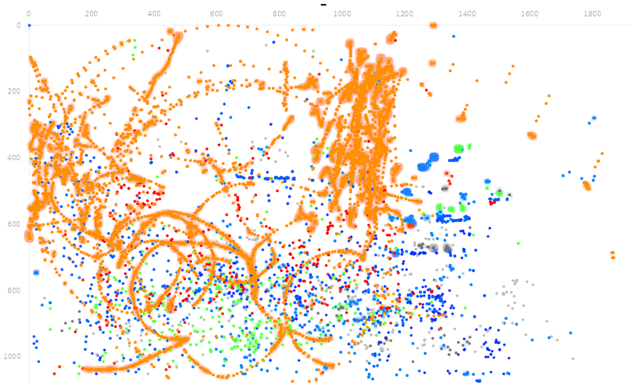


Figure 8. Touch interaction data of 9 users plotted on the desk space. The data shows circular movements of resources around visualisations (orange color). Other colors represent the touch input of participants which is primarily focussed in the front middle of the desk.

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.