A Cross-Device Drag-and-Drop Technique

Adalberto L. Simeone
Lancaster University
Lancaster, UK
a.simeone@lancaster.ac.uk

Julian Seifert
Ulm University
Ulm, Germany
julian.seifert@uni-ulm.de

Dominik Schmidt
Hasso Plattner Institute
Potsdam, Germany
dominik.schmidt@hpi.uni-potsdam.de

Paul Holleis
DOCOMO Euro Labs
Munich, Germany
holleis@docomo-euro.com

Enrico Rukzio
Ulm University
Ulm, Germany
enrico.rukzio@uni-ulm.de

Hans Gellersen
Lancaster University
Lancaster, UK
hwg@lancaster.ac.uk

ABSTRACT

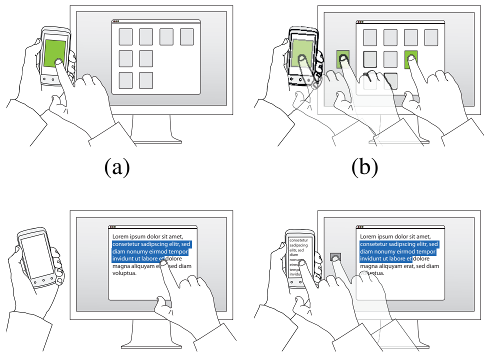
Many interactions naturally extend across smart-phones and devices with larger screens. Indeed, data might be received on the mobile but more conveniently processed with an application on a larger device, or vice versa. Such interactions require spontaneous data transfer from a source location on one screen to a target location on the other device. We introduce a cross-device *Drag-and-Drop* technique to facilitate these interactions involving multiple touchscreen devices, with minimal effort for the user. The technique is a two-handed gesture, where one hand is used to suitably align the mobile phone with the larger screen, while the other is used to select and drag an object between devices and choose which application should receive the data.

Categories and Subject Descriptors

H5.2 [**Information interfaces and presentation**]: User Interfaces - Graphical user interfaces

Keywords

Mobile devices; data transfer; drag-and drop



(a) (b)

(c) (d)

Lorem ipsum dolor sit amet,
consetetur sadipscing elitr, sed
diam nonumy eirmod tempor
invidunt ut labore et dolore
magna aliquyam erat, sed diam
voluptua.

Figure 1: Drag-and-Drop concept: (a) a user holds the mobile phone next to the desktop screen and selects a data item. (b) The user drags it inside the screen. (c) In the other direction, a user selects data on the PC and (d) drops it on the phone.

1. INTRODUCTION

Interactions frequently extend beyond a single device. A phone number is more easily searched on a larger screen, but once found the call is issued with the mobile. A photo can be quickly snapped with a mobile, but its integration in a document is easier on a larger screen. Friends text us place names on our mobiles, but route directions can be better looked up and printed from a larger device. We might also look up an address on a desktop screen, and then use it on our mobiles to navigate to the location. All these are examples of interactions that require users to select data on one device and apply it to another. In this situation, current practices hinder this process by adding extra steps that divert users from the primary goal of applying data from one device to another. Frequently, users recur to typing numbers, names or addresses off a screen because transfer via a sharing protocol is more cumbersome. This type of interaction between mobile and situated devices has been widely studied [9, 10, 11], including recent work focused on interaction with mobile phones as these have become data-rich devices [5, 12, 13]. However, we are concerned with cross-device interactions that have distinct characteristics. The interaction is spontaneous, and the data concerned frequently only emerges during interaction, for example as a result of a search, or of a communication received. The data does not dictate the application, and users might want to use data items in different ways, for instance apply contact data to an address book, navigation tool, or phone application, depending on their interaction goals.

In a previous study we sought to discover what methods users currently adopt to face these needs and which ones they would instead prefer using, if there were no technological limitations and anything was possible. In this paper, we contribute a cross-device *Drag-and-Drop* technique for spontaneous interaction between phones and touchscreens, one of the methods that received the most suggestions in the study. Drag-and-Drop is conceived as a two-handed interaction, with one hand holding the phone and mov-

MUM '13, Dec 02-05 2013, Luleå, Sweden
Copyright 2013 ACM 978-1-4503-2648-3/13/12
http://dx.doi.org/10.1145/2541831.2541848 ...$15.00.

ing it to align with the touchscreen, while the other is used to select and drag an object by touch from one device to the other, where it is dropped on a target application. Figure 1 illustrates the concept: showing selection of an object on the phone (a) and drag-and-drop transfer to a desktop application (b), and selection of text on the desktop (c) that is then dragged and dropped onto the phone (d). We have implemented a prototype system that demonstrates Drag-and-Drop in two ways: with a generic interface and with a custom application. The generic interface is realized with a *Bridge* application on the mobile phone that detects objects dragged onto the phone and provides an interface through which users can drop objects on the available target applications. In addition, we have also implemented a custom email application to illustrate how Drag-and-Drop might be used if it were natively supported. Finally, we report observations on user feedback of the deployed system.

2. RELATED WORK

A range of techniques have been developed to allow users to transfer data between mobile devices and situated computers. Most mechanisms support fine-grained selection of objects on the source device but only coarse-grained selection of a target device. Assuming device discovery in the background, users can select target devices by name and initiate transfer of selected objects. On the target device, the objects are delivered to a default location, which may be the application used for transfer (e.g., Email) or a dedicated dropbox, from where users can pick the data up for further processing. In contrast, Drag-and-Drop is designed for direct point-to-point transfer from a location on one device to a specific target on the receiving device.

Early work that demonstrated point-to-point interaction across devices includes Pick-and-Drop [10] and HyperDrag [11], that relied on instrumentation of the devices or environment. Recent work includes Touch&Interact [5] and PhoneTouch [12], both enabling users to select objects on a phone and to transfer it to a target location on a touchscreen, by directly touching the screen with the phone. However, data transfer from the screen to the phone is generic and requires additional interaction for transfer to a specific target on the phone. Other recent work has been geared towards interaction over a distance, for example by combining touch on a smartphone with pointing at a remote screen to facilitate data transfer to a specific target location on the screen, and vice versa [2]. However, this is limited to environments instrumented for gesture tracking. Close to our work is also Mistry *et al.*'s "SPARSH" concept, envisioning touch-based pick-and-drop between devices [8]. Drag-and-Drop differs from SPARSH as it can integrate with existing technologies, without needing any external cloud-based service.

As a technique, Drag-and-Drop is characterized by spontaneous alignment of phone and screen for interaction, and extension of drag-and-drop to work transparently across the temporarily aligned devices. Transparent drag-and-drop across aligned devices was previously demonstrated by Hinckley et al. [6] in work that investigated pen interaction across multiple displays. Their work emphasized the "stitching" of the involved devices to create a combined display space for cross-device interaction, while we designed our technique for spontaneous transfer where the alignment of the devices is fast and just long enough to facilitate the continuous dragging gesture. Alignment of devices for cross-device interaction has also been investigated in work on proximal interaction [7] and direct touch interaction with phones on touch surfaces [12].

In our work, drag-and-drop is extended transparently across devices. Other work has explored techniques for settings in which the target is out of reach. This includes Drag-and-Pop and Drag-and-Pick, which have been investigated for use on wall-size displays but also demonstrated for interaction across multiple touchscreens [1, 3]. Our technique exploits two-handedness in an asymmetric manner. Guiard's Kinematic Chain model is therefore of key relevance to our work, describing use of the non-dominant hand for setting the frame of reference in which the dominant hand operates [4]. In our case, the non-dominant hand is used to roughly align phone and screen, thus setting the frame of reference for the finer-grained drag-and-drop movement with the other hand.

3. DESIGN OF DRAG-AND-DROP

On a conceptual level the *Drag-and-Drop* technique allows users to drag data from a mobile device and drop it into a desktop screen and vice versa. It is performed through one single uninterrupted touch gesture and hence, it requires that both screens support touch input. The technique itself is articulated into four contiguous phases: placing the mobile device in proximity of the desktop screen (1), identifying the source object of the drag-and-drop action (2), performing the actual movement into the other screen (3) and finally, identifying the target destination where the content being dragged will be "dropped" (4). The first phase builds on the assumption that placing the mobile device next to the screen will form an ideal "bond" between them allowing users to perceive the two different screens as contiguous. This requirement is mainly needed to instil the idea that it is possible to drag data "outside" the physical boundaries of one screen and inside another one due to their close proximity to each other. In fact, the devices have no knowledge of their respective locations and do not need to: it makes the way it works seem more easily understandable by first-time users. Although we have mainly focused on the technique itself, this "bonding" moment could be also used to establish the pairing itself between the devices. In our implementation we used an explicit point-to-point connection which has to be initiated before each session and lasts until the mobile device is explicitly disconnected. Sensors placed on the sides of the desktop screen could streamline the process.

4. IMPLEMENTATION

Our prototype implementation consists of a server application on the PC and two different clients on the mobile phone, communicating over a wireless network. The desktop, a Windows 7 PC connected to a 23" HP2310ti touch screen, runs the *PC Drag Detector*, an application that handles touch detection and networking. The mobile phone, an Android 4.0.3 device, alternatively runs one of the two client applications which address different usage scenarios: first, the *Bridge Application* enables existent applications on the phone (e.g., a map applications or image galleries) to use Drag-and-Drop without modifications; second, the *email client* demonstrates an application extended by Drag-and-Drop to simulate a possible future native integration of our technique by the operative system.

4.1 The PC Drag Detector

The PC Drag Detector is a background application that enables Drag-and-Drop on the desktop. It captures drag events at the system level (i.e., independent of any specific desktop application). When a drag gesture is detected, it displays a *detector window* at the sides of the screen. Once the user enters or exits the screen, the relevant window captures outgoing or incoming data (Figure 2b). Depending on the drag direction, data is sent over the network to the phone or received from it to complete the drag gesture on the desktop. After which the detector window is hidden.

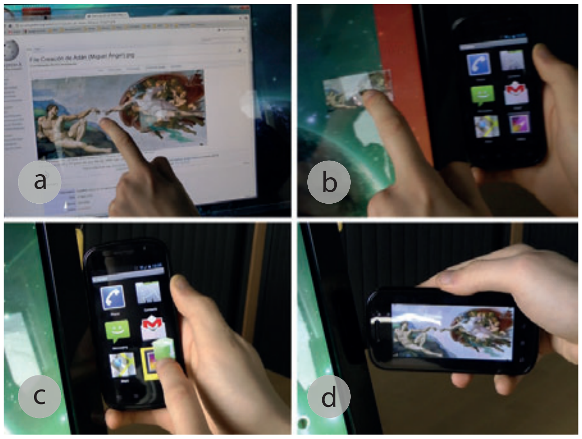


Figure 2: Picture transfer from desktop to phone: the user selects a picture displayed in a webpage (a) and proceeds to drag it across the screen and over the semi-transparent *detector window* (b); the drag gesture is continued on to the mobile screen where the user drops the picture over the viewer icon (c); finally, the picture is automatically displayed (d). The technique also works in the opposite direction.

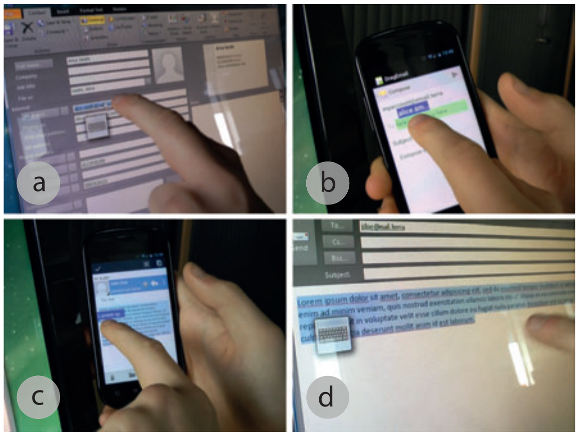


Figure 3: To send an email address available on a PC to a mobile email client, the user can drag it towards the phone (a); once on the mobile, the user can drop it over the relevant field. To copy text available on a mobile to a desktop application, the user selects it normally on the mobile and drags it towards the screen (c); once on the desktop screen, the user can release the finger over any application accepting text to apply it there.

4.2 The Bridge Application

The bridge application, the first of our two mobile implementations, connects Drag-and-Drop to existent applications on the same phone. It handles bidirectional interactions between desktop and mobile, using two different UIs described in the following.

Receiving Data from the Desktop After starting the bridge application, it presents a grid of icons to the user, each representing a distinct application class. Together, these six classes address a range of typical usage scenarios (Figure 2c). In particular, we are covering 1) phone diallers, 2) contact managers, 3) text messaging, 4) email clients, 5) maps and 6) picture viewers. For example, after finding a restaurant while browsing the web on the desktop, the user can simply select the corresponding address and drag it onto the *maps* icon on the phone to immediately start the navigation. By applying data onto a particular application class, the user disambiguates how the phone handles the corresponding data. This basic concept readily extends to other use cases. For instance, to call a number displayed in a desktop application, the user applies it to the diallers icon to immediately get connected. In doing so, Drag-and-Drop offers a quick and convenient alternative to manually transferring the required information from the desktop to the mobile phone, for example by re-typing it.

From a technical point of view, once the user starts a drag gesture on the desktop, the finger will eventually enter a drag detector window. The PC application then queries the data to discern its underlying type and provides feedback on the mobile once the finger enters its screen. Then, an icon representing the dragged data is displayed as a preview and moves alongside the user's finger (Figure 2c). Feedback about the validity of the drop location on the mobile is shown to the user through background changes of the currently highlighted application class (i.e., yellow for a valid drop target, red otherwise). When the finger is released, the dragged data is applied to the chosen class. Through a mechanism called *implicit intents*, the Android OS determines eligible applications that can handle the data from those available in the user's device.

Sending Data to the Desktop The Bridge Application also allows dragging items from the phone to the desktop to address situations where data originating from a mobile device is better viewed or edited on a larger screen. In order to support this, we use Android's built-in share feature, a method to internally share data between various applications. In particular, users have to first select the data they wish to apply on another device from within an arbitrary application on the phone (e.g., by opening a picture in the gallery viewer). Second, they invoke the share feature which will bring up the bridge application in *send* mode. Due to platform restrictions users cannot initiate a drag gesture directly from a mobile application. Thus, users are shown a list of icons representing data types. Compatible ones will be highlighted, in situations where data might be treated in different ways (e.g.: a picture sent as a URL or as a binary file). From a technical perspective, once the finger enters the detector window on the desktop, the dragged data will be encapsulated into a simulated local drag-and-drop event, transparent to both the user and the target application on the PC.

4.3 Custom Email Client

To demonstrate how Drag-and-Drop interactions are further simplified with a native integration of our technique, we implemented a second mobile application, a custom email client. This is to simulate a possible future scenario where Drag-and-Drop is directly built into the mobile OS without the requirement of an application in between (i.e., the bridge application). Now, users can perform cross-device drag-and-drop gestures directly between two applications running on different devices, without the need for any intermediate steps. In particular, users can select data from the desktop (e.g., an email address) and apply it to any field on the mobile client (Figure 3 a-b). Users can decide where exactly to drop the data; the backgrounds of the email fields will turn green to indicate that the data can be dropped inside or red otherwise. In the other direction, users can drag data from any email field to the desktop. After performing a long touch on the chosen field, a preview of the data is shown (Figure 3 c-d), ready to be dragged across to the desktop.

5. USER FEEDBACK

We deployed the system to collect feedback concerning the system from 15 participants (3 female) aged between 21 to 38 (M=27, SD=5). We intended to collect users' feedback on the cross-device Drag-and-Drop technique and how they would interact with it. In order to do so we designed two sets of tasks, prompted by an email invitation to a field trip. It exposed users to situations that we believe to be often recurrent in daily life. In each session users would be asked to use our proposed technique in lieu of traditional approaches. Before handing participants a mobile device where the technique was enabled, the investigator demonstrated how it worked. After the study, he interviewed them (semi structured) about their overall experience with the system.

In the first set of tasks, participants were required to confirm their attendance to the event by calling a number stated on the email invitation. By using the Bridge Application, they had to drag the phone number inside the mobile device so that they could use it to place the call. Successively, they had to drag the address of the meeting place to their mobile device, in order to visualize the route before heading out. In the successive set of tasks, participants were told that while they were on the field trip, they had received a work related email containing information that they needed for a report they were working on. They were thus asked to select the part of the email they needed and drag it inside a Word document. Finally, once back from the trip, they were asked to drag a picture they had taken with the phone inside a photo editing application on their PC so that they could enhance it before sending it to their friends.

We observed that almost all participants were able to use the interaction technique from their first attempt even though only few of them had previously used a touch screen to interact with a PC. Most participants indicated that they liked the interaction technique citing its ease of use (stated five times) as one of the main reasons. When asked to compare this approach to current existing practices (such as using a cable to transfer files, sending emails with attachments to oneself, or re-accessing an email on the PC that was first read on a mobile device) they unanimously agreed that if the method was commercially available they would have preferred it over the currently known alternatives. For instance, P4 emphasized that using the technique "felt really natural". Several participants pointed out that they would like to use the technique for transferring music and pictures from and to their phones. P15 indicated that he would like to use this technique for picking up information from public displays. P8 suggested that transferring trough drag-and-drop a web browser session including all open tabs would be really helpful when one has to leave the computer at home and wants to continue to interact with some web pages later. Another suggestion was made by P9: "I wish I could select several objects and drag them at once instead of dragging them one by one."

In summary, the user feedback we gained supports the conceptual advantages of the interaction technique: participants stated they they found it very natural to perform and easy to learn. One user commented (P5): *"Its feels natural and seems to come from the real world. It's like passing a sheet of paper to another person."*. Although we implemented the technique in the context of touch-enabled device, it could be extended to non-touch desktop screens by building on the concept of OS notifications. Once an item is received by a non touch-enabled device (such as a regular desktop PC) it could be displayed as a temporary notification. Users could then use the mouse to drag the item from within the notification window to where it needs to be used.

6. CONCLUSION

In this paper we have presented a novel interaction technique that allows user to intuitively share data between desktop computers and mobile devices. Through touch gestures, the cross-device Drag-and-Drop technique provides a practical solution to a need that is still today not adequately supported. The user feedback we gained highlighted its positive aspects of being easy to learn and perform. We plan to further investigate how to adapt the technique for use in public and semi-public settings such as train stations, airports, museums, retail stores, etc.

7. ACKNOWLEDGEMENTS

This work was supported by the MULTITAG project funded by DOCOMO Euro-Labs. The first author was supported by a Marie Curie Intra European Fellowship within the 7th European Community Framework Programme.

8. REFERENCES

[1] P. Baudisch, E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. Bederson, and A. Zierlinger. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch and pen-operated systems. In *Proc. Interact '03*, pages 57–64. IOS Aress, 2003.

[2] A. Bragdon, R. DeLine, K. Hinckley, and M. R. Morris. Code Space: Touch + Air Gesture Hybrid Interactions For Supporting Developer Meetings. In *Proc. ITS '11*, pages 212–221. ACM, 2011.

[3] M. Collomb and M. Hascoët. Extending drag-and-drop to new interactive environments: A multi-display, multi-instrument and multi-user approach. *Interacting with Computers*, 20(6):562 – 573, 2008.

[4] Y. Guiard. Asymmetric division of labour in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behaviour*, 19:486–517, 1987.

[5] R. Hardy and E. Rukzio. Touch & Interact: Touch-based interaction of mobile phones with displays. In *Proc. MobileHCI*, pages 245–254. ACM, 2008.

[6] K. Hinckley, G. Ramos, F. Guimbretiere, P. Baudisch, and M. Smith. Stitching: Pen Gestures That Span Multiple Displays. In *Proc. AVI '04*, pages 23–31. ACM, 2004.

[7] N. Marquardt, R. Diaz-Marino, S. Boring, and S. Greenberg. The Proximity Toolkit: Prototyping Proxemic Interactions in Ubiquitous Computing Ecologies. In *Proc. UIST '11*, pages 315–326. ACM, 2011.

[8] P. Mistry, S. Nanayakkara, and P. Maes. Touch and Copy, Touch and Paste. In *Proc. CHI EA '11*, pages 1095–1098. ACM, 2011.

[9] B. A. Myers. Using handhelds and PCs together. *Comm. ACM*, 44:34–41, 2001.

[10] J. Rekimoto. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *Proc. UIST '97*, pages 31–39. ACM, 1997.

[11] J. Rekimoto and M. Saitoh. Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. In *Proc. CHI '99*, pages 378–385. ACM, 1999.

[12] D. Schmidt, F. Chehimi, E. Rukzio, and H. Gellersen. PhoneTouch: A Technique for Direct Phone Interaction on Surfaces. In *Proc. UIST '10*, pages 13–16. ACM, 2010.

[13] A. D. Wilson and R. Sarin. BlueTable: Connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In *Proc. GI '07*, pages 119–125, 2007.