Mobile Pen-and-Paper Interaction

Infrastructure Design, Conceptual Frameworks of Interaction and Interaction Theory

Dissertation

genehmigt zur Erlangung des akademischen Grades
eines Doktor-Ingenieurs (Dr.-Ing.)

vorgelegt von
Dipl.-Inform. Farid Hamza Felix Heinrichs
geb. in Frankfurt am Main

am Fachbereich Informatik
der Technischen Universität Darmstadt

Gutachter	Prof. Dr. Max Mühlhäuser (TU Darmstadt)
	Prof. Dr. Johannes Schöning (Universiteit Hasselt)
Tag der Einreichung	26. August 2015
Tag der Disputation	12. November 2015

Darmstadt 2015
D17

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

”In the name of God, the Most Gracious, the Most Merciful”

Abstract

Although smartphones, tablets and other mobile devices become increasingly popular, pen and paper continue to play an important role in mobile settings, such as note taking or creative discussions. However, information on paper documents remains static and usage practices involving sharing, researching, linking or in any other way digitally processing information on paper are hindered by the gap between the digital and physical worlds. A considerable body of research has leveraged digital pen technology in order to overcome this problem with respect to static settings, however, systematically neglecting the mobile domain.

Only recently, several approaches began exploring the mobile domain and developing initial insights into *mobile pen-and-paper interaction* (mPPI), e.g., to publish digital sketches, [Cowan et al., 2011], link paper and digital artifacts, [Pietrzak et al., 2012] or compose music, [Tsandilas, 2012]. However, applications designed to integrate the most common mobile tools pen, paper and mobile devices, thereby combining the benefits of both worlds in a *hybrid mPPI ensemble*, are hindered by the lack of supporting infrastructures and limited theoretical understanding of interaction design in the domain.

This thesis advances the field by contributing a novel infrastructural approach toward supporting mPPI. It allows applications employing digital pen technology in controlling interactive functionality while preserving mobile characteristics of pen and paper. In addition, it contributes a conceptual framework of user interaction in the domain suiting to serve as basis for novel mPPI toolkits. Such toolkits ease development of mPPI solutions by focusing on expressing interaction rather than designing user interfaces by means of rigid widget sets. As such, they provide the link between infrastructure and interaction in the domain. Lastly, this thesis presents a novel, empirically substantiated theory of interaction in hybrid mPPI ensembles. This theory informs interaction design of mPPI, ultimately allowing to develop compelling and engaging interactive systems employing this modality.

Zusammenfassung

Obwohl Smartphones, Tablets und andere mobile Geräte mittlerweile weit verbreitet sind, spielen Stift und Papier weiterhin eine wichtige Rolle in mobilen Situationen, beispielsweise beim Anfertigen von Notizen, oder in kreativen Diskussionen. Information auf Papierdokumenten bleibt dabei jedoch statisch und Verwendungspraktiken, wie beispielsweise teilen, recherchieren, verlinken oder in einer anderen Art Information digital weiterverarbeiten werden durch den Bruch zwischen digitaler und physischer Welt erschwert. Eine ganze Reihe von Forschungsarbeiten nutzt digitale Stifttechnologie um dieses Problem in statischen Situationen zu adressieren, mobile Situationen werden jedoch bislang systematisch vernachlässigt.

Erst in letzter Zeit begannen einige neue Ansätze die mobile Domäne zu explorieren und initiale Einsichten in *mobile Stift-und-Papier Interaktion*, im Englischen *mobile pen-and-paper interaction* (mPPI), zu entwickeln, beispielweise um digitale Sketche zu veröffentlichen, [Cowan et al., 2011], Papier und digitale Artefakte zu verlinken, [Pietrzak et al., 2012], oder Musik zu komponieren, [Tsandilas, 2012]. Derartige Ansätze, entworfen um die meist verbreiteten mobilen Werkzeuge Stift, Papier und mobile Geräte zu integrieren und dabei die Vorteile beider Welten in einem *hybriden mPPI ensemble* zu kombinieren, werden jedoch durch einen Mangel an unterstützenden Infrastrukturen und eingeschränktes theoretisches Verständnis im Hinblick auf Interaktionsdesign in der Domäne gehindert.

Die vorliegende Dissertation stellt einen neuartigen Infrastrukturansatz vor, welcher mPPI direkt unterstützt, und leistet damit einen wichtigen Beitrag zum Voranschreiten dieses Forschungsfelds. Dieser Ansatz erlaubt Applikationen, digitale Stifttechnologie zum Steuern von interaktiver Funktionalität einzusetzen und erhält gleichzeitig mobile Charakteristiken von Stift und Papier. Weiterhin trägt diese Dissertation einen konzeptuellen Rahmen für Stift-und-Papier basierte Nutzerinteraktion bei, welcher dazu geeignet ist als Basis von mPPI Toolkits zu dienen. Derartige Toolkits vereinfachen die Entwicklung von mPPI Lösungen durch Fokussierung auf das Beschreiben von Interaktion, im Gegensatz zur klassischen Fokussierung auf rigide Widget sets. Somit stellen sie das Bindeglied zwischen Infrastruktur und Interaktion dar. Abschließend, stellt diese Dissertation eine neuartige, empirisch fundierte Theorie der Interaktion in hybriden mPPI ensembles vor. Diese Theorie dient zur Unterstützung des Interaktionsdesigns und erlaubt es letztlich, spannende und einfach zu bedienende, interaktive, Stift-und-Papier basierte Systeme zu entwickeln.

Acknowledgements

I am deeply indebted to those people who have made this thesis and its finalization possible. It is owed to the continuous and relentless support of my family, colleagues, advisors, friends and students, who never gave up on me even when I almost did. I want to take this chance to say ”thank you”. Although I cannot name everyone here who had a part, I want to extend my special gratitude to Prof. Dr. Max Mühlhäuser for giving me the chance to embark on this journey, advising me throughout it and putting up the patience for me to finish it. Also, I want to thank Prof. Dr. Johannes Schöning for reviewing this thesis and making its finalization possible. Further, I want to thank my former colleagues and students at the Telecooperation group TU Darmstadt, especially the support team around Elke Halla. Here, another special and warm hearted ”thank you” goes to Dr. Daniel Schreiber, who accompanied me throughout the long journey, taught me the skill of ”just doing it” (among others) and put up incredible patience listening to all my doubts, continuously motivating me. Thanks Daniel, also for reading through the drafts although you are a busy family man and work some place else now, I really appreciate it. Finally, I want to thank my family, in particular my mother, for always being there, always listening to the endless stream of weird technical jargon while at the same time motivating and supporting me. And I want to thank my wife Nadia, without whom this thesis never would have been finished. Thank you for trusting in me, thank you for your patience and support, thank you my sunshine.

Contents

1 Introduction

Synopsis: *In this chapter the stage for the thesis is set by introducing problem domain and research questions. Using a case study of digital grocery shopping support as running example, the concept of mobile pen-and-paper interaction and the underlying motivation is introduced. Based on this, the main research questions are elaborated and employed research methodologies are outlined. Finally, contributions are summarized and an overview of the structure of this thesis is given.*

Mobile devices, such as smartphones and tablets, have become increasingly popular and are among the most widespread contemporary information processing tools. Mobile devices support users in a broad range of applications. They provide ubiquitous access to digital information, compelling and engaging interaction with digital contents and communication beyond traditional boundaries. As such, mobile devices have fundamentally changed our practices with respect to information processing in mobile or nomadic, that is temporarily stationary, contexts.

Despite these advances, however, paper remains one of the most influential tools in mobile contexts. It is neither combating screen readability issues when used in bright daylight, nor does it require battery power or network connectivity in order to operate. It affords instantaneous use, no startup time required. Spatial layout of information on paper provides important clues facilitating recall [Sellen and Harper, 2003]. Paper artifacts can be easily carried and can be easily passed to others. Paper is cheap, robust, light- weight and extremely flexible with respect to usage context.

This applies in particular to tasks related to creativity, social interaction and note-taking, [Harboe and Huang, 2015]. Imagine a creative discussion in the cafeteria with colleagues about a challenging design issue. Chances are that somebody takes out a paper artifact, i.e., a sheet of paper, a paper notebook or even a napkin, to quickly jot down a sketch of relevant ideas. However, the problem later on is: in order to share this sketch with colleagues, it has first to be converted to a digital representation before it can be send to others.

In this example, the user experiences a disruptive *media transition*. A media transition thereby refers to a transition in terms of interaction devices, information available and communication facilities during an activity [Steimle, 2009b]. In the example above, the user has to change between paper artifact and mobile device in order to

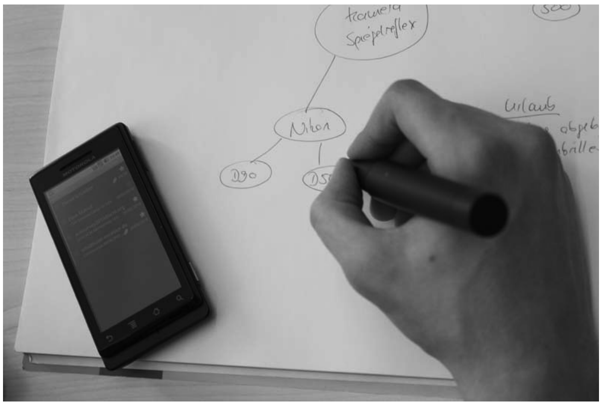


Figure 1.1: A *hybrid mPPI ensemble* consisting of pen, paper and mobile device

distribute the design sketch. However, relevant information (the sketch) remains exclusively on the paper artifact: the user activity is hindered by a gap between the digital and physical worlds, between mobile devices and paper documents.

Despite sophisticated technology, this gap remains a problem. Numerous predictions of a decreasing role of paper have been proven false during the last decades, e.g., as reported by Sellen and Harper in their famous work on *The Myth of the Paper-less Office* [Sellen and Harper, 2003]. More recent studies further corroborate this, showing that despite the broad availability of digital devices, pen and paper remain predominant tools due to their unique affordances and physical form factor, e.g., in the context of education [Malacria et al., 2011] and knowledge work [Chapman et al., 2009].

Thus, in order to bridge the gap between the digital and physical worlds, a growing body of research focuses on *connecting* paper and the digital world through digital pen technology, rather than *replacing* paper by digital devices, e.g., [Liao et al., 2010b, Steimle, 2009a, Yeh et al., 2006a]. This approach leads to a new style of human computer interaction: *Pen-and-Paper Interaction* (PPI).

In this context, addressing the mobile domain becomes increasingly relevant as pen and paper provide effective and flexible support for mobile information management, e.g., [Cowan et al., 2011, Tsandilas, 2012, Pietrzak et al., 2012]. Applying PPI in mobile settings connects the most common mobile information processing tools, i.e., mobile devices and paper. This leads to the following important definitions:

Pen-and-Paper Interaction (PPI) This term refers to a form of human-computer interaction, where the *user* interacts with a *digital system* by means of a *digital pen*[1] and *paper*.

[1] Please refer to chapter 2 for a definition and detailed introduction into the underlying technologies.

Pen-and-Paper User Interface (PPUI) This term refers to a user interface employing PPI in addition to, or instead of, a *graphical user interface* (GUI). The term PPUI was originally coined by Steimle to denote a tangible user interface with a digital system employing PPI as its main modality, [Steimle, 2009a], e.g., by offering printed interactors on paper allowing to access digital functionality (c.f., section 2.4.1).

mobile Pen-and-Paper Interaction (mPPI) This term refers to interaction between a *user* and a *digital system* by means of *digital pens*, *paper* and *mobile devices* in *mobile* or *nomadic*[2] settings.

Hybrid mPPI Ensemble This term refers to a combination of *digital pens*, *paper* and *mobile devices* (c.f., Fig. 1.1). Based on the definition of *mobile device ensemble* by Schilit and Singupta [Schilit and Sengupta, 2004], hybrid mPPI ensembles strive to form a cohesive whole supporting workflows spanning its components by means of mPPI. Thereby, *hybrid* emphasizes the challenge of bridging the digital and physical worlds within a cohesive ensemble.

Supporting hybrid mPPI ensembles proves challenging as of today. Developers are hindered by the lack of *infrastructure* for developing integrated, mobile systems preserving the original, mobile characteristics of pen and paper. At the same time, initial approaches explore promising mPPI interaction techniques for hybrid mPPI ensembles in experimental settings. However, there is no deeper theoretical understanding of mPPI, i.e., *interaction* between users and digital systems by means of digital pen and paper in the mobile domain.

In this thesis, a novel infrastructural approach toward supporting mPPI is described and a corresponding reference implementation is introduced. In addition, a conceptual framework for describing PPI is developed and it is shown how this framework can serve as a foundation for toolkits supporting interaction designers in developing compelling PPUIs for hybrid mPPI ensembles. Finally, an empirically substantiated theory of interaction in hybrid mPPI ensembles is described and a set of concrete guidelines deepening the understanding of interaction design in the domain is derived from this theory. Thereby, this thesis is structured based on the dichotomy of *infrastructure* and *interaction* related challenges, with the conceptual framework as connecting element.

The remainder of this chapter is structured as follows. Section 1.1 establishes why hybrid mPPI ensembles are worthwhile studying and why the ensemble consisting of smartphone, pen and paper provides a relevant representative of hybrid mPPI ensembles. It presents results of an initial observational study, as well as an in-depth case study of an application of hybrid mPPI ensembles and its underlying design rationale:

[2] in this context referring to temporarily stationary settings at arbitrary locations

the *Digital Grocery List*. Subsequently, section 1.2 outlines the research questions addressed and contributions made in respectively by this thesis in detail. Additionally, it describes the employed research methodology and enumerates previous publications containing material presented in this thesis. Finally, section 1.3 describes the structure of this thesis and lays out the contents of chapters to come.

1.1 Hybrid mPPI Ensemble Research

This section reviews the motivation to combine paper and mobile devices in hybrid mPPI ensembles. Several studies underline the continued importance of paper as an information processing tool, e.g., [Sellen and Harper, 2003], [Steimle et al., 2008c] or [Harboe and Huang, 2015]. However, a re-examination of their results in the context of mobile use and recent technological advances is required. Toward this end, this section reports on the results of an ethnographic study underpinning the continued importance of pen and paper in mobile settings at the same time yielding hybrid mPPI ensembles as a promising setting for further study. Following this, it introduces a detailed case study of a mobile application involving paper artifacts and the need for interactive data access: the *digital grocery list* application. Results are used to set the stage for the research questions addressed in this thesis as outlined in section 1.2.

1.1.1 Mobile Devices vs. Paper

Mobile practices have changed rapidly throughout the past decade. Increasingly affordable and powerful mobile devices of various form factors offer (almost) anytime, anywhere connectivity and interactive access to dynamic content as envisioned by the late Mark Weiser [Weiser, 1991]. Contemporary mobile devices offer engaging and compelling interactive functionality and support users in a broad range of contexts. Thereby, a *mobile device* in the context of this thesis is characterized by a form factor that allows for *mobile use* for all practical intents and purposes, an operation scheme *independent* of a continuous power source[3], *connectivity* to the Internet or other backing networks (e.g., cellular networks), and *interactive* capabilities.

Today, mobile devices offer a broad range of functionality and varying form factors. Mobile devices range from Notebook PCs, over the smaller and more portable Sub-Notebook and Netbook classes, to tablets and smartphones. This list is by no means exhaustive as mobile devices continue to evolve and smaller, more light-weight units have been introduced, e.g., the *Google Glasses*[4] or the *Samsung Smartwatch*[5].

[3] i.e., a device that is able to operate on battery

[4] https://developers.google.com/glass/ (accessed: July 2015)

[5] http://www.samsung.com/us/explore/gear-s-features-and-specs/ (accessed: July 2015)

However, mobile devices still suffer from several drawbacks. These include problematic form factors, short battery cycles and handling issues, such as inconvenient content entry or the so called *up-and-running time* (the delay between grabbing the device and being able to actually use a desired application). Such drawbacks considerably reduce the actual usefulness of mobile devices.

Another set of problems relates to the Computer-Human Interface of mobile devices. For instance, they require too much attention in settings where users can devote only a fraction of their attention to the device, e.g., while driving or in a meeting[6].

Complex trade-offs further complicate the situation. While limited screen real-estate reduces the amount of information that can be digested and interaction that can be supported, smaller form factors and more portable devices typically require limiting the available screen real-estate.

The present thesis pays particular attention to such Human Interaction issues. It follows the paradigm of additional modalities that complement the prevalent touch screen interaction. Moreover, instead of adding voice *Input/Output* (I/O), or *Tangible User Interfaces* (UI) [Ishii and Ullmer, 1997], it uses the affordances of pen and paper.

Pen and paper offer many of the missing aspects of mobile devices. Essentially, pen and paper are inherently mobile tools not affected by annoying startup times, screen unlock procedures, contrast problems or short lived batteries. Therefore, these tools remain prevalent media in many domains of our daily lives, ranging from tasks such as knowledge work to seemingly trivial tasks as organizing shopping trips. Particularly in informal and mobile settings, such as note-taking, pen and paper remain the prevalent tools for capturing and storing information [Brandl et al., 2010].

This is due to their inherent flexibility, robustness and the instant-on nature paired with a form factor allowing to effortlessly transport them in almost every situation [Yeh et al., 2006a]. Paper being ubiquitous and at close to zero cost, there are virtually no constraints with respect to real-estate available for users.

On the other hand, pen and paper are strictly limited as they do not offer any interactive functionality, lack communication abilities and to not allow to access to digital information. Here, hybrid mPPI ensembles strive to combine the advantages of both worlds.

Ethnographic Study

An initial ethnographic study was conducted in order to investigate contemporary use of pen and paper in mobile settings. It aimed to compare the use of mobile devices and pen and paper in the domain of knowledge work by observing strategies and practices employed by users in mobile settings (without intervention). Thereby, a typical case

[6] although arguably some users tend to devote their attention to the mobile device nevertheless

of mobile settings are meetings, as these (in most cases) do not take place at the users workplace, but rather in dedicated meeting rooms or in an ad-hoc manner, e.g., in the coffee kitchen. The observed meetings were of varying nature with respect to the factor mobility and pre-planning. They can be classified into the following three groups

formal meetings follow a formal process or protocol (e.g. regular meetings in a project where each participant reports on progress) and are arranged in advance.

planned meetings have been planned and arranged in advance, yet do not follow a prescribed structure.

casual meetings have a spontaneous character and occur without a planing process. They do not follow any predefined structure

Over a study period of 40 days 13 meetings taking place were covertly observed. For each meeting, it was recorded which *devices* the participants used during the meeting. Such recording took place for three types of devices and their usage frequencies were compared: *pen and paper*, *mobile devices* including smartphones and tablets, and *laptop PCs*. This turned out to be exhaustive for the observed meetings, i.e., there were no other classes of devices in use. In addition to device use, combinations of device use during a meeting were observed, e.g., whether a participant used laptop and mobile device in combination. Besides quantitative data on device use, qualitative data was gathered and participants showing interesting behavior were subsequently engaged in a short interview.

Results. In total $n = 148$ subjects were observed in the 13 meetings (3 formal, 8 planned, and 2 casual meetings). Most of the meeting participants had a strong background in computer science yielding a considerably more technology affine sample than average society. Still, 57% of the observed subjects used pen and paper during meetings. On the other hand, as expected, the subjects did also use mobile devices and laptops in order to access digital information. Fig. 1.2 shows the frequencies of device use in relation to number of participants per meeting. Here, a *Kruskal-Wallis* test suggests a significant difference in the usage rate between pen and paper and other devices ($p \leq 0.05$): Pen and paper were significantly more often employed than other devices in the context of the study. With respect to combinations, results show a similar situation pointing toward the combination of pen and paper with mobile devices: users preferred this over any other combination of tools ($p \leq 0.05$, *Kruskal-Wallis*). The use of combinations of devices in relation to participants per meeting is shown in Fig. 1.3.

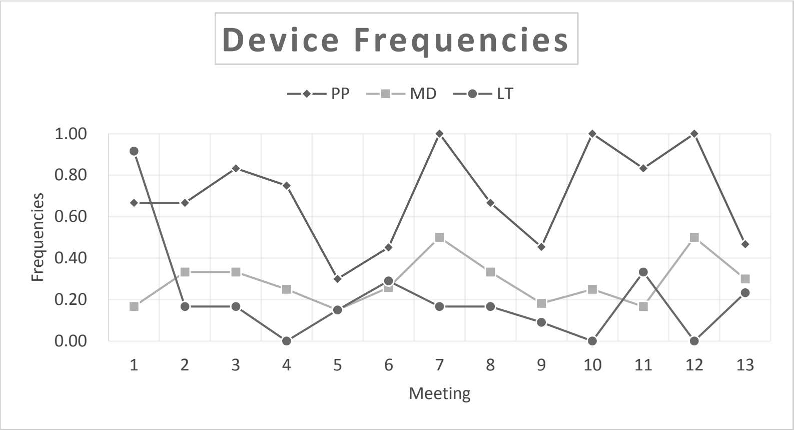


Figure 1.2: Frequency of device use in meeting scenarios (PP: Pen and Paper, MD: Mobile Device, LT: Laptop)

Qualitative data showed that paper documents were used to *access* and *generate* information in written form. Laptops were used for similar purposes. In contrast to this, mobile phones were almost exclusively used to access information, e.g., personal calendars and emails. If device combinations were used, the participants repeatedly switched back and forth between devices. Accessed information on paper was either handwritten, i.e., no digital form of the data existed, or printed. Examples for handwritten information were agenda points to discuss in a meeting, conceptual drawings or sketches of development ideas. Printed information existed, e.g., in the form of paper printouts of digital documents. Several times we observed that handwritten and printed information was combined, e.g., paper printouts where augmented with handwritten annotations. Different types of paper documents were used by meeting participants to generate information. We observed three categories of paper documents which we will denote in the remainder as follows

dedicated paper This refers to paper documents dedicated to serve the purpose of documenting handwritten information, typically kept for more than one occasion (e.g., personal notebooks).

environment paper This characterizes paper documents that existed in the environment and were used to document information although not being specifically designed for this purpose; typically only kept for one specific occasion (e.g., blank sheets of paper).

occupied paper This denotes paper documents that were actually designed to serve another purpose than documenting handwritten information (e.g. printouts of

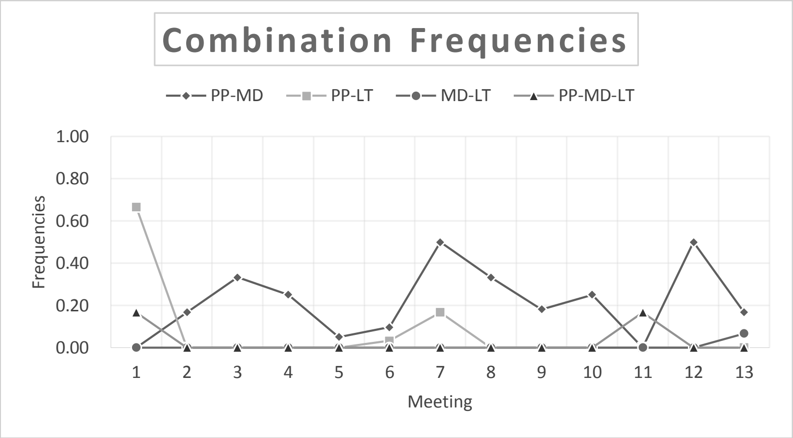


Figure 1.3: Frequency of device combinations in meeting scenarios (PP: Pen and Paper, MD: Mobile Device, LT: Laptop)

digital documents) that were ”borrowed”; typical examples are the unused back side of printed documents, or the remaining white space at the sides.

An interesting observation with respect to mobility was that participants employed not only *brought along* paper artifacts, e.g., dedicated paper in the form of notebooks, but also *encountered* paper artifacts, e.g., environment paper or occupied paper available in the meeting room. Furthermore, exchange of paper artifacts took place, e.g., participants handed paper printouts to other participants. Subsequently, these encountered artifacts were used by the recipient to document and process information, both as occupied paper (information unrelated to content) and in the form of annotations (information related to content).

Discussion. Results of the ethnographic study show that pen and paper continue to play an essential role in the mobile domain. Thus, these results confirm earlier studies for stationary settings [Sellen and Harper, 2003, Malacria et al., 2011] and reestablish their claims in spite of recent technological advances: even contemporary smartphones have not yet succeeded in replacing pen and paper as mobile information processing tools. On the contrary, the numerous advantages of paper in the mobile domain prompt users to rely on paper artifacts despite the availability of powerful and versatile mobile devices.

However, where interactive functionality is required, paper artifacts are complemented rather by mobile devices than laptops indicating that this combination is important for nomadic and mobile scenarios. Mobile devices offer access to dynamic information, e.g., calendars or web pages, allow communicating with others and provide the link to the digital world yet are easier to carry around, even compared to devices of the Netbook class. Also, the use of mobile devices seems to be more socially acceptable and possible with less dedicated attention.

Thus, a combination of pen and paper and mobile computing devices seems to be an ideal solution. Factually, 26% of the participants used pen, paper and mobile device already in parallel despite lack of integration between these tools. However, further integration is required in order to leverage full support without experiencing a disruptive transition between pen, paper and the mobile device, the *digital-physical gap*. In order to do this with respect to knowledge work, users should be able to interactively work with, communicate and process documented information in a mobile setting. Furthermore, interaction with *brought along* and*encountered* paper artifacts of all three classes should be supported. Providing such integration is the central goal of this thesis.

1.1.2 Case Study: Digital Grocery Shopping Support

The first case study reported above revealed the potential of improving the integration of pen, paper and mobile devices. However, it did so in the quite specific domain of knowledge work that is marked by pervasive use of information technology and technology affine users. In a second study we aimed at a deeper understanding of the potential of such integration for everyday (mobile) work and arbitrary human users.

Toward this end, an initial in-depth case study of an application scenario was conducted. Aim of this case study was to derive requirements for a real-world system supporting users with common tasks in the mobile domain. Additionally, the case study allowed identifying and establishing challenges with respect to interaction design of such a system. Furthermore, this case study serves as the prevalent running example of the concepts introduced in this thesis.

Grocery Shopping. The selected mobile activity was the common everyday task of *grocery shopping*. Paper plays an important role in planning grocery shopping in the form of handwritten shopping lists and paper leaflets, e.g., as shown in Fig. 3.14. Using paper here is extremely common and intuitive: One may say that handwritten grocery lists and quick glances at paper leaflets with special offers distributed by retailers are common in almost every household.

Additionally, a broad variety of mobile applications supports this task. This can be illustrated by reviewing figures from the applications stores for mobile devices. A search for ”shopping list” delivers more than two hundred different applications for

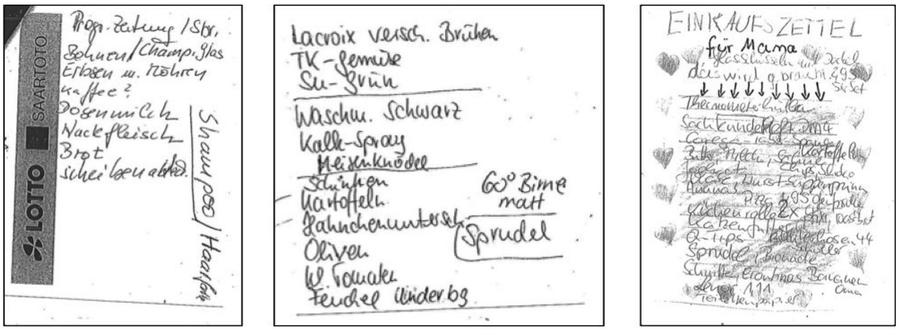


Figure 1.4: Paper-based shopping lists: a) A list mainly containing only product categories. b) A list containing product instances grouped into categories. c) A list containing promotion articles designated for a special person (in this case “mama”).

creating and managing shopping lists for the iOS platform in the Apple App Store[7], while there are more than two thousand different applications available for the Google Android platform[8]. These applications aim at supporting the process of grocery shopping through additional digital functionality, e.g., reminders, integration with online resources and live list updates. This illustrates the potential of mobile device support for grocery shopping.

Case Study. Based on these findings, the *Digital Grocery List* (DGL) case study applies the concept of a hybrid mPPI ensemble, i.e., a much more tightly integrated combination of pen, paper and mobile device through digital pen technology as discussed above, to the domain of grocery shopping. The following three distinct steps were employed in order to elicit requirements for such an application and corresponding system support, as well as highlight its particular challenges

1. analysis of practices related grocery shopping in the literature
2. analysis of existing applications for grocery shopping support
3. a field-study in a big German retail market in order to gain deeper insight in user practices and the role of paper artifacts

Subsequently, obtained data was analyzed and a a set of design implications for a DGL application was derived employing mPPI as enabling factor of a tight integration

[7] https://ssl.apple.com/search/?q=shopping%20list&section=ipoditunes (accessed: July 2015)

[8] https://play.google.com/store/search?q=shopping+list&c=apps (accessed: July 2015)

between application components. This yielded the design of a hybrid mPPI ensemble based application supporting the everyday task of grocery shopping. In particular, the impact of design guidelines on required infrastructures and questions related to interaction design was analyzed.

In Depth Analysis: The Grocery Shopping Process

The everyday task of grocery shopping has been extensively studied in the consumer and retail research community, e.g., by Puccinelli et al. [Puccinelli et al., 2009]. In most cases, the planning phase for grocery shopping is quite extensive consisting of multiple planning cycles [Thomas and Garland, 2004, Block and Morwitz, 1999]. It can be distinguished from the actual purchase phase. Thereby, Block and Morwitz refer to these distinct phases as the *List Writing Stage* and the *List Fulfillment Stage* [Block and Morwitz, 1999]. Most households make use of grocery lists in the process of shopping planning, although sometimes the list is only mentally maintained [Bassett et al., 2008].

Grocery lists are typically prepared either collaboratively, or by a person responsible for the need management of the household [Bassett et al., 2008]. Thereby, written grocery lists often serve as an external memory aid to facilitate the shopping process [Block and Morwitz, 1999]. In addition, they are used as a way to communicate needs to other household members [Bassett et al., 2008]: The grocery list is passed as planning document to the person doing the actual shopping [Block and Morwitz, 1999]. Block and Morwitz found that there are not only need based, but also financial incentives for assigning groceries to the list, i.e., coupons or bargain offers found in leaflets [Block and Morwitz, 1999]. This means that these paper documents are also included in the shopping planning process.

Existing System Support for Grocery Shopping

Toward digital system support for grocery shopping, Shekar et al. suggested a *personal digital assistant* (PDA) based application to provide ubiquitous access to a digitally managed grocery list [Shekar et al., 2003]. Although the proposed application provided several means to add items to the shopping list, e.g., a barcode scanner, it did not support paper lists as typically used in the shopping planning process. Another PDA based approach was suggested by Newcomb et al. [Newcomb et al., 2003]. Its design was based on an extensive ethnographic study. Their findings highlighted the importance of ubiquitous access to additional data, such as dynamic information from the web, during the shopping planning process.

Nurmi at al. [Nurmi et al., 2008], [Nurmi et al., 2009] introduced a grocery retrieval system mapping shopping lists written in natural language to products in the store's portfolio. The system was developed based on nine months of shopping basket data from a large Finnish supermarket. An application for shopping list creation using multiple input devices was introduced by Jain et al. [Jain et al., 2008]. It encompasses entry on desktop PCs, smartphones, land line or cell phones and supports multimodal formats, e.g., structured text, audio, still images, video, unstructured text and annotated media. Wu et al. presented a mobile shopping assistant to demonstrate a novel architecture enabling efficient integration of mobile applications and Web Services [Wu and Natchetoi, 2007].

Interestingly, none of the existing mobile applications has considered pen and paper as input modality, although *Pen-and-Paper User Interfaces* (PPUI) have been used by others, e.g., in *ButterflyNet*, [Yeh et al., 2006a], and *NICEBook*, [Brandl et al., 2010] (see chapter 2 for a detailed overview). In order to overcome this limitation, Liwicki et al. developed a shopping list application employing PPI [Liwicki et al., 2011] in parallel to first publication of the results described throughout this section[9]. It uses a sophisticated handwriting recognition approach coupled with a product ontology to predict users purchases. However, in contrast to the design described here, it targets a stationary setting only and neglects mobile aspects of the domain.

Field Study

A field study was conducted as a third step in order to understand creation and usage of paper-based shopping lists and complement the findings reported above. In this study, two experimenters examined the shopping behavior of customers in the *Globus Markt*, a big German retail store, during 2 weeks in March 2010 at different times of the day (morning, noon, afternoon, in the evening shortly before the shop was closed). Thereby, the customers were asked if they want to take part in our study, directly after they had paid their goods at the cashiers. 270 customers agreed to participate in the study.

Experimenters elicited demographic data of participants, including age, gender, income, shopping experiences and shopping frequency. The overall background of the study sample is as follows: $2/3$ of the participants were women, while $1/3$ were men. More than 90% of participants were familiar with the market since more than 3 years. The mean age of all participants was 38.2 years. The monthly incomes were around $1800 - 2000$ EUR. 50% of the participants visited the store $1 - 2$ times a week. 20% of the participants visited the store $3 - 4$ times a week and another 14% just a couple of times a month.

47% of the participants relied on shopping lists in their preceding shopping, however, only 3% used electronic shopping lists, e.g., smartphone applications. All re-

[9] published in [Heinrichs et al., 2010a] and [Heinrichs et al., 2011b] respectively

maining shopping lists were paper based lists. This is a strong indication that the majority of people still relies on paper-based shopping lists.

Three examples of different shopping lists are presented in figure 1.4. The shopping lists contained 13.3 (mean) items (Median 11, Min. 2, Max. 47). The lists often contained very little detail: just 10% of the lists showed an amount or unit label for the product. Often the participants used generic terms on their lists (about 2/3 of all items on a list). Instead of specifying products by their proper name, they used generic terms such as *"beer"*, *"fruits"* or *"some sweets for the kids"*.

In contrast to this, people often had specific items from promotions on their lists. In many instances, these items contained a lot of detail, e.g., location of the market, price, discount rate etc.). In addition, the lists often contained *pointers* to family members such as: *"marmalade for Eva"*, *"food for Mietze[10]"*. In some cases, different scripts indicated that different people had collaborated in writing the list together.

Design Guidelines

In the following, the results reported above will be used to derive a set of design guidelines for an mPPI based shopping application, the *Digital Grocery List* (DGL). These guidelines lay the foundation for formulating detailed general questions for infrastructure and interaction research in the domain of hybrid mPPI ensembles. As such, they outline a running example used to illustrate the concepts introduced in this thesis.

Collaborative Creation and Editing. Findings in retail research point toward a collaborative list creation process for shopping list. This is corroborated by different scripts used to create the shopping lists collected in our field study. Hence it can be derived that shopping planing and creation of shopping lists is often done collaboratively, i.e., involving multiple members of a household. Furthermore, if creation is executed collaboratively, the same should be supported for subsequent editing in order to maintain consistency. Thus, applications supporting the grocery shopping task should be designed to support *collaborative creation and editing* of shopping lists.

Access to Inform. The shopping list serves as communication medium in the context of shopping. As shopper and planner are not necessarily the same person, instructions what to buy have to be communicated to the shopper, i.e., the household member actually performing the shopping trip. Ideally, this should be achieved in (near) real-time in order to minimize errors, e.g., buying the wrong items, or not

[10] colloquial German for "cat"

buying required items. *Accessing* the list to inform the shopping process *in the store* should be supported by an application.

Household Vocabulary. The study also revealed that users add additional information to items (e.g., *"[...] for Eva"* as shown in Fig. 1.4). We believe that these hints are highly personal markers for product details, e.g., *"marmalade for Eva"* means marmalade from a specific brand that Eva likes. The vocabulary used for specifying items on the list is also highly heterogeneous: users tend to use acronyms, colloquial expressions and textual clarifications, as can be seen in Fig. 1.4. Therefore the user should be able to specify shopping list items with arbitrary, individually chosen names. Thus, the system should support a *household vocabulary* for list items.

Hybrid mPPI based Design & Handwritten Creation. With respect to the household vocabulary described above pen and paper input provides a clear benefit: keying in such item names is very tedious on mobile devices, while it is easily done with pen and paper. At the same time, the continued use of pen and paper suggests that it is a very natural and convenient tool to create shopping lists in general. Therefore, the application supporting the grocery shopping task should employ a *hybrid mPPI* design, i.e., combine a mobile GUI and a PPUI (c.f., hybrid mPPI ensemble).

In addition, only 5% of the collected shopping lists bear marks of active editing in the market, e.g., check marks or crossed out items. Thus it can be assumed that handwritten lists primarily serve the planning phase: not the handwritten list itself, but rather its content, is used during the actual shopping (c.f. access in the store described above). So a hybrid mPPI shopping list application should allow *handwritten creation* of lists, yet the usage of this handwritten list in the market can be neglected in regard to other design considerations.

Arbitrary Paper Artifacts. Regarding the handwritten creation of shopping lists, it can be observed that people use heterogeneous types of paper artifacts as writing media, ranging from notepads to old paper envelopes (c.f., Fig. 1.4). This is evident from the sample of handwritten lists collected, as well as anecdotal evidence. Users tend to re-use any available paper real estate, similar to the occupied and environment paper classes described in the ethnographic study reported in section 1.1.1 As a result, hybrid shopping list applications should support the usage of *arbitrary paper artifacts*.

Additional Resources and Information. As reported by Block and Morwitz, additional paper artifacts play a role in the planning phase, e.g., leaflets, special offers and coupons [Block and Morwitz, 1999]. These artifacts are distributed by the supermarket and thus encountered by users in the shopping planning process. An application aiming to support the planning process therefore needs to incorporate *additional resources*, especially in the form of linked paper artifacts such as leaflets. Furthermore, people often require additional information regarding particular items on their list, e.g., for price comparison. So a shopping support application should integrate *additional information* about list items as well and link this information to items in the list. This extends to the actual list fulfillment stage, as users might require this information in the store.

In summary, the design guidelines are the following:

- support collaborative creation and editing
- allow access in the store to inform the shopping process (manipulation in the store is less relevant)
- provide a household vocabulary
- employ hybrid mPPI based design and allow for handwritten creation
- support arbitrary paper artifacts
- provide access and links to additional resources and information

Thereby, interaction in the mobile domain as well as the integration of different paper artifacts has to be supported. As in the ethnographic study reported in section 1.1.1, paper artifacts are both *brought along*. e.g., the grocery list, and *encountered*, e.g., the leaflets constituting additional resources and information. Furthermore, paper artifacts might be used in varying contexts as indicated by the use of arbitrary paper artifacts.

1.1.3 Characteristics of Hybrid mPPI Ensembles

The studies in sections 1.1.1 and 1.1.2 show that pen and paper play an important role in mobile usage practices despite the availability of powerful interactive mobile devices. This situation is unlikely to change, as paper itself affords mobile use naturally and intuitively while it offers unique advantages with respect to spatial information management, sorting and storing information without constraining the user in terms of contents [Sellen and Harper, 2003]. However, mobile devices offer interactivity, access to dynamic content and communication facilities. Both aspects are needed by users as demonstrated above.

Using pen, paper and mobile devices side by side is possible and common as shown in section 1.1.1. However, when interacting with paper and digital devices side by side, the user experiences a disruptive transition, the *digital-physical gap*

[Steimle, 2009b]. In order to overcome the digital-physical gap and integrate workflows in the mobile domain, pen, paper and mobile device should be combined into a cohesive whole using digital pen technology. This yields the notion of a *hybrid mPPI ensemble*, as defined above. Here, *pen-and-paper interaction* (PPI) is leveraged to the mobile domain, thus enabling *mobile PPI* (mPPI): interaction with a mobile device using digital pens and paper (c.f., definitions given on page 2 ff.).

Typical applications benefiting from this style of interaction are support for mobile everyday tasks, e.g., the grocery shopping process outlined in section 1.1.2, and mobile knowledge or creative work, e.g., mobile note-taking as in [Yeh et al., 2006a] or [Pietrzak et al., 2012]

Mobility Schemes. Hybrid mPPI ensembles aim at combining pen and paper with mobile devices in order to support users in mobile contexts. Thereby the mobile context refers to both *mobile* and *nomadic*, i.e., temporarily stationary, use. Examples for mobile use are interaction in the subway, during a meeting or while walking. Examples for nomadic use are seated in a colleague's office, at the workplace or in a hotel. Thereby boundaries are not fixed entirely. Beyond these mobile contexts, the studies in section 1.1.1 and section 1.1.2 respectively highlight two distinct *forms of mobility* with respect to paper artifacts that transcend hybrid mPPI ensembles

User Mobility Here the user interacts with brought along paper artifacts, e.g., the dedicated paper class described in section 1.1.1. Thereby, the paper artifact is not mobile with respect to the user, however, the user interacts with it in a mobile or nomadic setting. An example here is a user interacting with a brought-along notebook or the paper version of a grocery list brought to a store.

Document Mobility Here the user interacts with encountered paper artifacts, e.g., the environment paper class, or the paper artifact is used in multiple contexts, e.g., the occupied paper class described in section 1.1.1. Thereby the paper artifact itself is mobile, either with respect to the user, or to its usage context. An example here is the leaflet distributed by the supermarket that the user encounters in the grocery shopping process (c.f., 1.1.2) or the paper printout passed from one colleague to another in the context of knowledge work.

Smartphones as Representative. Section 1.1.1 showed that pen, paper and mobile device are the most often observed combination of mobile tools supporting knowledge work. As such, this combination yields a promising hybrid mPPI ensemble for further studies. Thereby, the smartphone is used as representative for mobile devices with Internet access. Recent projections estimate that in 2014 approximately

1.75 billion people possess a device of the smartphone class[11]. With a world population of 7.2 billion, this corresponds to almost one quarter of the world population, while for American adults this figure is even higher (60%[12]).

Thus, it is safe to assume that smartphones are a widespread form of mobile device. Smartphones are characterized by their ability to host software beyond the mere use of the device as a phone, e.g., note-taking software or games, by additional sensors, e.g., built-in cameras and accelerometers, and interaction paradigms revolving around multi-touch interaction. For the scope of this thesis, smartphones will serve as the main representative of mobile devices unless stated otherwise. Contributions, however, generalize beyond this class alone.

1.2 Research Questions and Contributions

This thesis advances integration of hybrid mPPI ensembles and the design of interaction spanning the most common mobile tools, i.e., pen, paper and mobile device. Thereby, the goal is to leverage the complementary benefits of each ensemble component to shape a whole that is greater than the sum of its parts.

In particular, this thesis focuses on limiting the adverse effects that *infrastructure* design has on user interaction. This forms a challenging and important problem: poor infrastructure design can hinder interaction design by limiting available and beneficial design choices and usage practices [Edwards et al., 2010]. At the same time, this thesis aims to support compelling and engaging user *interaction* through adequate infrastructural support and understanding of important interaction concepts with respect to hybrid mPPI ensembles.

As a result, contributions are situated between the fields of *human computer interaction*, *mobile and ubiquitous computing* and *software engineering*, focusing in particular on infrastructure and interaction design. In this dichotomy between infrastructure and interaction design, conceptual frameworks of interaction form the connecting element, as they define *what* the infrastructure needs to support and *how* interaction can be modeled. As such these conceptual frameworks provide appropriate abstractions of interaction that can be directly supported by the underlying infrastructure, i.e., in the form of interaction toolkits.

These three central aspects also determine the main contributions and structure of this thesis (c.f., section 1.3). Fig. 1.5 depicts research questions addressed in this thesis, relationships between contributions and related fields of research. The red, dashed box marks research questions and contributions addressed by and presented in

[11] http://www.emarketer.com/Article/Smartphone-Users-Worldwide-Will-Total-175-Billion-2014/1010536 (accessed: July 2015)

[12] http://www.pewinternet.org/fact-sheets/mobile-technology-fact-sheet/ (accessed: July 2015)

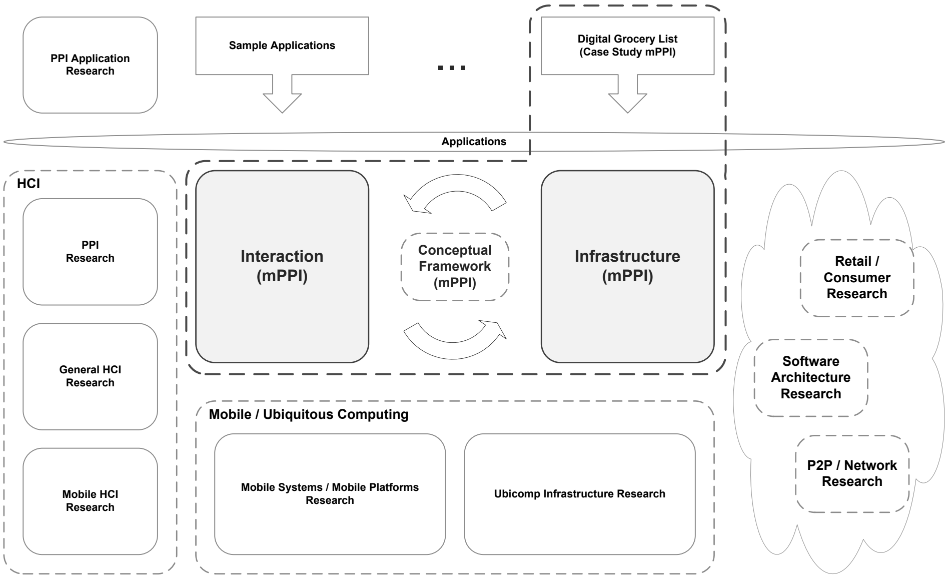

PPI Application Research
Sample Applications
...
Digital Grocery List (Case Study mPPI)
Applications
HCI
PPI Research
General HCI Research
Mobile HCI Research
Interaction (mPPI)
Conceptual Framework (mPPI)
Infrastructure (mPPI)
Mobile / Ubiquitous Computing
Mobile Systems / Mobile Platforms Research
Ubicomp Infrastructure Research
Retail / Consumer Research
Software Architecture Research
P2P / Network Research

Figure 1.5: Infrastructure vs. Interaction Design in Hybrid mPPI Ensembles

this thesis respectively. *Interaction* and *infrastructure* research of support for mPPI forms the two central aspects. Thereby, research of *conceptual frameworks* describing mPPI connects these aspects. Research on the *Digital Grocery List* (DGL) (c.f., section 1.1.2) provides a case study exemplifying these concepts.

Related topics offer additional insights into these fields and serve as basis for further research, e.g., existing infrastructures for PPI serve as foundation for research on mPPI infrastructure. These topics are marked by a gray, dashed box containing boxes of particularly related sub-fields. Loosely related fields are shown in the cloud on the right. Although not directly related to the contributions, certain insights, techniques or approaches applied in this thesis stem from these fields.

Infrastructure. Research questions addressed by this thesis with respect to *infrastructure* are: What infrastructure supports integrating and combining mobile devices, pen and paper while preserving their mobile characteristics? What architecture is required to support mPPI? Thereby, the infrastructure has to support hybrid mPPI ensembles in mobile or nomadic settings, while supporting the mobile usage schemes of pen, paper and mobile device, in particular *user mobility* and *document mobility* (as laid out in section 1.1.3). Current infrastructures thereby fail to address these issues as demonstrated in chapter 2. As a result, research and development of applications for hybrid mPPI ensembles is severely hindered. Contemporary approaches restrict themselves to investigating hybrid mPPI ensembles through experimental setups relying heavily on non-mobile components, e.g., stationary servers for processing (some parts of) user interaction as in [Pietrzak et al., 2012] and [Tsandilas, 2012].

Contribution. This thesis advances the state of the art with respect to infrastructural support for mobile PPI by demonstrating why current infrastructures based on the *monolithic PPI processing pipeline* fail to support mobile PPI on an architectural level. It contributes a novel architecture for infrastructures supporting mobile PPI remedying these limitations: the *distributed interaction processing pipeline*. In addition to that, it contributes *Letras*, an architecture and reference implementation of infrastructure basing on this novel architecture, demonstrating practical relevance of these findings. Thereby, the novel architecture enables system support for mobile PPI preserving usage characteristics of real pen and paper.

Conceptual Framework. The conceptual framework allows describing interaction and therefore forms the connecting element between interaction and infrastructure design. However, its contributions are more closely related to the field of interaction. Research questions addressed in this context are: How can pen-and-paper interaction be described in a way that enables the infrastructure to interpret user actions? How can the design space of PPI in general be described? And, with respect to connecting infrastructure and interaction: How can the infrastructure provide appropriate abstractions to an interaction designer, that allow fast and convenient development of interaction techniques for hybrid mPPI ensembles?

Contribution. Toward this end, this thesis contributes an analytically derived conceptual framework for PPI allowing to describe and categorize interaction techniques involving pen and paper in general and in the mobile domain in particular. The conceptual framework maps to logic programming, thus, interaction techniques can be interpreted by digital systems. This allows building infrastructural components aimed at interaction design support, e.g, toolkits. A reference implementation of such a toolkit demonstrating the practical relevance of theoretical findings is also contributed. Thereby, the conceptual framework connects interaction design and infrastructural support in hybrid mPPI ensembles.

Interaction. Research questions addressed by this thesis with respect to *interaction* are: How should interaction spanning mobile devices, pen and paper in mobile settings be designed, i.e., what does an interaction designer need to consider with respect to hybrid mPPI ensembles? Which interaction techniques should be chosen when?

What are the key characteristics of interaction in hybrid mPPI ensembles? What are common problems and pitfalls preventing engaging and compelling interaction spanning multiple components of these ensembles? Thereby interaction design needs to take component heterogeneity into account as highly interactive mobile devices and more or less static paper artifacts need to be combined into a cohesive whole.

Contribution. This thesis advances the state of the art with respect to interaction design for mobile PPI by presenting a novel, empirically substantiated theory of interaction in hybrid mPPI ensembles. This theory provides the first structured approach to interaction design in the domain and allows designers to obtain a deeper understanding of phenomena and their respective relations. Additionally, a set of concrete design guidelines derived from the theory are reported demonstrating how interaction designers can benefit from improved understanding gained through the theory. Thereby, this theory enables improved interaction design with respect to interaction techniques spanning pen, paper and mobile devices.

1.2.1 Methodology

The research questions formulated in section 1.2 with respect to infrastructure and interaction in hybrid mPPI ensembles are characterized by an *enabling* and *explorative* nature. On the one hand, leveraging combinations of pen, paper and mobile device through digital pen technology to its full extent became possible only recently due to advances in mobile platforms and technology. The advent of smartphones in the 2000's, e.g., the introduction of the *iPhone* in 2007 and the first *Android* phones in 2008, provided mobile platforms with sufficient computational power, communication facilities and interaction capabilities to enable mPPI. As a result, existing approaches are still limited. On the other hand, research in human computer interaction in general aims to explore strategies to optimize interaction between human users and interactive systems.

As a result, an explorative research approach was chosen. This approach combines a broad spectrum of methods ranging from theoretic domain analysis and proof- of-concept system design [Olsen, 2007] to qualitative, empiric research allowing to generate empirically substantiated theory, i.e., a research approach inspired by grounded theory [Corbin and Strauss, 1990]. Thereby, the research methods employed in this thesis follow the dichotomy of infrastructure and interaction research as shown in Fig. 1.6.

Infrastructure. Contributions with respect to infrastructure aim to *enable* mPPI, that is, no reference systems for the mobile domain exist. As such, the employed research method follows Olsen's approach to evaluate novel user interface systems with

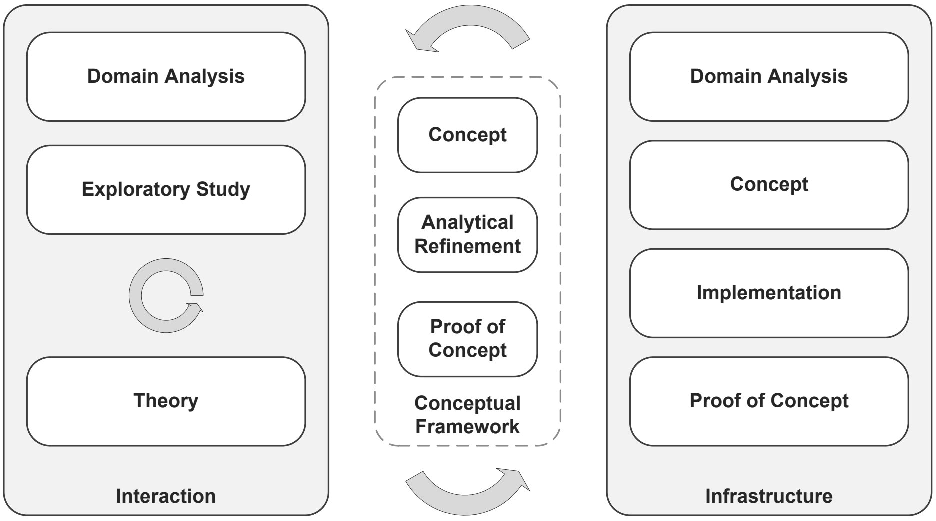


Figure 1.6: Employed Research Method with respect to Infrastructure, Interaction and Conceptual Frameworks of Interaction

respect to common infrastructure [Olsen, 2007]. Here, an analytic method combined with proof- of-concept evaluation was employed as opposed to an empiric method, e.g., measuring performance metrics. The research method follows the steps laid out in Fig. 1.6: a concept for infrastructure supporting mPPI in hybrid mPPI ensembles was developed based on a theoretic analysis of the domain and existing infrastructural approaches. This concept was implemented in a concrete infrastructure. Subsequently, a proof-of-concept evaluation was performed, consisting of theoretic analysis, an in-depth case study and a set of applications demonstrating the practical impact of introduced concepts with respect to supporting mPPI (c.f., chapter 3).

Conceptual Framework. Similarly, contributions with respect to the conceptual framework aim to *enable* infrastructural support for mPPI design through a formal description of interaction. As such, the proposed conceptual framework of mPPI (or, more precisely PPI in general) connects interaction and infrastructure. Thereby, the purely theoretical conceptual framework can directly serve as basis of infrastructure components (here: toolkit). This allows providing support for interaction techniques at the infrastructure level. Therefore, the employed research method corresponds roughly to the proof-of-concept approach laid out above. Thereby, an additional theoretic refinement step aims to assert expressive power of the conceptual framework (see chapter 4 for details) and replaces the case study.

Interaction. Contributions with respect to interaction theory aim to *explore* mPPI, i.e., provide an initial theoretical understanding of human computer interaction in hybrid mPPI ensembles. Thus, the proposed theory of interaction in hybrid mPPI ensembles was derived using a qualitative empirical, theory developing research approach substantially inspired by grounded theory, [Corbin and Strauss, 1990]. Thereby, a profound domain analysis combined with existing theory of interaction in hybrid mPPI ensembles yielded the setup of an explorative study. Qualitative analysis of study results was used to generate an initial theory of interaction that was iteratively refined in order to obtain a deeper understanding with respect to important concepts. Chapter 5 describes the chosen research approach in detail.

1.2.2 Publications Related to this Thesis

Parts of this thesis and the research presented throughout this thesis have been previously published by the author in proceedings of international and national conferences and workshops. Concepts associated with pen-and-paper interaction in general were published in [Heinrichs, 2009]. The infrastructure for mobile pen-and-paper interaction was published in [Heinrichs et al., 2010b] and received an honorable mention at the *2nd ACM SIGCHI symposium on Engineering interactive computing systems* in 2010. The conceptual framework for describing mPPI was published in [Heinrichs et al., 2011a]. The digital grocery list application and its design were published in [Heinrichs et al., 2010a] and [Heinrichs et al., 2011b]. Finally, the empirically substantiated theory of interaction in hybrid mPPI ensembles and the associated study were published in [Heinrichs et al., 2012] at the *ACM annual conference on Human Factors in Computing Systems (CHI)* 2012. Thereby this thesis provides an in-depth report of this research an puts it into context.

1.3 Structure of this Thesis

The main structure of this thesis is based on the division between research with respect to enabling *infrastructures* and research with respect to *interaction* design. These two parts are connected through a conceptual framework of mPPI, as outlined in section 1.2. Thereby, the thesis consists of 6 chapters, out of which chapters 3 to 5 present the three essential contributions to the field, i.e., a novel mPPI infrastructure, a conceptual framework and a theory of interaction in hybrid mPPI ensembles. The following paragraphs provide an initial overview of chapters contained within this thesis.

Chapter 1. This marks the current chapter. It provides the introduction into the field of mPPI and motivates the research conducted in this thesis through a set of two studies: an ethnographic study about the use of mobile information processing tools in the context of knowledge work and a case study about digital grocery shopping and systems supporting digital grocery shopping by means of mPPI. Based on these studies, the chapter lays out the research question addressed in this thesis and provides an overview of employed research methodology.

Chapter 2. This chapter sets the stage for all coming chapters and provides a detailed introduction into the domain of PPI in general and mPPI in particular. It begins by introducing essential terminology and describing the technology enabling PPI. It then provides an in-depth review of the field and presents its application domains. An overview of the state of the art in both infrastructures for PPI (mPPI) and interaction research with respect to hybrid mPPI ensembles follows. This includes research on existing conceptual frameworks for interaction in the domain. Thereby, it discusses the deficits in existing approaches with respect to the research questions addressed in this thesis.

Chapter 3. This chapter presents the contributions with respect to infrastructures enabling mPPI. It begins by extracting the common base architecture employed in existing approaches and demonstrating its deficits with respect to supporting mPPI. Subsequently, it introduces the novel *distributed interaction processing pipeline* that allows supporting both user and document mobility, thus enabling mPPI. It then presents *Letras*, an infrastructure based on this architecture, alongside its reference implementation. A detailed analytical and proof-of-concept evaluation taking up the digital grocery list case study introduced in section 1.1.2 concludes this chapter.

Chapter 4. This chapter presents the contributions toward a conceptual framework of PPI serving as basis for toolkits. It begins by introducing W^5, a conceptual framework to formally describe interaction techniques based on first order predicate logic. It then analytically derives basic interaction predicates from a subset of interaction techniques described in the literature and demonstrates how a small set of basic interaction predicates allows supporting a broad range of existing interaction techniques. As such, it demonstrates that the approach facilitates the design of open toolkits for mPPI. A proof-of-concept evaluation describing the design of such a toolkit based on

W^5 concludes the chapter and demonstrates the practical relevance of derived theoretical findings.

Chapter 5. This chapter presents the contributions with respect to interaction design in hybrid mPPI ensembles. It begins by introducing the research approach taken in order to derive a theory of interaction in hybrid mPPI ensembles. Subsequently, it describes an extensive exploratory study aimed to develop an empirically substantiated theory of interaction. It then presents the results of this study and derives a theory of interaction in hybrid mPPI ensembles. Finally, it presents a set of design guidelines derived from this theory informing interaction design in the targeted setting.

Chapter 6. This chapter concludes the thesis. It sums up contributions, reviews the approach chosen and discusses insights obtained through the presented research. In addition, it points toward future directions for research.

2 Background and Related Work

Synopsis: *This chapter describes the state-of-the-art in (mobile) pen-and-paper interaction. It introduces important concepts and provides a detailed review of existing PPI and mPPI application areas and approaches. Following this, it establishes domain requirements for mobile settings dividing the analysis into infrastructure, conceptual frameworks and interaction theory related questions. Analyzing current infrastructures, conceptual frameworks and interaction theory for PPI, it demonstrates their shortcomings with respect to mPPI and sets the stage for the contributions described in the chapters to come.*

A growing body of research addresses the domain of *Pen-and-Paper Interaction* (PPI). Most approaches thereby focus on leveraging the benefits of PPI in specific application domains, e.g., note-taking or collaborative knowledge work. In addition, several platforms aiding the development of PPI-based user interfaces have been introduced. Only recently, the need for theoretical insights has led to initial results on how to actually design pen- and-paper interaction. However, despite the highly mobile nature of paper, most existing approaches exclusively support stationary settings, almost entirely neglecting *mobile Pen-and-Paper Interaction* (mPPI).

This chapter reviews the state-of-the-art in PPI in general and with a focus on mobility in particular. First, section 2.1 establishes the basic terminology used to describe challenges and common concepts in PPI and mPPI. It introduces the underlying technology enabling PPI and its particular requirements on supporting infrastructures, as well as its impact on the design of interaction. Second, section 2.2 reviews existing PPI and mPPI based applications. It demonstrates how introduced concepts are applied in state-of-the-art approaches and illustrates their shortcomings with respect to mobility. Third, sections 2.3 to 2.5 provide a detailed analysis of the state of the art in the three main areas of contribution of this thesis: section 2.3 analyzes how existing infrastructures support PPI and mPPI. It thereby demonstrates their infrastructural limitations in supporting mobile usage practices, i.e., *User Mobility* and *Document Mobility*. Section 2.4 analyzes existing conceptual frameworks. It introduces existing interaction techniques in PPI and mPPI and shows the limitations of existing conceptual frameworks with respect to toolkit design and expressive power. Section 2.5 describes and discusses research related to interaction design and theories of interaction with respect to PPI and mPPI, thereby demonstrating the lack of interaction theory for hybrid mPPI ensembles. Finally, section 2.6 concludes this chapter and summarizes its main arguments.

2.1 Background Pen-and-Paper Interaction

Todays computing devices provide powerful tools to access, manipulate and communicate information. However, paper remains among the most important information processing tools, especially in the context of knowledge work [Steimle, 2009b] and mobile note-taking [Yeh et al., 2006a].

The continued importance of paper can be attributed to its unique characteristics as compared to digital systems. Besides other characteristics, paper documents are lightweight, cheap, flexible with respect to usage context and physical form and extremely robust, [Signer, 2005]. Furthermore, the physical characteristics of paper naturally afford certain tasks related to information navigation and manipulation, especially in the domain of knowledge work.

Sellen and Harper showed in their studies why paper is unlikely to be replaced in the context of work environments in the near future [Sellen and Harper, 2003]: it provides unmatched navigation capabilities, e.g., by skimming through a book, supports markup during reading and allows for arrangement in physical space to aid navigation tasks. And, more than ten years later, these arguments still hold true: despite sophisticated interactive technology, paper affordances remain essential in knowledge work and creative tasks, [Harboe and Huang, 2015].

With respect to the mobile domain, it becomes apparent that paper itself is a highly mobile medium. It is very convenient to carry and share paper documents, as well as to (re-)use them in a variety of situational contexts. Pen and paper allow instantaneous interaction without annoying start-up times, e.g., easily supporting the practice of mobile note-taking [Yeh et al., 2006a]. This still provides a benefit compared to purely digital solutions, e.g., note taking applications on smartphones, e.g., such as *Evernote*[1]. Several studies corroborate these findings, e.g., [Bellotti et al., 2004, Chapman et al., 2009, Ispas et al., 2010b].

In contrast to this, digital systems offer advantages in information management and search, hyperlinking, communication etc., and of course processing and computation capabilities enabling true interactivity. Therefore integrating both worlds by bridging the gap between paper and digital systems via PPI and, in the context of mobile use, mPPI, has been the goal of numerous approaches (e.g. [Yeh et al., 2006a, Steimle, 2009a, Guimbretière, 2003] and [Tsandilas, 2012, Pietrzak et al., 2012].

[1] http://evernote.com/ (accessed: July 2015)

This section provides background knowledge on how PPI and mPPI can be realized. First, it establishes basic terminology and introduces important concepts. Second, it presents an introduction into the underlying technology of digital pens and paper.

2.1.1 Essential Terminology

In order to follow a discussion of PPI and mPPI, several important definitions and concepts are required. This section introduces these key concepts and definitions used throughout the present thesis. Additional concepts will be defined where required in the remaining chapters.

Essentially, PPI refers to interaction between a user and a digital system by means of pen and paper. Thereby, the user employs a *digital pen* producing *digital ink* on one or more *interactive regions*. This digital ink is then transfered to the digital system in one of the two essential *operation modes*. Thereby, the four essential concepts are defined as

Digital Pen A *Digital Pen* refers to a pen-shaped input device, that allows tracking its movements in relation to paper documents or other display devices. A digital system then can access any data recorded, either using wired or wireless communication technology (e.g., USB or Bluetooth). In contrast to a passive stylus, which allows for interaction only on a special digital surface, e.g., the Wacom Cintiq 24[2], a digital pen is primarily an active component designed for interaction on paper. However, several approaches allow using digital pens on digital surfaces in addition to paper [Steimle, 2009a, Hofer and Kunz, 2010, Brandl et al., 2007], thus diminishing this clear distinction.

Numerous different technologies for digital pens are available on the market. Section 2.1.2 will provide an initial overview and introduce the widespread Anoto[3] digital pen technology used in the scope of this thesis. For a detailed review of existing technologies the interested reader can refer to [Steimle, 2012].

Digital Ink The digital pen records movements and other actions, e.g., clicks, on the paper surface. This data recorded by the pen is referred to as *digital ink*. Recorded digital ink can trigger system responses, as defined in the application employing PPI. For instance, it can visualize drawings by rendering a facsimile of the digital ink on a computer display or interpret it as a command gesture.

Normally, the pen leaves physical ink traces on the paper surface during interaction, just as a regular ball-point pen. However, in some applications special

[2] http://www.wacom.com/en/us/creative/cintiq-24-hd (accessed: July 2015)

[3] http://www.anoto.com (accessed: July 2015)

non-inking cartridges are used which prevent the pen's tip from actually inking paper documents. As a result, gesturing on paper using a digital pen without physically inking it may also generate digital ink, highlighting why it is important to distinguish between physical and digital ink.

Operation Modes Two modes of operation for the pen exist which are fundamentally different from an interaction perspective. The digital pen operates either in *batched mode* or *interactive mode* [Yeh et al., 2008]. In batched mode, pen movement data is recorded and stored in the pen. Upon user request (e.g., when the user plugs the pen into a cradle, or ticks a box on paper) the stored data is transferred to the application in bulk. In interactive mode, pen data is continuously streamed to the application and enables the application to provide instant feedback to the user.

Interactive Region In order to access digital functionality, *Pen- and-Paper User Interfaces* (PPUIs, c.f., page 2 ff. in chapter 1) use specific regions, which are bound to digital functionality. Hence the most important abstraction in the system design of PPUIs is the *interactive region*. An interactive region corresponds to a physical region on paper (or on other surfaces as in [Steimle, 2009a] and [Brandl et al., 2007]), where digital ink triggers certain functionality in a digital system. An interactive region might for example allow the application to render pen movements on a screen or capture gestures and execute appropriate actions.

Interactive regions are a fundamental concept of PPUIs: each application defines at least one interactive region linking the physical and digital worlds. Because of this linking nature, the concept of interactive regions is sometimes regarded as a special form of hyperlinks, e.g., in the RSL model serving as foundation of the iServer and iPaper framework [Signer and Norrie, 2007a].

2.1.2 Underlying Technology: Digital Pens and Paper

Several technologies exist for capturing pen input on paper documents. The technology used determines which capabilities of PPUIs are available to the designer. Thus, it has a significant impact on possible interaction schemes and on the requirements to supporting infrastructures. This section provides an overview of available technologies (for an extensive review see [Steimle, 2012]). Following this, it describes the Anoto[4] technology used in most contemporary approaches in more detail. This technology drives the reference implementations of the concepts introduced in this thesis.

[4] http://www.anoto.com (accessed: July 2015)

Available Technologies

Existing approaches to capture pen input on paper can be classified into relative and absolute positioning techniques.

Relative positioning techniques. These approaches record pen movements relative to a *local* frame of reference, e.g., a paper document. A widespread relative positioning technique are Time-of-flight based approaches, e.g., the mobile digital scribe by IOGear [5]. These approaches measure the time a signal emitted by a digital pen takes to reach two (or more) reference receivers with known positions (with respect to the frame of reference). The position of the pen tip is then computed using triangulation. In this relative positioning technique, the user has to calibrate the receivers relative to paper documents, e.g., by using a clipboard to which the paper is attached.

Another relative positioning technique are inductive surface based approaches, e.g., as in the Wacom Intuos line [6] (although this product is not primarily designed for work with paper, it uses technology that can be used in combination with paper). In such approaches, the user poses a paper document onto a tablet device, typically designed as a clipboard (calibration). Then, the system determines the position of the pen tip using induction.

The advantage of such relative approaches is that they work on normal paper, without any preparation on the paper side. Their disadvantage is that they require a separate *calibration* step. Additionally, these approaches cannot determine on which paper artifact interaction occurs without additional technologies, e.g., via visual or electronic markers [Steimle, 2012].

Absolute positioning techniques. These approaches record pen movements relative to a *global* frame of references, e.g., a global coordinate system overlaying all interactive surfaces. An example for an absolute positioning technique is camera based tracing. Camera based approaches use computer vision techniques in order to track either the pen traces [Wellner, 1993] or directly the pen [Holman et al., 2005]. An advantage of such approaches is the ability to track other input, e.g., touch input, on paper using the same technology. Their disadvantage is the complicated setup, their lack of robustness and the effort required to uniquely identify on which paper document interaction occurs, despite a global frame of reference. Additionally, a line-of-sight is required to track pen movements correctly. Furthermore, camera based approaches used to be feasible in stationary settings only, ruling them out as potential design choices for mobile pen-and-paper interaction. Recent work by Liao et al.

[5] http://www.iogear.com/product/GPEN200NF1/ (accessed: July 2015)

[6] http://www.wacom.com/en/us/creative/intuos-pro-m (accessed: July 2015)

[Liao et al., 2010b] and Iwata et al. [Iwata et al., 2009], however, shows that camera based tracking might become an alternative in the future to the Anoto technology described below, even for uniquely identifying paper documents [Yang et al., 2011].

Drawbacks. The common drawback of these technologies is their impact on the interaction with paper documents. If documents require a complicated set- up phase, or only work in a very limited environment, the flexibility and naturalness of interaction is reduced considerably. Additionally, applications often require identification of paper documents to map them to certain functionality. Ideally the capturing technology covers this directly. As of today, the only technology providing such functionality is the *Anoto*[7] technology, an absolute positioning technique introduced in the next section.

Anoto Technology

Anoto offers a robust global positioning technique based on a two dimensional barcode printed on paper documents and miniature tracing camera built into the pen tip. This technology finds widespread use in both, academia, e.g., [Yeh et al., 2006a, Steimle, 2009a, Cowan et al., 2011] as well as commercial products, e.g., LeapFrog LeapReader[8] or LiveScribe Echo[9]. Anoto offers a robust mechanism of capturing pen input and the ability to uniquely identify paper documents used, is rotation invariant, i.e., the user can write from any direction on paper documents, and does not require setup or calibration. This enables this technology to preserve most usability aspects of traditional paper. The Anoto technology consists of two basic components

- a *digital pen* as input capturing device (described in the international patent [Pat, 2010])
- and a special *dot-pattern* printed on paper to encode position information (described in the international patent [Pat, 2001])

Digital Pen. The Anoto digital pen is essentially an extended ballpoint pen. It features a standard ISO 12757-2 [10] ink cartridge. For this cartridge non-inking types are also available, allowing pen usage without inking, e.g., on interactive surfaces. The

[7] http://www.anoto.com (accessed: July 2015)

[8] http://www.leapfrog.com/en-us/store/p/leapreader-reading-and-writing-system/_/A-prod21301 (accessed: July 2015)

[9] http://www.livescribe.com/en/smartpen/echo/ (accessed: July 2015)

[10] http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=23718 (accessed: July 2015)

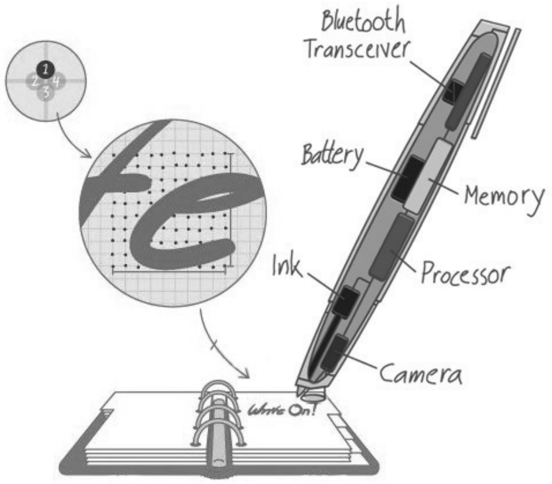


Figure 2.1: The Anoto digital pen (courtesy Anoto AB)

pen has a small built-in infrared camera, a processor, battery, memory unit and an optional bluetooth transceiver as shown in Fig. 2.1. Additionally, the pen features a pressure sensor to determine the force applied while writing, e.g., to allow interactions, such as ”clicking” on the paper surface.

A setup including a bluetooth transceiver enables the pen to communicate its position in real-time to a digital system, or to store digital ink for later retrieval using its memory unit. Therefore, the Anoto digital pen in theory supports both operation modes: batched and interactive mode, as introduced above in section 2.1.1.

In praxis, however, several different pen models are currently available on the market and their hardware configuration differs considerably. For instance, the Anoto Digital Pen ADP-301 [11] does not support the batched interaction mode, although the Logitech io2 (Bluetooth) [12] does.

Dot-Pattern. To track pen movements, the Anoto technology relies on a proprietary dot-pattern. While the pen is placed on paper documents, the built-in camera samples the current position of the pen by scanning a special dot-pattern printed on paper. Technically, this is done by beaming infrared light (via an IR LED in the pen tip) onto the paper surface and scanning reflected light for ”holes” marking the position of dots on paper. The sampling frequency again depends on the pen model and

[11] https://support.anoto.com/hc/en-us/sections/200470288-Pens-ADP-301 (accessed: July 2015)

[12] http://www.logitech.com/images/pdf/io2_with_bluetooth_data_sheet.pdf (accessed: July 2015)

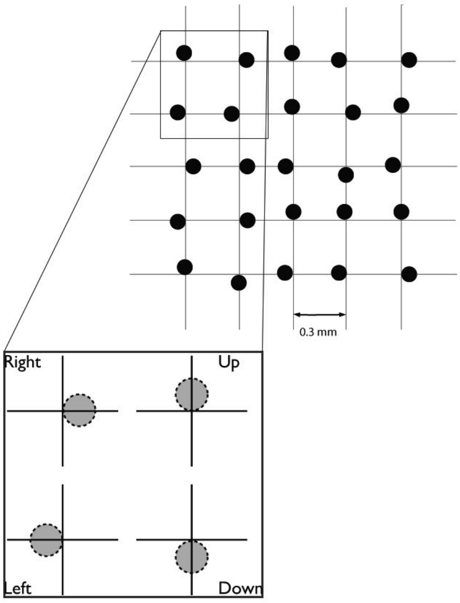


Figure 2.2: The Anoto dot-pattern used to encode position information (courtesy Anoto AB)

ranges from $100Hz$ for the ADP-301 pen, over $75Hz$ to as low as $35Hz$ for the Nokia SU-1B pen.

This dot-pattern encodes position information similar to techniques used in two dimensional barcodes. Thereby, Anoto employs an absolute positioning technique which allows for determining the pen tip position in a global frame of reference: a two dimensional cartesian coordinate system, or *pattern space*, spanning all interactive surfaces, e.g., paper artifacts. Global here means that every single position is unique inside the pattern space, e.g., two paper documents can be uniquely identified if they span different regions of the pattern space.

This is achieved by a sophisticated position encoding technique: Dots are conceptually placed in a grid, where each point is slightly displaced either to the left, to the right, upward or downward as shown in Fig. 2.2. This allows encoding $2Bit$ of position information in each grid cell. As the pen camera scans 6 by 6 grid cells in each sample, the position information encoded in each sample corresponds to $72Bit$. This mechanism is described in detail in the international patent [Pat, 2001].

With a nominal grid spacing of $0.3mm$, the resolution of the pattern is almost $850DPI$. Due to the small size of the dots, which lies between 30 and $50\mu m$, the human eye perceives the pattern only as a grayish surface which does not confound the visual structure of the document itself (comparable to recycled paper). However, in theory it is even possible to print the pattern completely invisible to the human eye: if the ”dots” are printed with invisible ink absorbing IR light, yet the remaining contents of the document reflect IR light, the IR camera built into the pen will detect dots even if the human eye cannot.

Paper and Beyond. Using the described absolute positioning technique, Anoto pens can determine and communicate their current position on a given sheet of paper. In addition, the system can uniquely identify the paper document the pen is moving on, as each paper document can be assigned to a certain partition of the pattern space. Here each paper document forms a special interactive region.

However, use of the pattern is not limited to paper: Brandl et al. demonstrated how to print the pattern to a special semi-transparent foil, enabling the construction of large interactive surfaces [Brandl et al., 2007, Brandl et al., 2008, Leitner et al., 2009]. In their setup, the foil served at the same time as diffuser in a rear-projection tabletop and interactive whiteboard. Here the digital pen becomes an input device on these surfaces in addition to touch input.

One drawback of their solution was, that it did not support using the same digital pen to simultaneously work with paper documents and interactive surfaces. Therefore Steimle introduced a combined rear-projection setup featuring an Anoto enabled tabletop system in combination with several interactive paper documents [Steimle, 2009a]. Here the user could work with both, physical paper documents and interactive surfaces simultaneously, enabling tight and instantaneous integration. This was the basis for bridging the disruptive media transition between paper documents and computer, e.g., by supporting the ad-hoc creation of cross-media hyperlinks between paper and digital documents [Steimle et al., 2008a].

This idea was extended by Liwicki et al. into a sophisticated mixed touch and pen input tabletop system [Liwicki et al., 2010]. In these approaches, semi-transparent foil was employed in a rear-projection setup. Hofer successfully demonstrated that it is possible to use the same technique on LCD computer screens [Hofer and Kunz, 2010], taking the concept of ubiquitous availability of Anoto enabled surfaces one step further.

Applicability to PPI. The major advantage of the Anoto technology lies in its minimal adverse effect on the handling of paper documents and thus on interaction between user and digital systems in a PPI based application. Using a digital pen, the user can start writing immediately without a calibration step. Also the user can pause at any time. Additionally, the pattern based movement tracking proves to be very robust coping with any orientation of the paper sheet (or foil) used. From the infrastructural point of view, Anoto allows mapping interactive regions directly to partitions of the global pattern space. This does not require additional tracking technology to identify the interactive region where interaction occurs on. Furthermore, interactive regions are not limited to paper alone, considerably increasing the flexibility of this technology. Based on these reasons, Anoto technology will be used as main representative of digital pen and paper technology throughout this thesis.

2.2 Application Domains: Categorization and Related Work

Pen-and-paper interaction has been successfully employed in numerous application domains. Thereby, applications range from scientific prototypes to commercial products. However, mPPI based applications are still rare despite several recent prototypes of mobile applications exemplifying their potential. This section provides an overview and categorization of application domains described in the literature, such diverse as note-taking, active reading and education, as well as document editing and form filling. It demonstrates the potential of mPPI in these domains and discusses the extent to which existing approaches actually support mobile practices. As such, this section sets the stage for sections 2.3 to 2.5, where related work with respect to enabling infrastructures, conceptual frameworks and theory of interaction is analyzed in detail.

2.2.1 Note-taking

Note-taking represents an important class of applications for mobile PPI. The inherent flexibility and mobility of paper documents makes them ideal tools for quick information documentation, while their robustness allows using paper documents in a plethora of environments, e.g., during field research trips [Yeh et al., 2006a]. Consequently, several note-taking applications based on PPI have been described throughout the literature.

Note-taking: Existing Systems

Existing note-taking systems employing PPI and mPPI exist both for stationary settings and mobile note-taking. In addition to research prototypes, there are several commercial note-taking applications based on PPI available.

Stationary Note-Taking Applications. *NiCEBook*, [Brandl et al., 2010], is a generic note-taking solution based on PPI. In this approach, the user writes notes with a digital pen into a special paper notebook. A digital version of the notebook stores digital ink generated and provides retrieval and search capabilities. *ARENO*, [Ispas et al., 2012], a system for continuously capturing notes and ”todo” items in a desktop environment, takes a similar approach. These applications are designed for stationary use on a desktop computer only; mobile characteristics of note-taking are not supported. However, both applications show the potential of synchronous interaction with a digital system using the interactive mode of the digital pen.

Hurlbutt and Klemmer reported on a note-taking solution supporting programmers in early planning and requirement elicitation phases [Hurlbutt and Klemmer, 2006]. Here, user stories and programmer tasks are captured on handwritten paper cards. This allows for convenient note arrangement and input. However, the system employs only the batched interaction mode and is exclusively designed for Desktop PC use. Similarly, Liwicki et al. reported on a shopping list application, which uses a sophisticated handwriting recognition approach coupled with a product ontology to identify handwritten shopping list items [Liwicki et al., 2011]. However, it mainly targets a stationary setting; mobile use is only supported through batched interaction.

Mackay et al. introduced *A-Book*, [Mackay et al., 2002], an application supporting the note-taking needs of biologists in a hybrid paper / digital laboratory notebook solution. This application consists of a capture unit for digital ink, formed by a tablet PC posed underneath normal paper, and a system to link written content to digital resources. A later version of the application, *Prism*, [Tabard et al., 2008], added support for Anoto digital pen technology and collaboration features, e.g., sharing of notes. In *A-Book*, the user can issue and follow links between content written on paper and digital entities [Mackay et al., 2002]. Thereby, a PDA device serves as a looking glass, i.e., it displays additional digital information besides the contents on paper.

Although the prototypical setup itself is not mobile, *A-Book* demonstrates the potential of hybrid mPPI ensembles for note-taking applications: Employing the *eye-in-hand* metaphor [Fitzmaurice, 1993], it utilizes a tight integration of paper and mobile digital device to create compelling information management tools.

Mobile Note-Taking Applications. Toward mobile note-taking, Yeh et al. introduced *ButterflyNet*, [Yeh et al., 2006a], a mobile note-capturing application supporting the note-taking needs of field biologists. It allows associating handwritten notes with multimedia content captured during a field research trip, e.g., digital photographs. Content is associated either using an automatic approach based on capture time, or by issuing a special gestural command (c.f., the *Hotspot Association* gesture described in section 2.4.1). A Desktop PC application then combines these data sources later, when digital ink is transfered to the PC. Subsequently, the system presents notes recorded in a browsable multimedia notebook, as shown in Fig. 2.3. Although capture of notes

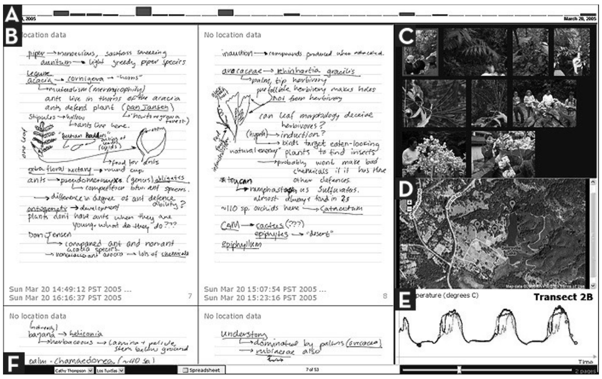


Figure 2.3: The ButterflyNet Notes Browser (source [Yeh et al., 2006a])

and digital contents is designed for mobile use, i.e., to support user mobility (document mobility is not supported), it only employs the batched interaction mode of the digital pen. Accessing interactive functionality in the field is not supported.

Weibel et al. took a similar approach based on temporal association of data in *ChronoViz*, [Weibel et al., 2011a, Weibel et al., 2012]. This system supports observational and behavioral research practices involving paper notes [Weibel et al., 2012]. Here, handwritten notes and data from other sources, e.g., camera recordings, are aligned and displayed at the desktop computer on a single timeline. The combined data can be navigated based on capture time, to quickly discover interesting regions in long video recordings. Subsequent research extended the correlation of video data and digital ink, e.g., to analyze movement trajectories in observational research [Fouse et al., 2013]. Although capturing and association are also designed for mobile use in *ChronoViz*, the approach only explores the batched operation mode of the digital pen [Weibel et al., 2012]. Mobile access to the notebook, or captured data is not possible.

S-Notebook, [Pietrzak et al., 2012], extends the ideas above by allowing users to associate paper notes with a broad range of specific parts of digital artifacts, e.g., parts of videos, images, pages in pdf documents. In contrast to the other mPPI based note-taking approaches described above, *S-Notebook* explores the interactive mode in mobile settings. Here the user has to initiate association by using a touch gesture on the mobile device during note capture. As such, this approach exemplifies the potential of note-taking systems supporting user mobility (although document mobility remains unsupported) and provides a valuable first step toward supporting mobile practices associated with note-taking.

Commercial Systems. Several commercial products aim to support the note-taking and retrieval process in addition to - or derived from - the scientific prototypes introduced above. An example for a wide-spread commercial application is the *Live-Scribe echo smartpen*[13]. It allows capturing notes in specially prepared notebooks directly on the pen and later transfer these notes to a desktop PC to aid searching and browsing of notes, combining them with audio recordings. Another, similar note-taking solution is the *Oxford Easybook*[14] and *capturx*[15], an application providing integration into Microsoft OneNote. Additionally, a note-taking application based on digital pens has been developed to support police report digitization in Wuhu city, *China*[16]. These applications only support the batched interaction mode, where the digital pen stores captured digital ink until the pen is synchronized with a desktop PC. Mobile interaction with a digital system, e.g., the instant sharing of a design sketch, and the exploration of the interactive mode remains unsupported.

The only commercial system currently supporting mobile note-taking based on mPPI is offered by LiveScribe through their novel *LiveScribe Smartpen 3*[17]. It allows integration of handwritten notes with several cloud based note-taking services, e.g., Evernote[18]. As this application also supports the interactive mode of the pen, this provides a valuable first step toward leveraging the full potential of mPPI for note-taking applications. However, the lack of an accessible, shared infrastructure and missing support for document mobility leave considerable room for improvement.

Note-taking: Practices

Studies of note-taking behavior have shown that note-taking applications need to satisfy the user's need of a quick and flexible, incidental note-taking support in combination with adequate storage and retrieval in mobile settings [Lin et al., 2004]. Thereby, a fast and convenient note entry is most important to the user, even favored over storage and retrieval functionality [Kim et al., 2009].

Dai et al. established in an extensive user study that users favor recallability over accuracy of notes [Dai et al., 2009]: Rather than having a completely accurate digital representation of notes, users would opt for quicker note entry as long as the over all structure of the stored notes allows recall of their contents. This especially holds for handwritten notes, as the layout and graphical representation provide additional visual cues helping recall [Ispas et al., 2010b].

[13] http://www.livescribe.com/en/smartpen/echo/ (accessed: July 2015)

[14] http://www.oxfordeasybook.com/ (accessed: July 2015)

[15] http://www.adapx.com/products/capturx-onenote (accessed: July 2015)

[16] http://www.inphoactive.com/wp-content/uploads/2012/07/WuHuCity.pdf (accessed: July 2015)

[17] http://www.livescribe.com/en/smartpen/ls3/ (accessed: July 2015)

[18] https://evernote.com/ (accessed: July 2015)

In this context, pen and paper interaction provides a natural, fast and convenient way to support note entry. Using the instant-on functionality of digital pens and the robustness of paper, this modality facilitates entering note contents into digital systems, [Andrew et al., 2009]. This comes at the cost of less accurate and more error-prone entry, e.g., as handwriting recognition introduces recognition errors. However, as argued above, recallability and convenient entry pose more important criteria in the design of note-taking systems.

Chapman et al. report on an extensive ethnographic study concerning the value of PPI based note-taking solutions in various contexts [Chapman et al., 2009]. They studied the note-taking behavior of a set of professionals and students in the US and Japan over the period of one year. Additionally, they conducted on-site visits in several cooperations, to examine currently established practices around pen and paper. Their findings show that although participants acknowledge the added value of using PPI based note-taking solutions, several barriers exist with respect to successful large-scale deployment of these solutions.

Partly these barriers are strictly technical limitations associated with the employed digital pen technology, i.e., the necessity to provide paper with the Anoto dot-pattern. However, further barriers point toward infrastructural challenges: the main barrier forms the lack of of support for mobile practices such as user mobility, e.g., review and edit notes anywhere, and document mobility, e.g., use any paper artifact and share notes [Chapman et al., 2009]. The findings of Ispas et al. further corroborate this fact: although users in general acknowledge the added value of digital pen based note-taking solutions, they are reluctant to change existing practices revolving around pen and paper [Ispas et al., 2010b].

Note-taking: Discussion

Related work in PPI based note-taking systems shows that the domain is an important application domain for PPI in general and mPPI in particular. Numerous PPI based approaches exist for general note-taking systems and note-taking support for expert groups. Research about note-taking practices underlines the need for systems supporting pen based note entry and mobile usage practices, e.g., interactively editing paper and digital contents in a mobile setting (user mobility) and passing paper artifacts on to others (document mobility).

Current approaches mostly lack support for these mobile practices. Most existing systems either focus on stationary settings, or limit mPPI support to batched interaction. Only a small set of recent approaches explores the interactive operation mode of the digital pen toward offering user mobility at an application level. Despite neglecting support for other mobile practices, e.g., document mobility, these approaches demonstrate the potential of mPPI based note-taking applications.

However, solutions at an application level do not offer insights into the design of infrastructures, conceptual frameworks and interaction theories for mPPI.

2.2.2 Knowledge Work: Active Reading and Education

Another important application domain application domain for PPI is active reading and knowledge work, [Hong et al., 2012], as well as the related field of education [Oviatt et al., 2006]. Recent studies confirmed the fundamental importance of paper artifacts and their affordances with respect to practices associated with knowledge work [Harboe and Huang, 2015]. As such it provides a prime use cased of PPI. Although not as inherently mobile as note-taking, this domain contains several mobile and nomadic use cases. Active reading needs support in the library, at home and in the office. Education encompasses classroom activities, field trips and exam preparation that might occur in mobile or nomadic situations.

Knowledge-Work: Existing Systems

It has been shown, that reading on paper better supports cognitive processes than reading on a computer screen [O'Hara and Sellen, 1997]. This is partly because paper documents facilitate navigation [Sellen and Harper, 2003], but also due to the convenient way in which pen and paper allow handwritten notes and annotations supporting the cognitive processing of contents. Thereby, the process of simultaneously reading, taking notes, underlining, annotating, excerpting and in general ”working” with the contents of a document is referred to as *active reading*.

At the same time, digitally augmented workplaces offer interactive capabilities required by today's knowledge workers, in particular when combined with digital pen and paper technology [Gebhardt et al., 2014]. As such, digital pen technology offers a convenient tool supporting practices of knowledge workers, while at the same time opening the stage for the use of digital information, e.g., the immense knowledge source of the web [Steimle et al., 2008c]. Thus, several applications have been designed to support active reading by means of PPI.

Active Reading. Norrie et al. introduced a special digital library setup, where users can follow hyperlinks printed on paper using PPI to access digital content on computers in the library [Norrie et al., 2008]. Similarly, *PLink*, [Steimle et al., 2011], supports active reading at the office desk connecting digital and physical workplaces through PPI. Here users can link paper artifacts and quick notes on large paper surfaces with digital resources, e.g., web sites, through cross-media links. This concept has also been used in commercial applications designed to support education and active reading, e.g., the Leapfrog Leap Reader system[19]. The targeted setup of these approaches is, however, a stationary environment; mobile use is not supported. In addition, these systems lack support for many practices typically associated with active reading, e.g., content structuring and collaboration.

To overcome this, Steimle et al. introduced *CoScribe*, a sophisticated PPI based learning platform for students [Steimle et al., 2008c, Steimle et al., 2008a]. It supports content structuring and collaborative practices, e.g., sharing of structuring and indexing information [Steimle et al., 2008b]. It can be stationary deployed on Desktop PCs as well as on Tabletop computers. In the Tabletop setting, *CoScribe* features intuitive interaction techniques to issue cross-media links, i.e., links between interactive regions on paper and such regions on digital documents [Steimle, 2009a]. In a similar approach using pen and touch enabled table-top displays, Matulic and Norrie found that this combination supports active reading practices far better than normal pen and paper combined with digital tools, i.e., without digital pen support [Matulic and Norrie, 2012]. As such it poses a valuable application domain for PPI, however, neglecting the factor mobility.

With respect to knowledge work beyond active reading, the domain of education provides a vital field of study for PPI and mPPI based approaches: recent studies have shown that despite the broad availability of digital media today, pen and paper remain the predominant tools for knowledge work in the context of education [Malacria et al., 2011].

Education. Applying PPI to support education in general, beyond active reading alone, has been emphasized by Oviatt et al. [Oviatt et al., 2006]. They showed in a comparative study that PPI outperforms other types of interaction, i.e., tablets with stylus input and PCs, when it comes to classroom use due to the un-intrusive interface. PPI based systems let the user concentrate more on the main task, e.g, solving a mathematical problem, and introduce less interruptions at the interaction level. Thus, PPI based systems support the cognitive processes of learning better than purely digital systems [Oviatt et al., 2006]; especially in the mastering of complex skills, e.g., in mathematical education [Oviatt et al., 2007, Leitner et al., 2010]. In this context Oviatt, Cohen and Weibel also published a research corpus of data on mathematical education consisting of digital ink, speech captures and photos that was obtained using digital pen technology [Oviatt et al., 2013].

Toward PPI based applications for education, Miura et al. introduced a PPI based system designed for interactive classroom use [Miura et al., 2007]. The central aspect of this system is communication between students and teacher: the teacher can adapt content and explanation during a lecture to the students' needs by interactively

[19] http://www.leapfrog.com/en-us/store/c/leapreader/_/N-82g (accessed: July 2015)

following the notes and questions written by students. To achieve this, each student uses a digital pen to take notes and solve exercises [Miura et al., 2010], which the teacher can subsequently review. Real-time processing of notes is supported through the interactive operation mode [Sugihara et al., 2010b]. The system design is thereby based on an extensive study on the respective needs of students and teacher [Sugihara et al., 2010a]. However, mobile settings, e.g., during homework or self-study periods, are not supported.

PaperCP, [Liao et al., 2009], follows a similar approach enabling real-time interaction between students and teacher through PPI. Here university students can take notes and submit questions, sketches, exercises etc. in real-time on an anonymous submission channel to the instructor. Similarly, *U-Note*, [Malacria et al., 2011], explores temporal association of digital, e.g., audio, web pages, slides and videos, and paper based learning material through PPI. These approaches provide valuable first steps toward mobile systems in this context as, e.g., *U-Note* provides access to material on a student's personal mobile device. However, they neither support full user mobility, e.g., data entry in mobile settings, nor consider document mobility.

Knowledge-Work: Discussion

Related work regarding knowledge work, especially with respect to active reading and education, shows that this domain too is an important application domain for PPI. Pen and paper are essential tools in contemporary knowledge work practices. Their benefits have been clearly shown in multiple contexts, e.g., facilitating cognitive process associated with learning. PPI has been successfully applied to provide support for knowledge work.

Thereby, existing approaches mostly focus on stationary settings, although mobile use cases for knowledge work exist, e.g., nomadic use in the library or mobile support for learning during field trips. Initial steps toward mPPI, e.g., by providing mobile access to learning material including notes taken on paper, exemplify the potential of mPPI based applications for knowledge work. However, the full potential of mPPI, as well as supporting infrastructures, conceptual frameworks and interaction theories, remains yet to explore.

2.2.3 Document Editing and Form Filling

Working with digital documents often involves work with paper versions (print- outs) of these documents [Sellen and Harper, 2003, Norrie et al., 2006b]. Thereby changes to the physical version of the document, annotations and corrections need to be transferred back to the original digital version. To automate or support this process, several approaches rely on digital pen technology. Here PPI helps in bridging the gap between the digital and physical worlds, rendering document editing another important application domain for PPI. Although these activities traditionally occur in a controlled office environment, several mobile scenarios exist: collaboratively editing meeting minutes, working with documents in nomadic settings, e.g., to efficiently use idle time during a train ride, or adapting reports during field trips.

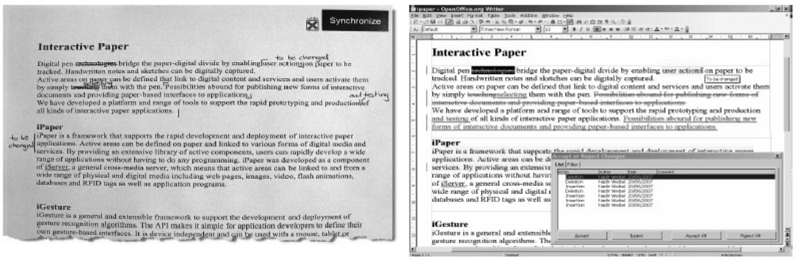
Document Editing: Existing Systems

Existing systems providing PPI based support for document editing can be classified into two categories: systems providing generic support for *editing documents* through paper printouts of digital documents, i.e., systems targeting the document life-cycle, and form filling systems, i.e., systems aiming to elicit (handwritten) data in forms to facilitate data capturing.

Document Editing. *PADD*, [Guimbretière, 2003], supports the iterative workflow of editing a document in the digital world, printing it, editing the printout and finally integrating the changes back into the digital document. Thereby *PADD* stands for *Paper Augmented Digital Documents*. The system consists of a client-server architecture, where a central document server manages digital documents, processes PPI and provides a central database storing annotations on documents. *PapierCraft*, [Liao et al., 2005, Liao et al., 2008], provides a gesture based command system on top of *PADD*. It supports interactive document editing, e.g., copy and paste gesture commands, using the interactive operation mode of the pen. Both systems are exclusively designed for stationary use. However, the *PADD* system forms one of the earliest systems targeting infrastructural support for PPI (c.f., section 2.3.2).

iDoc, [Weibel et al., 2007], supports document editing based on the *iServer / iPaper* infrastructure (c.f. section 2.3.2). Thereby, *iDoc* supports PPI based document editing during the complete document life-cycle (create- print-edit) [Weibel et al., 2007]. A later extension of the *iDoc* document editing system interprets the semantic meaning of annotations and integrates these either into the document (changes), or into a personal knowledge base [Liwicki et al., 2009]. The *iServer / iPaper* infrastructure fundamental to *iDoc* has also been used to develop various domain specific document editing applications, e.g, *PaperProof*,[Weibel et al., 2008], as depicted in Fig. 2.4. Here, scientific papers can be proof-read and corrected in their paper version. Thereby, the system feeds all changes back to the digital document version.

Although these systems support stationary use only, *edFest*, [Signer et al., 2006, Signer et al., 2007a, Norrie et al., 2007], forms an early, yet simple mobile application using the same infrastructure. Here the user can access multi-media content on a mobile device by clicking with the digital pen on a paper map. Thereby the approach focuses rather on content access than document editing and sharing in the mobile



Interactive Paper

Digital pen ~~technologies~~ bridge the paper-digital divide by enabling user actions on paper to be tracked. Handwritten notes and sketches can be digitally captured.
Active areas on paper can be defined that link to digital content and services and users activate them by simply ~~touching~~ them with the pen. Possibilities abound for publishing new forms of interactive documents and providing paper-based interfaces to applications.
We have developed a platform and range of tools to support the rapid prototyping and production of all kinds of interactive paper applications.

iPaper

iPaper is a framework that supports the rapid development and deployment of interactive paper applications. Active areas can be defined on paper and linked to various forms of digital media and services. By providing an extensive library of active components, users can rapidly develop a wide range of applications without having to do any programming. iPaper was developed as a component of iServer, a general cross-media server, which means that active areas can be linked to and from a wide range of physical and digital media including web pages, images, video, flash animations, databases and RFID tags as well as application programs.

iGesture

iGesture is a general and extensible framework to support the development and deployment of gesture recognition algorithms. The API makes it simple for application developers to define their own gesture-based interfaces. It is device independent and can be used with a mouse, tablet or

ipaper - OpenOffice.org Writer

Interactive Paper

Digital pen technologies bridge the paper-digital divide by enabling user actions on paper to be tracked. Handwritten notes and sketches can be digitally captured.

To be changed

Active areas on paper can be defined that link to digital content and services and users activate them by simply ~~touching~~selecting them with the pen. ~~Possibilities abound for publishing new forms of interactive documents and providing paper-based interfaces to applications.~~
We have developed a platform and range of tools to support the rapid prototyping and production and testing of all kinds of interactive paper applications. Possibilities abound for publishing new forms of interactive documents and providing paper-based interfaces to applications.

iPaper

iPaper is a framework that supports the rapid development and deployment of interactive paper applications. Active areas can be de
services. By providing an extensive
range of applications without havin
of iServer, a general cross-media se
wide range of physical and digital r
databases and RFID tags as well as

iGesture

iGesture is a general and extensible
gesture recognition algorithms. The

Accept or Reject Changes

Action	Author	Date	Comment
Deletion	Nadir Weibel	20/09/2007	
Deletion	Nadir Weibel	20/09/2007	
Insertion	Nadir Weibel	20/09/2007	
Deletion	Nadir Weibel	20/09/2007	
Insertion	Nadir Weibel	20/09/2007	
Insertion	Nadir Weibel	20/09/2007	
Insertion	Nadir Weibel	20/09/2007	

Figure 2.4: PaperProof Document Editing Tool (source [Weibel et al., 2008])

domain. However, it supports the interactive operation mode in mobile settings and provides a valuable first step toward supporting user mobility at an application level.

iJIT, [Ikeda et al., 2006b, Ikeda et al., 2006a], follows a similar approach: it manages links between handwritten notes on paper artifacts and digital documents including handwriting on printouts of documents [Konishi et al., 2007]. *iJIT* has been successfully deployed in an office context where all print-outs have been prepared to support digital pen input [Fujisawa, 2007]. Thereby, *iJIT* aims to avoid information loss between handwritten notes and digital documents. However, *iJIT* only supports a strictly stationary setting.

Besides research prototypes, several commercial products support document editing through PPI, including the *Anoto LivePdf suite*[20] supporting annotations on top of PDF documents and *capturx*[21]. However, these approaches support only information on top of digital contents in a strictly stationary setting, i.e., a desktop environment. Although supporting the basic print cycle, they lack support for active document editing, e.g., as in *PaperProof* [Weibel et al., 2008].

Form Filling. Form filling is another use case closely related to document editing. Instead of adding information or editing a document on paper, the document is used as means of structured data entry. Form filling provides an important industrial use case, e.g., for maintenance work and health-care appliances. Therefore, several commercial solutions support it, e.g., *Mi-Forms* [22] or the *Anoto Live Forms suite* [23].

Typically, form filling applications based on PPI focus on quick evaluation of data entered combined with convenient data entry. An application exemplifying this qual-

[20] http://www.livepdf.net/ (accessed: July 2015)

[21] http://www.adapx.com/products/capturx-markup-pdf (accessed: July 2015)

[22] http://www.mi-corporation.com/mi-forms-mobile-forms/mi-forms-software/mi-forms-client/ (accessed: July 2015)

[23] http://www.anoto.com/enterprise/products/anoto-live-forms/ (accessed: July 2015)

ity is the *Augmented Patient Chart*, [Zamarripa et al., 2007], the reports filled in by radiologists [Sonntag et al., 2011b, Sonntag et al., 2011a] or the work of trauma units [Kusunoki et al., 2013]. Here data can be recorded quickly in a structured way, while at the same time a less socially intrusive entry process allows for better communication with patients, reduces entry error rate and reduces workload on medical personnel. In addition, existing digital systems in the hospital can be integrated into the process, e.g., as suggest by [Kusunoki et al., 2013]. These systems, however, focus on supporting stationary, non-mobile use of PPI. Where they do support mobile use, they rely on the batched operation mode of the digital pen thus neglecting fully interactive mobile applications.

PartoPen, [Underwood et al., 2013a], takes a first step toward supporting mobile applications based on the LiveScribe platform. This system supports health-care professionals in the developing world in recording paper partographs. Here a quick, interactive evaluation of digital ink ensures a more structured data entry and hence to improve recorded data [Underwood et al., 2013b]. Furthermore, digital pen technology allows supporting medical personnel under stress by offering a robust, yet convenient way to data entry combined with reminder functions for eliciting critical data [Underwood et al., 2013b]. However, the LiveScribe platform limits interaction to the digital pen alone, e.g., by letting the pen signal when data is missing. The approach lacks integration of pen, paper and smartphone, as well as support for other mobile usage practices, e.g., document mobility.

Besides medical applications, PPI supports quick evaluation of form contents in settings where a large number of identical forms needs to be evaluated, e.g., in the case of fast projection of election results [Arzt-Mergemeier et al., 2007]. In this context, *Check Mates*, [Vines et al., 2012], presents a somewhat special form filling application supporting digital payment schemes for elderly people: It combines traditional paper checks with digital pen technology to facilitate check clearance. Recent advances in signature verification for digital ink [Malik et al., 2012] make this a promising trail of research, as paper as a physical token affords a more secure interaction, e.g., toward *non-repudiation*, compared to digital systems alone. Although these approaches mainly target applications in mobile domains they either rely on batched interaction alone, or on stationary deployments, i.e., Desktop PCs, to process data.

Document Editing: Discussion

Related work with respect to document editing and form filling shows that this is a relevant application domain for PPI. Numerous approaches support PPI based document editing throughout the document life-cycle; paper printouts of digital documents allow to attach digital ink to documents and edit structure as well as contents of those documents. Form filling applications support structured bulk data entry through digital pens using the interactive processing capabilities of digital systems for content validation and verification. Thereby, early approaches providing PPI based document editing support in stationary settings have particularly contributed to the design of supporting infrastructures, e.g., *PADD*, [Guimbretière, 2003], and *iServer / iPaper*, [Norrie et al., 2006a] (c.f., section 2.3.2).

Existing approaches explore the domain focusing mainly on stationary settings although mobile use cases exist for document editing in general and form filling in particular, e.g., nomadic document editing and mobile form filling applications. Most systems supporting mobile use cases limited PPI to employ the batched operation mode of the digital pen. This severely impedes mobile practices such as user mobility and document mobility. However, initial prototypes such as *EdFest*, [Signer et al., 2007a], demonstrate the potential of fully interactive mobile applications yet are hindered by infrastructures designed for stationary settings.

2.2.4 Other Application Domains

Several other promising applications domains exist for PPI and mPPI. Pen based input naturally supports free-form inking, drawing and writing. Thus, several applications support design related activities, e.g., sketching, through PPI. For similar reasons, a section of PPI based applications support applications in therapy, e.g., for elderly people, as well as collaboration and interaction on large scale, or "difficult to control" displays. Furthermore, the easy pointing and navigation on paper surfaces prompts several applications requiring a convenient remote control functionality to employ PPI. Finally, several applications supporting command and control scenarios leverage the combination of physical affordances of paper and digital systems through PPI.

Creativity

In a structured sketching approach, Dachselt et al. extended a UML modeling tool by PPI enabling intuitive input stimulating creativity within the formal visual alphabet of UML [Dachselt et al., 2008]. Thereby, the combination of pen and touch offers a novel palette of interaction techniques supporting collaborative activity in structured diagram design [Frisch et al., 2010]. Digital pen technology also naturally supports collaborative sketching in co-located sessions [Geyer et al., 2012]. Holzmann and Vogler extended this notion in a prototyping system: users can sketch user interfaces for mobile devices generating a paper prototype that can be converted to an executable prototype on the mobile device [Holzmann and Vogler, 2012].

Modelcraft, [Song et al., 2006, Song et al., 2009b], extends sketch based input into 3D space, by introducing a set of PPI based gestures for manipulating the digital representations of 3D paper models. This allows supporting creative techniques for architects during early model building phases. *Catch-Up 360*, [Perteneder et al., 2015], follows a similar approach and facilitates remote collaboration among industrial designers by enabling PPI based input on the surface of realworld, three dimensional models. This setup also adds projection based feedback to editing operation carried out with digital pens.

Toward supporting other groups of specialists, Tsandilas et al. demonstrated in *Musink*, [Tsandilas et al., 2009], how visual language drawing techniques can be used to support music composition via PPI. In their approach, the composer can write notes onto paper. Subsequently, notes will be transformed into playable midi sequences in the digital system. *PaperComposer*, [Garcia et al., 2014a], bases on the same concepts, yet adds a full-blown specialized PPUI builder for musical user interfaces, i.e., interactive regions triggering playback of audio files. *PaperTonnetz*, [Garcia et al., 2013], provides an application employing such a musical PPUI. It offers musicians the interactive exploration of two dimensional structures on paper, that correspond to audio sequences, i.e., playback of a *Tonnetz*[24].

However, these approaches exclusively target stationary settings for drawing, despite creative activities often involving the factor mobility, e.g., in creative discussions outside the normal workplace.

Mobile Creativity. Toward mPPI, Tsandilas introduced a mobile system extending *Musink* to help users interpret digital ink during creation using a combination of digital pen and touch input on smartphones [Tsandilas, 2012]. The system strikingly demonstrates the potential of hybrid mPPI ensembles: as in this particular task correct recognition of digital ink becomes crucial, it allows users immediately reviewing recognition quality (and correcting potential misinterpretations) of digital ink. This heavily bases on supporting user mobility and employing the interactive operation mode of the digital pen. Furthermore, digital pen technology offers the creative freedom and flexibility required in music composition tasks which cannot be supported by traditional, GUI based systems alone, [Garcia et al., 2014b].

PaperCAD, [Lee and Stahovich, 2014], represents another interesting step toward mobile creative support demonstrating the potential of mPPI and hybrid mPPI ensembles. It enables engineers to interactively explore *computer aided design* (CAD) drawings in mobile settings. As such it provides acoustic information and videos associated with certain interactive regions overlaying CAD documents. However, its PDA based prototype lacks supporting infrastructure that would allow other applications to share and access its resources, i.e., falls short of supporting document mobility.

UbiSketch, [Cowan et al., 2011, Weibel et al., 2010a], is a mPPI based application for communication in social networks using small sketches. The system continuously

[24] German for ”tone network”.

streams digital ink of user drawings on paper to a smartphone. Subsequently, the digitized drawings can be published on a social network site as images. It supports user mobility and the interactive operation mode in a limited way, yet neglects other mobile aspects, e.g., document mobility.

Recently, Ha, Park and Lee introduced a low-fidelity prototyping application based on mPPI, [Ha et al., 2014]. It enables designers to draw prototypes of mobile applications and directly transform these paper prototypes into executable models using mPPI. Their applications demonstrates the potential of hybrid mPPI ensembles in the application development process: paper prototypes become executable directly on targeted mobile devices.

Therapy

Other applications of PPI focused on user groups with special needs. Piper et al. presented an application in speech therapy, where users suffering form aphasia and apraxia can train their speech abilities with the help of the LiveScribe Pulse Smartpen [Piper et al., 2010, Piper et al., 2011]. Similarly, *Memento*, [West et al., 2007], aims to support elderly people in episode recall. It provides a multimedia scrapbook application on a desktop PC, where multimedia content (photo, video, audio) can easily be annotated and managed using PPI.

Piper, Weibel and Hollan found in a long term study investigating a similar approach that PPI offers a huge gain for social interaction and communication at an advanced age [Piper et al., 2013]. The authors subsequently introduced the design of a PPI based application assisting in communication therapy for elderly people, [Piper et al., 2014]. This also presents the first step towards mPPI: the LiveScribe platform used in their prototype allows for mobile use although limited to user mobility, i.e., document mobility is not supported.

Recently, Prange et al. successfully applied PPI in an application assisting dementia patients in social communication and preservative training of cognitive abilities, [Prange et al., 2015]. In their approach, patients control robot companions through a mixture of natural language and PPI. Thereby, the robot companions provide assistance in certain cognitive tasks, e.g., reminder functions.

Collaboration

Large paper surfaces can support co-located collaboration, e.g., for safety critical applications in a military setting [Cohen and McGee, 2004] or in collaborative knowledge work [Nyu and Miura, 2011]. An early application of this concept is the prototype introduced by McGee et al. [McGee et al., 2002] (c.f., command and control applications below): Here users can orchestrate missions using a paper map as central discussion and planing board and PPI to execute command and control tasks.

GigaPixel Prints, [Yeh et al., 2006b], further explores this idea through several small prototypes. In this approach, large sheets of paper serve as displays with a high spatial resolution in combination with a low temporal resolution (re-printing). This concept has been extended by Haller et al. in their augmented meeting room [Haller et al., 2010]. Their approach enables interaction with several digital systems deployed in a meeting room including large print-outs, tabletop PCs and wall sized interactive surfaces through PPI in combination with Anoto enabled surfaces. This concept supports, e.g., collaborative product design activities employing the large screen real-estate of the augmented meeting room and the creative freedom of PPI [Geyer and Reiterer, 2010].

In addition to support during meetings and collaborative activities, PPI lends itself to document and share meeting minutes [Ispas et al., 2010a]. Furthermore, Weibel et al. explored PPI based support for collaboration over distance using a remote sketching tool [Weibel et al., 2011b]. Users can draw on individual paper sheets using digital pens, while a shared virtual drawing board visualizes the drawings of all collaborators. Collaboration is further facilitated by using Skype[25] as voice-over-ip environment, where users can discuss their ideas while drawing.

However, these approaches exclusively target stationary settings, mPPI based support for mobile collaborative practices remains unexplored.

Paper as Remote Control

Signer and Norrie demonstrated that linking physical regions on paper to digital functionality facilitates PPI based remote control systems, e.g., to control and draw on slides in a presentation [Signer and Norrie, 2007b]. This concept has also been used in the commercial Oxford Papershow product[26]. In a similar approach, the program guide leaflet allows controlling the television program [Berglund et al., 2006]. Here the user manipulates the television program by check-marking interesting regions in the leaflet.

pRemote, [Hess et al., 2008], extends the notion of a paper based remote control to a universal personalized remote control system. Here, the user configures digital functions needed by drawing the layout of the remote. Thereby, the system attaches digital functionality to drawings using an end-user service composition approach [Borggräfe et al., 2008]. This paper based approach to end-user programing increases user satisfaction [Hess et al., 2011].

[25] http://www.skype.com (accessed: July 2015)

[26] http://www.papershow.com/en/ (accessed: July 2015)

Weibel et al. suggest to use such a PPI based remote control approach to interact with large scale displays: as the user cannot easily reach many parts of the screen, configuration takes place on a paper placeholder card in the *Hiperpaper* system, [Weibel et al., 2010c, Weibel et al., 2010b]. In a similar fashion, *Vodoosketch*, [Block et al., 2008], explored a tool palette printed on paper to conveniently interact with large vertical surfaces, e.g., drawing boards.

Despite remote controls offering an interesting mobile use case, these approaches target only stationary settings.

Command and Control Centers

PPI also serves as an important modality in command and control centers. Applications thereby range from air traffic control [Vinot et al., 2014] over military applications [McGee et al., 2002] to collaborative table top systems for first responders in emergency situations [Doeweling et al., 2013]. In this context, Cohen and McGee introduced *Rasa*, [Cohen and McGee, 2004], a prototypical system supporting military officers in the field by offering multimodal input on paper artifacts. This enables them to orchestrate military operations while using the convenient tangible properties of paper artifacts.

Letondal et al. followed a similar approach in *Strip'TIC*, [Letondal et al., 2013], a system designed to support air traffic control using the physical affordances and inherent support for co- located collaboration of paper artifacts. Here, however, paper artifacts are overlaid by digital information via projection and interactivity is increased by means of gaze tracking [Hurter et al., 2012]. Döweling et al. applied this approach to the domain of emergency first responders, where a wealth of heterogeneous information and differing media, e.g., paper notes and digital artifacts such as emails, had to be integrated in an intuitive and robust way to support interaction under stress [Doeweling et al., 2013].

Discussion

The applications discussed in this section demonstrate a plethora of additional application domains for PPI and mPPI. PPI naturally and conveniently supports creative tasks and co-located collaboration, as well as applications in therapy and interaction with large surfaces. However, current systems in these domains also mainly focus on supporting stationary settings. Only the domain of creative tasks poses a notable exception: examples of creative applications, e.g., the mobile music composition assistant, [Tsandilas, 2012], strikingly demonstrate the potential of hybrid mPPI ensembles. However, these approaches lack insights toward design of enabling infrastructures, conceptual frameworks and interaction theory.

2.3 Infrastructures for Mobile Pen-and-Paper Interaction

As shown in section 2.2, numerous application domains for PPI and mPPI exist. Applications leverage digital pen technology to provide natural and convenient interaction with digital systems. Although initial approaches addressing mobile settings demonstrate the huge potential of mPPI, these approaches remain particularly underrepresented. Especially, support for mobile usage practices associated with pen and paper, i.e., user mobility and document mobility, remains scarce. Where related work partially supports these practices, solutions for technical challenges only exist at the application level and fail to offer insights into mPPI support at the infrastructure level.

Digital pen technology alone does not suffice to enable the use of PPI. An application needs to access digital pen hardware in order to obtain recorded movement data. Then it needs to process this data further and relate it to its interactive regions. This is the role of the PPI *infrastructure*. It provides the environment for PPI based applications to leverage paper and pen as an input medium. At the same time, it supports application developers by providing useful abstractions and taking over required processing tasks. Herein we follow the definition of Edwards et al. in constraining the view on infrastructure to *"[. . .] software providing functions, capabilities, or services to other software"* [Edwards et al., 2010].

This section reviews existing infrastructure approaches for PPI with respect to support for mobile settings, i.e. mPPI. First, it derives a generic functional decomposition of required, logical infrastructure components and a set of concrete requirements which infrastructures need to satisfy in order to support mPPI. Then, it classifies and subsequently analyzes existing infrastructures with respect to these requirements. Thereby, it demonstrates why these approaches fail to provide full support of mobile usage characteristics of pen and paper.

2.3.1 Infrastructures: Functional Decomposition and Requirements

As described above, PPI (and hence also mPPI) infrastructures serve multiple goals. These goals allow a functional decomposition of infrastructures into coarse-grained logical components. Such a decomposition facilitates the analysis of existing PPI infrastructures, as different aspects of existing systems can be discussed individually. The analysis with respect to support for mPPI, however, bases on a set of requirements which infrastructures need to satisfy in order to support mobile usage practices associated with pen and paper, i.e., mPPI. This section provides the foundation for a subsequent analysis and discussion of existing PPI infrastructures by developing the functional decomposition and a deriving set of requirements for supporting mPPI.

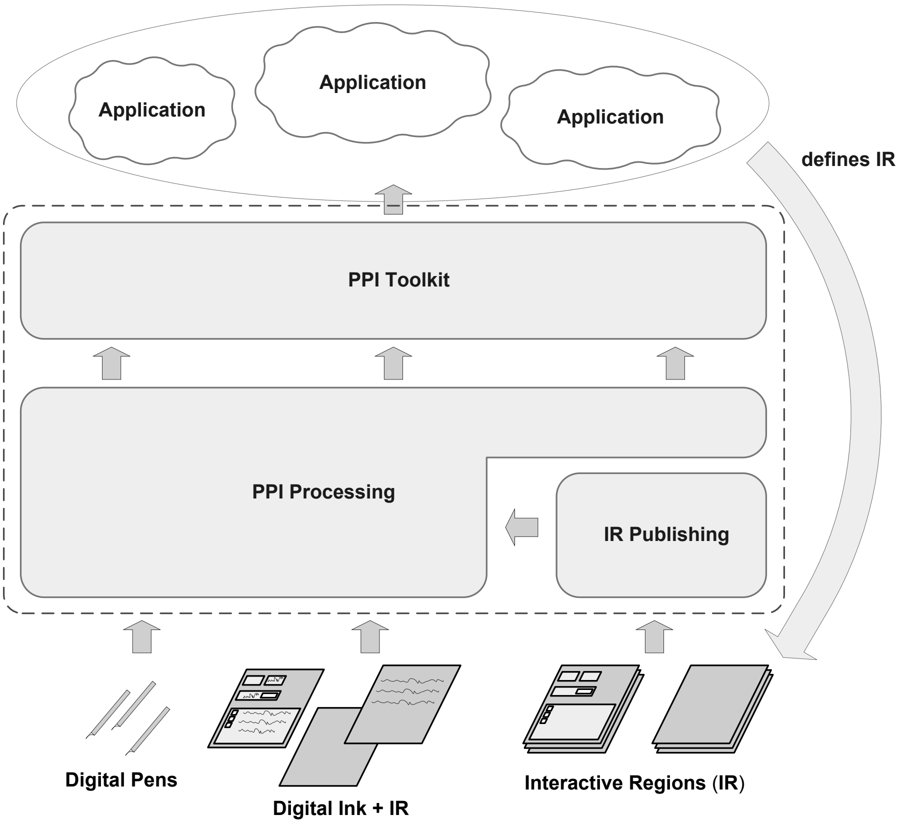


Figure 2.5: Functional Decomposition of PPI Infrastructure (IR: Interactive Region)

Functional Decomposition

A fundamental task of any PPI infrastructure is processing of PPI itself. Thus, the *PPI Processing* component serves as central runtime environment to connect and access digital pen hardware as well as process recorded data, i.e., digital ink. Thereby it associates digital ink with interactive regions defined by the application. This logical component is the core component required by all PPI infrastructures as it offers essential functionality.

However, relating digital ink to interactive regions, short *IR*, requires an *IR Publishing* component. This component manages knowledge on defined interactive regions, e.g., knows which interactive region belongs to which application. The scale of IR publishing that this component supports can thereby range from a single hard-coded interactive region (trivial IR publishing) up to managing billions of interactive regions, e.g., paper documents and screen surfaces, on a scale comparable to the Internet. In the latter case, this component takes on the role of an interactive region naming system comparable to DNS [Guimbretière, 2003].

Finally, developing applications by directly using raw digital ink data is not convenient: toolkits exposing more powerful abstractions to the developer are needed [Yeh et al., 2008]. Thereby, the *PPI toolkit* provides abstractions and re-usable functionality to the application developer comparable to GUI toolkits, e.g., Java Swing[27]. Here, the underlying conceptual framework of interaction (c.f. section 2.4) influences toolkit design and therefore determines the abstractions provided to the developer. However, in contrast to the logical components described above, the PPI toolkit can be considered as an optional component: it is perfectly possible to support PPI and mPPI without it, however, it considerably eases the development of PPI or mPPI based applications.

Fig. 2.5 depicts the three resulting structural components of PPI infrastructure. As depicted, applications using PPI define interactive regions, e.g., forms consisting of interactive region hierarchies or empty pages supporting free-form writing and drawing. The IR publishing component provides knowledge about interactive regions defined by applications to the PPI processing component. The PPI processing component, provides access to digital pen hardware and processes digital ink; thereby relating it to interactive regions. Finally, the PPI toolkit provides abstractions easing the task of application developers.

Infrastructural Requirements

Edwards et al. showed that infrastructure design plays a crucial role in human computer interaction [Edwards et al., 2010]: badly designed infrastructure may constrain user experience design alternatives and propagate undesirable conceptual abstractions to the user interface. Infrastructure therefore needs to support important user related characteristics of the domain and provide adequate abstractions. In the context of mPPI it is therefore imperative to provide full support for mobile practices associated with paper, specifically to support *user mobility* and *document mobility* (c.f., chapter 1, section 1.1.3).

R1: User Mobility. User mobility describes usage of paper in mobile or nomadic settings, e.g., writing notes into a brought along paper notebook during a train ride. Users need to be able to start working with a brought-along sheet of paper immediately in varying settings. Paper supports this form of mobility out of the box: its convenient and flexible form factor facilitates carrying, and its instant-on function-

[27] http://docs.oracle.com/javase/7/docs/technotes/guides/swing/index.html (accessed: July 2015)

ality and robustness allow for using paper documents in a plethora of environments, e.g., during field trips [Yeh et al., 2006a].

However, for the PPI infrastructure, supporting user mobility means it has to cope with a continuously changing environment in terms of available devices and services, e.g., due to loss of connection to backend services. At the same time, the infrastructure has to deal with resource-constraint mobile devices. Although this limitation seemingly has become less relevant due to increased processing power of mobile devices, certain digital ink processing tasks, e.g., handwriting recognition, still rely on external processing where possible, e.g., to save battery.

Facing these challenges, the mPPI infrastructure has to support the full potential of PPI while on the move. This results in the following requirements:

R1.1: Interactive Mode The infrastructure needs to support the interactive operation mode in mobile settings in order to allow for truly interactive use. Several mPPI based applications supporting user mobility focus on the batched operation mode only, e.g. ButterflyNet [Yeh et al., 2006a]. This severely limits the PPI design space, as there is no possibility of giving direct feedback to actions carried out with the pen on a mobile device. In order to enable the use of more sophisticated interaction techniques, e.g., by giving real-time feedback to the user, support for the *interactive mode* in mobile settings is required.

R1.2: Mobile Platform Using the digital pen in combination with a mobile device to employ PPI in mobile settings requires at least partial processing of recorded pen data on the mobile device itself. Especially the connection to digital pen hardware needs to be handled on the *mobile platform*. Therefore the infrastructure needs to provide executable components or services for common mobile platforms, e.g., Android or iOS. These components must be prepared to blend into the application model of the target platform.

R1.3: Flexible Deployment Continuously changing environments and the presence of resource constrained devices require the infrastructure to flexibly adapt to a broad variety of deployment configurations. Especially, the decision, which part of the processing needs to be performed on which device, e.g., whether processing occurs on the mobile device or in a backend service, should be open to the developer. Ideally, the deployment can be altered at runtime in order to allow for adaptive processing of digital ink. This is here referred to as support for *flexible deployment*. Depending on the usage scenario, the infrastructure needs to adapt PPI processing to available devices and services. This also requires a plug and play like interoperability between infrastructure components.

R2: Document Mobility. Document mobility refers to paper artifacts being mobile, either by being distributed or used in multiple application contexts. This includes the distribution of documents, i.e., the act of passing documents to others (on the distributing end) and encountering documents that have been passed by others (on the receiving end). Thereby, paper documents are passed on by individuals and organizations *intra-organizational*, e.g., a workflow document is handed from one department in a company to another, as well as *inter-organizational*, e.g., a leaflet is distributed from a retailing company to customer households.

For the PPI infrastructure supporting this mobility scheme, this means that it constantly encounters new, unknown documents. At the same time, mPPI based applications connected to these documents somehow ”travel” with the documents, i.e., the infrastructure can relate a newly encountered document to one or more mPPI based applications and channel digital ink accordingly. Additionally, documents become mobile in the sense that their contents refer to different contexts, e.g., a telephone number noted at the side of a scientific paper print-out. From the viewpoint of the PPI infrastructure this means that two different applications share the same document, or more precisely, the same interactive regions.

This yields the following requirements for mPPI infrastructure supporting document mobility:

R2.1: Resource Sharing The constant flux of available documents and connected applications, as well as the usage of documents in different contexts, imply multiple heterogeneous mPPI based applications being used simultaneously. Thereby users switch back and forth between applications as required by their current task. However, the resources used to interact with applications will remain the same, e.g., the same digital pen is used to write down notes and to annotate print-outs.

This also affects paper documents (more precise: interactive regions), e.g., the technical drawing sheet can also be used scribble some quick notes for a note-taking application. Here the PPI infrastructure needs to support sharing of resources between different applications. However, resource sharing not only applies to hardware resources: ideally, software components are shared between applications as well, to replicate as few as possible processing tasks.

R2.2: IR Discovery Interacting with encountered documents, as described above, means that the infrastructure knows the interactive regions defined by all applications in order to dispatch relevant data to an interested application. However, local storage of all known interactive regions in a system supporting document mobility cannot be achieved: there must be a way to map acquired pen data to previously unknown interactive regions.

This in turn requires a flexible distribution of interactive regions which is not problematic as long as a central control over the definition of interactive regions exists, e.g., in case a single organization issues all documents and applications. In reality, however, a multitude of heterogeneous organizations issue documents and applications.

Additionally, the same documents might be used by multiple applications, some of them only temporary. Here the infrastructure needs a highly scalable and distributed naming service, an *Interactive Region Lookup System* (IRNS), to identify and inform responsible applications or services. Such a system can be compared to *AutoId*[28] or the *Domain Name System* (DNS) for interactive regions [Guimbretière, 2003, Signer and Norrie, 2010].

2.3.2 Existing Infrastructures for Pen-and-Paper Interaction

Despite introduction of numerous PPI and mPPI based applications as demonstrated throughout section 2.2, only a limited amount of infrastructures supporting PPI/mPPI has been proposed. Several commercial solutions provide software development kits (SDK) supporting PPI processing and application development, as well as interactive region lookup in a limited scope. In addition to this, some infrastructure approaches have been introduced in the literature. Most of these focus on particular infrastructure components instead of the complete infrastructure.

Commercial Solutions

Currently, several commercial SDKs are available supporting the development of PPI based applications, mostly targeting the stationary domain.

Anoto. The digital pen developer Anoto itself provides a set of development and design tools [29] for its digital pens, here referred to as the *Anoto SDK*. The *Anoto SDK* mainly focuses on form processing. It includes a driver-like component for PPI processing, a server for locating pre-defined interactive regions (IR publishing) and a toolkit aiding the design of user interfaces on paper documents (as an Adobe Acrobat plugin). Thereby, it uses a document centered application model: an application corresponds to a document in which its interactive regions are defined.

The *Anoto SDK* supports both, the interactive and the batched operation mode of digital pens (R1.1), however, only in stationary settings, i.e., via a driver to the Microsoft Windows operating system for desktop PCs. Mobile platforms (R1.2) are not

[28] http://www.autoidlabs.org (accessed: July 2015)

[29] http://www.anoto.com/creative/technology-licensing/tool-kit/ (accessed: July 2015)

supported out of the box. With respect to PPI processing, the *Anoto SDK* employs a monolithic approach, i.e., a deployment of a all dedicated infrastructure packaged with the application, without support for flexible deployments (R1.3) or resource sharing (R2.1). It provides a desktop PC (Microsoft Windows platform) and web-based version. Dynamic interactive region discovery uses a similar client-server approach as *PADD*, [Guimbretière, 2003], (see next section), which does support only small scale deployments, e.g., supporting several different forms within an organization. Furthermore, due to the lack of dynamic allocation of interactive regions (interactive regions need to be deployed to the server), this approach toward IR publishing prevents sharing the same paper artifacts between applications as required in order to support document mobility (R2.1).

LiveScribe. As an officially licensed Anoto partner, *LiveScribe*[30], initially offered an *SDK* to support development of PPI based applications for their *Echo* and *Pulse* pen models, however, as of 29th of July 2011 this *SDK* has been officially withdrawn from the public[31]. Prior to that, the *LiveScribe SDK* had supported PPI processing to some extent and provided a simple toolkit to aid PPUI development. However, its application model differed considerably from other approaches, as applications were deployed on the pen itself [Schreiner, 2008].

Real-time wireless communication with another system, as needed for the interactive operation mode, is not supported at all (R1.1). On the one hand, deploying applications directly on the pen provides a mobile platform (R1.2), although interactive capabilities of the LiveScribe pens are severely restricted. It also allows using the same digital pen in different applications, i.e., the sharing of some of the interaction resources (R2.1). On the other hand, all interactive regions need to be defined at deployment time and stored on the digital pen without support for dynamic interactive region discovery (R2.2). This also limits the sharing of interactive regions among different application contexts (R2.1). Furthermore, the lack of real-time communication And execution of processing only on the digital pen also prevents flexible deployments (R1.3).

Based on the LiveScribe infrastructure and application model, Piper et al. introduced an end user development toolkit, which enables end users to quickly create small PPI based applications by drawing [Piper et al., 2011, Piper et al., 2012a]. However, this toolkit inherits the aforementioned restrictions of the LiveScribe platform. Interestingly, it is not aimed at programmers or software developers, but rather at the actual end users [Piper et al., 2012b]. As such it offers an interesting, although severely restricted approach toward development of PPI based systems.

[30] http://www.livescribe.com/ (accessed: July 2015)

[31] http://www.livescribe.com/errors/developer.html (accessed: July 2015)

As official successor of the former *LiveScribe SDK*, the company now offers a private SDK for its novel *LiveScribe Smartpen 3* for partner companies aiming to develop applications for mobile devices. In contrast to the original SDK, this novel approach introduced in autumn 2013 specifically targets mobile platforms[32]. Here LiveScribe deviates from their application model of deploying the complete application on the digital pen as applications are fully deployed on the mobile device[33] (R1.2). In this novel approach, the pen continuously streams data to the mobile device where it is further processed. Thus it supports the interactive mode in mobile settings (R1.1). However, it does not address the issue of flexible deployments (R1.3): everything has to be processed on the mobile device exclusively. At the same time, it does only allow to employ pre-configured paper artifacts, i.e., the LiveScribe notebooks. As a result, interactive region discovery (R2.2) is not supported. In addition, sharing interactive regions among applications, or sharing the same interaction resources, is not supported (R2.1).

MIL SDK. The MIL-Anoto Mouse Driver[34], developed at the Media Interaction Lab of the University of Applied Sciences Upper Austria in Hagenberg, provides a PPI processing infrastructure component supporting digital pens on interactive whiteboards and tabletop computers [Brandl et al., 2007]. It supports basic PPI processing targeted at the interactive operation mode of the digital pen (R1.1). However, it supports stationary, i.e., non-mobile, use only. Mobile platforms (R1.2) and flexible deployments (R1.3) therefore are not included. However, it allows the simultaneous use of multiple digital pens and hence partially supports resource sharing (R2.1). IR discovery mechanisms are not included (R2.2). It bases on a limited set of pre-defined, static interactive regions.

PPI Infrastructure Research

Similarly to commercial applications, PPI infrastructure research mainly targets the stationary domain. Thereby, many approaches focus on identifying paper documents and subsequent document retrieval, i.e., the IR publishing component of infrastructures. However, other components have been introduced as well.

PADD. Guimbretière introduced *PADD* (which stands for *Paper Augmented Digital Documents*), [Guimbretière, 2003], one of the earliest infrastructure approaches for

[32] The author likes to stress that this SDK for mPPI was only introduced in 2013, *three years after* initial publication of the contributions presented in this thesis, e.g., [Heinrichs et al., 2010b] (which included public, open source availability of its reference implementation, c.f., chapter 3).

[33] http://www.livescribe.com/en/smartpen/ls3/features.html (accessed: July 2015)

[34] http://mi-lab.org/products/ (accessed: July 2015)

PPI. It targets stationary settings, e.g., users working with documents at the desktop. Essentially, *PADD* revolves around the life- cycle for hybrid paper-digital documents consisting of digital editing, printing and document editing in the physical domain. Therefore, this approach includes support for PPI processing and interactive region publishing, but no PPI toolkit. Furthermore, its application model strictly revolves around the document paradigm, where functionality is associated with documents (either in a digital or physical representation). Other Interactive regions, e.g., digital screens supporting pen-based interaction in addition to paper documents, are not considered.

Initially, *PADD* was designed for using the batched interaction mode only. However, *PapierCraft*, [Liao et al., 2005, Liao et al., 2008], presents a gesture system on top of *PADD* that allows interacting with documents using the interactive mode (R1.1). *PADD* does not support mobile platforms (R1.2). Furthermore, its monolithic client-side PPI processing approach does not allow flexible deployments of different parts of PPI processing (R1.3) and the sharing of interaction resources (R2.1). However, *PADD* offers initial contributions toward IR discovery (R2.2): In *PADD* a central server, the *paper look-up service*, allows identifying documents, as soon as digital ink on a document is recored by the system [Guimbretière, 2003]. The data itself is processed locally and stored in a shared database associated with the particular document. This centralized architecture, although allowing for dynamic interactive region discovery in a small scale, severely limits the scalability of the approach. Additionally, it hinders the use of interactive regions in different contexts (e.g., as required to support resource sharing, R2.1), as only a single, shared view on any interactive region exists.

PaperToolkit. The freely available Stanford *PaperToolkit*, [Yeh et al., 2007], is another infrastructural approach for supporting PPI. It aims to provide a convenient toolkit aiding developers in the design of *Pen- and-Paper User Interfaces* (PPUIs, as defined on page 2) [Yeh et al., 2008]; it also features PPI processing capabilities. Dynamic IR publishing is not included; the platform enables application developers to define interactive regions at deployment time. These IRs relate to user interface components in a similar way to widgets in graphical user interfaces.

The *PaperToolkit* supports both, the batched and the interactive operation mode of digital pens (R1.1). However, it exclusively targets stationary settings and hence does neither support mobile platforms (R1.2) nor the flexible run-time deployment required to adapt to dynamically changing environments (R1.3): in the paper toolkit, all components are deployed on a single machine and packaged together with the application or as set of applications. The *PaperToolkit* also lacks support for dynamic discovery of interactive regions (R2.2) and the ability to share resources between different applications (R2.1), as required to support document mobility. Interestingly, its toolkit component introduces an event based architecture comparable to typical graphical user interface toolkits, e.g., Java Swing. This indirectly transfers the conceptual framework of traditional graphical user interfaces, e.g., point and click interactions, to the domain of PPI.

iServer and iPaper. The *iServer/iPaper* framework developed at ETH Zürich forms a comprehensive infrastructure supporting PPI. The main component of this framework consists of *iServer* [Norrie and Signer, 2005], a general purpose cross-media linking system [Norrie and Signer, 2003, Norrie et al., 2005]. Its architecture bases on the generic *Resource-Selector-Link* (RSL) model for cross-media linking systems [Signer and Norrie, 2007a] (see also sections 2.4 and 2.5). In RSL, *resources*, e.g., interactive paper documents or user interface components, can be linked by means of *selectors* and *locators*. Thereby, *selectors* allow specifying which part of a resource, e.g., a paragraph or even a word in a paper document, is addressed in or respectively by the cross-media link. The *locator* then specifies, how the system can resolve and access the resource.

Among other link types, *iServer* supports links between physical paper documents and digital documents via the *iPaper* plugin, [Norrie et al., 2006a], e.g., mapping paper print-outs directly to digital documents [Norrie et al., 2006b]. Actions on these documents, e.g., editing or annotations, can then be mapped back to the source documents enabling PPI in applications [Norrie et al., 2005, Signer and Norrie, 2011]. Extensions to the *iServer/iPaper* framework further add support for gesture recognition, e.g., *iGesture*, [Signer et al., 2007b], and digital ink segmentation, [Ispas et al., 2011]. Furthermore, *iServer* supports sharing of links over institutional boundaries via a special plugin [Signer et al., 2009]. This enables the *iPaper* plugin to publish interactive regions. In order to support authoring of paper documents, Signer et al. also introduced an authoring environment based on *iServer* and *iPaper* [Signer et al., 2014]. It essentially allows defining interactive regions on paper documents binding them to interactive functionality and is targeted at end users.

In summary, the *iServer/iPaper* infrastructure consists of components for PPI processing, an application model and IR publishing approach based on the RSL model, as well as authoring environments aiding the development of PPI based applications (comparable to toolkits).

iServer/iPaper supports the interactive and the batched operation modes (R1.1). Although initially designed for desktop use only, custom-built extensions to the framework support use of PPI on mobile platforms (R1.2). This has been demonstrated in *edFest*, [Signer et al., 2006, Signer et al., 2007a, Norrie et al., 2007] and *Ubisketch*, [Cowan et al., 2011, Weibel et al., 2011b]. Here, the actual processing of digital ink

Requirement		Anoto SDK	LiveScribe / SP3 SDK	MIL SDK	PADD + PapierCraft	PaperToolkit	iServer + iPaper
User Mobility	R1.1 Interactive Mode	x	- / x	x	x	x	x
	R1.2 Mobile Platform	-	(x) / x	-	-	-	(x)
	R1.3 Flex. Deployment	-	- / -	-	-	-	-
Document Mobility	R2.1 Resource Sharing	-	(x) / -	(x)	-	-	(x)
	R2.2 IR Discovery	(x)	- / -	-	(x)	-	x

Table 2.1: Mobile PPI Support in Existing Infrastructures

occurs on a backend instance of *iServer*; the mobile device only captures digital ink and transfers it to the backend. However, in general the *iServer/iPaper* framework does not support flexible deployments of PPI processing components (R1.3): processing normally occurs exclusively on the client side and cannot be distributed[35].

iServer/iPaper also does not support the sharing of resources between instances of the framework out of the box (R2.1). Especially, paper documents can only be used by a single application, as links define the handling entities via the selector concept. To cope with the problem of interactive region discovery, *iServer/iPaper* introduces a distributed hierarchical naming system for paper applications, the *Universal Interactive Paper Look-up Service* [Weibel, 2009]. This service can locate the responsible server for an encountered paper document (R2.2).

2.3.3 Deficits in Existing Infrastructures

Currently available infrastructures for PPI or mPPI do not support sharing of resources to it's full extent (R2.1). For instance, neither commercial approaches, nor *PADD*, [Guimbretière, 2003], or the *PaperToolkit*, [Yeh et al., 2008], allow simultaneously using the same pen in multiple applications. In approaches that do support sharing digital pen resources, e.g., *iServer/iPaper*, [Signer and Norrie, 2007b], using the same document in multiple contexts is not possible. This is due to their basic architecture employing a monolithic client-side processing component typically deployed together

[35] A special proxy handles this in the mobile systems described above. This further underlines the need for flexible deployments in mPPI infrastructures, as this would obsolete the need for such a proxy.

with the actual application. This ”locks” resources to this particular configuration alone; other applications cannot access the interaction resources anymore.

Some approaches back this monolithic, client-side deployment configuration of PPI infrastructure using a pre-configured central document server hosting digital document instances, e.g., *PADD*, [Guimbretière, 2003], or offer a more open approach to generic links between paper resources and digital documents, e.g., *iServer/iPaper*, [Signer and Norrie, 2007b]. Their aim is thereby to facilitate IR discovery (R2.2). However, these servers are not designed for dynamically coping with new and beforehand unknown applications and associated interactive regions. Basically, the design of existing PPI infrastructures bases on the assumption that all parts of the system (pen, paper artifacts, computer hardware, application software components) are deployed and controlled by a single authority.

Furthermore, dynamic and flexible deployment of the system and required components (R1.3) to aid processing in the mobile domain, e.g., in a plug and play manner, is neither supported nor facilitated by available infrastructures. Deployment on mobile platforms (R1.2), with the exception of the novel *LiveScribe Smartpen 3* platform and its SDK, are also not supported. Interactive mobile use (R1.1) is in most cases limited to fixed scenarios, where the infrastructure is specifically tailored for prototypical applications. However, introducing user mobility and document mobility at the application level is problematic. Such applications act as proxies to existing non-mobile infrastructures, e.g. in [Weibel et al., 2010a], and thus replicate necessary components. System support for mobile PPI needs to address mobile characteristics of paper at the infrastructure level in order to allow developers exploiting its full potential without replicating required parts.

In summary, current infrastructures do not support important mobile characteristics of ”real” pen and paper, i.e., *document mobility* and *user mobility*, to full-extent as shown in table 2.1. These restrictions obviate PPI based applications to appropriately target the mobile domain, ultimately leading to unsatisfying solutions (c.f. section 2.2.1). Particularly, the lack of infrastructures supporting the mobile domain has forced researchers to build prototypes lacking key characteristics of the systems to be explored, e.g., S-Notebook reports on cross-media links between notes and digital artifacts for the mobile domain, however, its architecture requires the digital pen to be connected to a stationary server [Pietrzak et al., 2012]. Similarly, Tsandilas introduces interaction techniques allowing to improve mobile digital ink recognition, while the employed prototype relies on a stationary setup [Tsandilas, 2012].

This raises demand for an infrastructure specifically designed to support the use of mPPi, i.e., PPI in mobile settings, while preserving all mobile characteristics of real paper.

2.4 Conceptual Frameworks of (Mobile) Pen-and-Paper Interaction

Actively designing interaction between users and digital systems plays a central role in the development of usable interactive systems [Beaudouin-Lafon, 2004]. Thereby, toolkits, as a logical component of PPI and mPPI infrastructures (c.f., section 2.3.1), are the connecting element between interaction design and infrastructures. Toolkits provide convenient abstractions aiding application development [Yeh et al., 2008]. Furthermore, they facilitate exploring interaction design alternatives as part of the user centered design process, e.g., as described by Rogers et al. [Rogers et al., 2011], without requiring a too early focus on technical peculiarities (which would limit the set of alternatives to be explored).

However, at the same time toolkits determine the design space available to developers, the design *vocabulary*. In this context, it becomes important to reflect upon the underlying principles of how interaction is described and modeled within such a toolkit, as this ultimately determines abstractions available as part of the toolkit. These principles are here referred to as the *Conceptual Framework* of interaction.

Most existing approaches, e.g., the *PaperToolkit*, [Yeh et al., 2007], base toolkit design on the principles of traditional GUI toolkits. This implicitly assumes a conceptual framework for *WIMP* (Windows Icons Menus Pointers, c.f., [Hinckley, 2007]) interaction, as found on desktop computers employing keyboard and mouse as main interaction devices. This results in *Pen-and-Paper User Interfaces* (PPUIs, c.f., definition on page 2) which closely resemble GUIs, where the pen is reduced to the role of a computer mouse, i.e., a mere pointing device.

However, interaction techniques in PPUIs differ from those employed in traditional GUIs. For example, unlike a mouse, the pen (normally) leaves an ink trail on paper. Using interaction techniques that require the user to mark the same paper area twice will render the content on the paper unreadable. Thus, novel interaction techniques exist for PPI based applications. With toolkits based on conceptual frameworks of other domains, e.g., WIMP, designers targeting interaction techniques specifically designed for PPUIs might end up implementing these on the application level after all. This considerably limits the convenience offered by the toolkit, ultimately reducing its utility.

In order to provide genuine toolkits for PPI or mPPI, the toolkit design has to be based on a conceptual framework of PPI. This conceptual framework needs to answer the question: how can pen-and-paper interaction techniques be described? How can they be mapped in the actual system? Here, the conceptual framework should provide the designer with a precise vocabulary during interaction design and enable formal description of (mobile) PPI. This vocabulary provides a solid foundation for PPI/mPPI toolkits as it defines their basic building blocks. Furthermore, it aids structured design space exploration.

This section reviews and classifies existing interaction techniques for PPI and mPPI. It also briefly discusses respective theoretical implications toward mobile PPI for different classes of techniques (c.f., section 2.5). Based on this, it derives a set of requirements that allow conceptual frameworks of (mobile) PPI serving as foundation for toolkits. It then analyzes existing conceptual frameworks and discusses their limitations regarding expressive power and application to toolkit design.

2.4.1 Pen-and-Paper Interaction Techniques

As defined by Hinckley, an *interaction technique* consists of *input* combined with appropriate *feedback* [Hinckley, 2007]. It essentially represents a mechanism employed by a user to invoke certain functionality, e.g., the drag and drop technique in GUIs to open a particular file in an application. Thereby, the interaction technique refers to the mechanism *how* a certain functionality can be invoked.

Functionality invoked can either be self-consistent or apply to data specified as part of the interaction technique, e.g., crop-marks to mark text used in a copy command [Liao et al., 2008]. In the latter case a selection technique forms part of the interaction technique, where the user employs another technique to specify data. This concept is referred to as *chunking and phrasing* or *chaining* [Buxton, 1986]. Here, chains of interaction techniques can be constructed to form a complex technique out of several basic building blocks.

In the domain of PPI, three main classes of interaction techniques have been proposed: *Pidgets and Proxies* on one end of the spectrum, *Gesture Systems* on the other, and *Cross-media Links* supporting tight integration of paper and digital artifacts.

Pidgets and Proxies

The first class of pen-and-paper interaction techniques relies on attaching digital functionality to certain regions on paper. Whenever the user positions the pen on such a region, the system executes digital functionality associated with this region. Iconographic representations of this functionality help visualizing the concept to the user. This mechanism can be compared to clicking a button in a GUI, as depicted in Fig. 2.6. Anoto coined the term *Pidget* interaction to describe such techniques in the documentation of their paper SDK [36]. Thereby, a *Pidget* corresponds to an interactor represented by an icon printed on paper that triggers system functionality when ”clicked” on with a digital pen [Signer et al., 2014]. This technique is often combined with selection techniques, e.g., by drawing a line around a document area,

[36] http://www.anoto.com/creative/technology-licensing/tool-kit/ (accessed: July 2015)

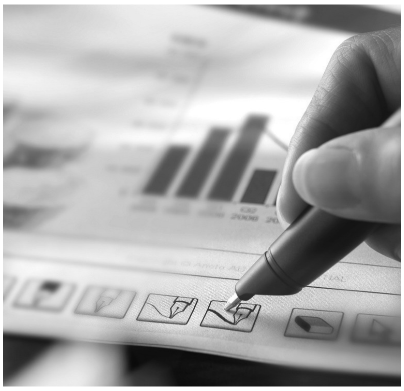


Figure 2.6: The Pidget Interaction Technique (courtesy Anoto AB)

to specify input to the command invoked when subsequently ”clicking” a Pidget [Costa-Cunha and Mackay, 2003].

Many PPI based systems use Pidgets, as they provide a very intuitive approach toward PPI both from the developer as well as from the user perspective. Examples are *PaperPoint*, [Signer and Norrie, 2007b], which employs the concept as the main paradigm to control slides in a presentation; and *NICEBook*, [Brandl et al., 2010], which categorizes contents of notes via Pidgets (similarly to [Steimle et al., 2008a]). Furthermore, Pidgets provide a very convenient way to support palettes in structured diagramming applications, e.g., in [Dachselt et al., 2008], as well as for remote control applications based on PPI, e.g., in [Berglund et al., 2006].

The concept of attaching functionality to a certain location makes complex segmentation operations, i.e., determining which digital ink refers to created content and which refers to control commands, completely obsolete. However, the main disadvantage of Pidgets is their static nature: a region on paper has to be attached to functionality during design time and the graphic representation has to be printed on paper documents. Encountered paper documents cannot be instantly used by such applications, as they would lack the printed representations. Additionally, using Pidgets renders re-use of paper documents in other contexts problematic at best (c.f., document mobility as described in section 2.3.1).

Furthermore, the amount of paper real estate dedicated to representing Pidgets as opposed to the real estate dedicated to contain user generated content, needs to be carefully balanced. This introduces an upper limit for the amount of Pidgets, which might exceed the requirements of a given application.

Gesture Systems

The second class of pen-and-paper interaction techniques bases on associating functionality with gestures. Whenever the user performs a gesture with a digital pen, the system processes and interprets the digital ink, subsequently mapping it to an alphabet of pre-defined gestures. This process is referred to as *gesture recognition*. If the system recognizes a known gesture, it triggers functionality associated with that gesture. Thereby, chaining and combining gestures is possible, e.g., combining selection techniques with gestures to specify input data as in *PapierCraft*, [Liao et al., 2008].

The actual recognition of gestures is a problem not specific to PPI or mPPI. A broad set of recognition algorithms exists. Recognition algorithms such as *Hidden Markov Models* (HMM), [Sezgin and Davis, 2005], *Dynamic Time Warping* (DTW), [Choe et al., 2010], and feature based statistical classifiers, e.g., the *Rubine classifier*, [Rubine, 1991, Blagojevic et al., 2010], have been used to recognize PPI gestures. However, simple geometric techniques often satisfy the need for fast and accurate gesture recognition, while imposing less complexity into the underlying recognition system. Examples for this class of algorithms are the famous *$1 Gesture Recognizer*, [Wobbrock et al., 2007, Anthony and Wobbrock, 2010], and the *¢1 Recognizer* [Herold and Stahovich, 2012].

In order to support gesture recognition on the system side, Signer et al. introduced *iGesture*, [Signer et al., 2007b], a flexible gesture recognition toolkit on top of the *iServer/iPaper* infrastructure. It offers several standard recognition algorithms in the domain of PPI to chose from. Other algorithms can be integrated by developers if needed.

Gestures have been used in a broad range of PPI based applications, e.g., *PapierCraft*, [Liao et al., 2005, Liao et al., 2008], a gesture system to manipulate documents, and *PaperProof*, [Weibel et al., 2008], a hybrid paper digital proof-reading system for scientific publications. Special gestures are used to mark certain regions for later use, e.g., in the "hotspot association" gesture described by Yeh et al. [Yeh et al., 2006a], where the user draws a set of crop-marks to act as a placeholder for a digital image. Gestures can also control mixed paper digital environments, e.g., in *Strip'TIC*, [Gauthier et al., 2014], where gestures are designed to span paper and virtual artifacts.

Gestures do not require any pre-printed interactors on documents. As a result, gesture systems support the use of documents in different application contexts, i.e., document mobility, considerably better than Pidget based techniques. However, a main problem of gesture systems is the discrimination between gestures and user generated content. Either complex segmentation techniques can be used, e.g. as proposed by Ao et al. [Ao et al., 2006] or Ispas et al. [Ispas et al., 2011]; or the user needs to explicitly define when a gesture starts, e.g., by pressing a button as in PapierCraft [Liao et al., 2008]. Furthermore, chaining of gestures requires methods to distinguish individual gestures in the chain, e.g., by introducing special markers [Hinckley et al., 2005, Hinckley et al., 2006].

Besides this additional design complexity, gesture based interaction also imposes problems regarding *learnability* and *recallability* [Norman and Nielsen, 2010]: the user needs to learn and remember gestures defined in the alphabet, because the interface itself typically does not offer any clues regarding available functionality. This can quickly become a problem in more complex systems. However, the actual performance of users depends on the design of the particular gesture set and the application itself, e.g., Liao and Guimbretièere showed that the gesture set employed in PapierCraft can be learned roughly within 30 minutes [Liao and Guimbretièere, 2012].

Cross-Media Links

The third class of interaction techniques combines actions on paper and digital artifacts in a single coherent cycle. In most techniques, it is thereby important where actions are carried out, e.g., the user needs to perform her actions on a designated paper area. Typically, these techniques are applied to connect paper and digital documents in order to establish *cross-media links*.

For instance, Steimle proposed a cross-media linking technique involving physical and digital artifacts in one cycle of actions [Steimle, 2009a]: in order to establish a link between a section of a paper document and a digital document, the user marks content by drawing a vertical line at a specially designated area on paper (link source) and another line on a digital document displayed on a screen (link target). After a link has been established in this manner, the markup, i.e., the line drawn on paper, acts as a Pidget and can be ”clicked” to activate the link, i.e., open the section of the digital document on the screen.

Similar concepts have been used in the context of links between multiple paper documents [Liao et al., 2008, Brandl et al., 2010]. These gestures typically base on the *stitching* concept, as originally described by Hinckley et al. [Hinckley et al., 2004]: the user lays paper documents and / or digital resources physically close together and draws a line spanning these resources to issue a link. Subsequently, the physical markup presented by the line can be used to follow the cross-media links. Thereby, actions in the digital world, e.g., pressing buttons as in [Liao et al., 2008], serve to initiate or confirm these links.

From an interaction point of view, cross-media linking techniques stand halfway between Pidgets and gestures. On the one hand, these techniques encompass behavioral components, i.e., the user needs to ”do something”. This resembles gestures in the fact that users have to learn, memorize and remember these components. On the other hand, these techniques typically involve a location component which enables applications to provide visual clues aiding recall, similarly to Pidgets. As such, cross-media linking techniques inherit advantages and disadvantages of both other classes to a certain degree.

Thereby, the digital component of cross-media links allows overcoming limitations inherited from Pidgets: Tsandilas and Mackay extended the concept of interaction proxies in the form of *knotty gestures* [Tsandilas and Mackay, 2010]. Here the user issues a gesture-like command (c.f., gesture systems as described above) which is dynamically bound to the region where this command has been drawn. This concept creates interactors similar to Pidgets without requiring a pre-configured and dedicated document as the regular Pidget technique. However, this approach still suffers the same penalties on learnability and recallability as gesture systems.

2.4.2 Conceptual Frameworks: Requirements for Toolkit Design

First and foremost, conceptual frameworks serving as the foundation for PPI and mPPI toolkit design have to support expressing the PPI / mPPI design space. In order to achieve this, such frameworks need to allow describing existing interaction techniques in the domain as discussed in section 2.4.1, i.e., the design vocabulary of the domain. Furthermore conceptual frameworks need to support extensibility with respect to novel interaction techniques. This in turn, requires conceptual frameworks to support basic concepts of forming interaction techniques in the domain, e.g., chaining and phrasing. Last but not least, a conceptual framework must offer a way for infrastructural components to ”understand” user interaction. This enables the infrastructure to recognize whether a certain technique was executed and what functionality to invoke (or feedback to give to the user).

This yields the following requirements toward conceptual frameworks of PPI and mPPI:

R3.1: Design Vocabulary Any conceptual framework must offer a basic design vocabulary of the domain. This forms basic building blocks offered to the interaction designer in order to describe interaction techniques and ultimately determines abstractions provided by toolkits. On the one hand, such a design vocabulary needs to reflect the basic nature of interaction techniques in general, as defined in section 2.4.1: invoking functionality through user actions, coupled with appropriate feedback. On the other hand, it needs to be tailored to the domain, i.e., allow expressing the existing design space and in particular the existing classes of interaction techniques. This includes the three basic categories of interaction techniques in the context of PPI and mPPI: Pidgets and Proxies, Gestures and Cross-Media Links (c.f., section 2.4.1).

R3.2: Composition A conceptual framework needs to support the composition of interaction techniques in order to enable more complex interaction techniques based on concepts such as chunking and phrasing, e.g., to specify input to subsequent commands as in the copy & paste interaction technique described in *PapierCraft*, [Liao et al., 2008]. Essentially, this allows interaction designers creating new interaction techniques based on existing building blocks, by forming expressions out of existing design vocabulary. Supporting such a concept in a conceptual framework requires to support composition of arbitrary (sub-) expressions.

R3.3: Openness and Extensibility In addition to basic design vocabulary, conceptual frameworks need to support adding new vocabulary to the framework. This is based on the fact that one cannot assume to cover all possible types of interaction techniques. Ultimately it depends on the creativity and acumen of researchers and interaction designers to explore new, promising interaction techniques. Any conceptual framework of interaction intended to serve as foundation for toolkit design therefore needs to embrace the fact that design vocabulary, however elaborately conceived, can never cover the entire design space. A conceptual framework therefore has to be designed for openness and extensibility.

R3.4: Machine Understandable For a conceptual framework of PPI or mPPI to be used as base of a toolkit, description of interaction alone does not suffice. Besides human readable description of interaction, it requires the infrastructure to ”understand” what user actions should be mapped to what functions and system responses: the infrastructure needs to be able to understand and act upon interaction techniques described through the conceptual framework; it requires the conceptual framework to be machine understandable.

2.4.3 Existing Conceptual Frameworks

Essentially, Beaudouin-Lafon described two levels for analysis and design of interaction between a digital system and a person [Beaudouin-Lafon, 2004]: *Interaction paradigms* provide a user centered high-level conception of the phenomenon of interaction. For instance, *Reality based interaction*, [Jacob et al., 2008], describes interaction at a high layer of abstraction using concepts of the physical world; essentially a specialized view on interaction between people and digital systems designed to conceptualize interaction techniques. In contrast to this, *interaction models* offer operational descriptions of the course of interaction. Here the interaction itself can be modeled and mapped to specific user actions and system responses. For instance, *instrumental interaction*, [Beaudouin-Lafon, 2000], or *direct manipulation*, [Shneiderman, 1983], describe the process of interaction between a user and a digital system and the concepts used to compose specific interaction techniques.

A conceptual framework for PPI and mPPI thereby couples these two aspects. It aims at describing interaction between a person and a digital system by means of pen and paper, while at the same time offering a formalized, operational description of the course of interaction, e.g., as required for toolkit design.

Traditionally, conceptual frameworks and models of interaction range from low level, formal human processor models, e.g., *GOMS* or *KLM*, [Card et al., 2000], to abstract descriptions, e.g., *Reality-based interaction*, [Jacob et al., 2008]. As for PPI, interaction with pen, paper and a digital device forms a subset of tangible interaction for which conceptual frameworks of interaction exist, e.g., *TAC*, [Shaer et al., 2004]. Expressing PPI with these models, however, is cumbersome as primitives relevant for PPI, e.g., gesture and Pidgets, must be constructed out of generic primitives for tangible interaction in general. Hence its vocabulary does not completely fit the domain of PPI and does not lend itself to support PPI toolkit design without introducing additional complexity.

Therefore, in order to support toolkit design, as elaborated in section 2.4.2, a conceptual framework needs to fit the domain as closely as possible, e.g., to provide an adequate vocabulary (R3.1) while at the same time offering machine understandable execution semantics (R3.4). Thus, only conceptual frameworks for the domain of PPI / mPPI provide promising candidates.

In the domain of PPI, only two conceptual frameworks or interaction models have been described so far: *RSL*, [Signer and Norrie, 2007a], and Steimle's conceptual framework [Steimle, 2009a].

Resource-Selector-Locator (RSL)

The *Resource-Selector-Link* (RSL) model proposed by Signer and Norrie has been used to model PPI as theoretical underpinning of the *iServer* and *iPaper* framework, [Norrie et al., 2006a]. *RSL* essentially describes a hyper-document system allowing to link between various resources, both digital and physical. Thereby, it defines *links* between different types of *resources*, where *selectors* specify which particular part of a resource is the source or target of a given link [Signer and Norrie, 2007a]. In the context of PPI, paper artifacts are modeled as resources linking to digital functionality. Selectors specify which part of the paper document links to which functionality. Interaction is modeled thereby exclusively as the invocation of links, i.e., following a certain link triggers a system response.

RSL can be used to describe a broad range of different cross-media links and provides machine understandable expressions (R3.4). It also allows chaining of expressions, as links may refer to other links (R3.2). However, *RSL* does not explicitly model the interaction between user and system and does not offer an interaction vocabulary (R3.1). It focuses on the invocation of functionality alone, i.e., triggering system responses. It also does not support openness and extensibility on the conceptual framework level (R3.3)[37].

Furthermore, *RSL* models interaction through selectors specifying a particular part of a resource. This limits the range of interaction vocabulary that can be supported at a conceptual level to Pidget and Cross-media link classes; as those closely follow the hyperlink model and location on paper artifacts forms an essential part of interaction. More sophisticated interaction techniques cannot be expressed without extending the model, e.g., gesture based techniques or approaches involving interactive system responses require additional concepts.

To deal with these conceptual problems, the selector semantics have been extended by the authors to allow for defining more complex interaction techniques, e.g., in the *iGesture* system on top of *iPaper / iServer*, [Signer et al., 2007b]. This corresponds to strategies used in modern Web applications, where *JavaScript* frameworks and *AJAX* are used to work around the hyper-document system nature of the Web. As a result, however, it introduces additional complexity to the design of toolkits, while at the same time lacking concrete domain vocabulary for conceptual modeling.

Steimle's Conceptual Model

To aid the design of interaction techniques employed in PPUIs (c.f., definition on page 2), Steimle proposed a conceptual framework grounded on empirical research [Steimle, 2009a]. It focuses on answering the questions which interaction techniques are available and which are appropriate in the chosen setting of collaborative knowledge work (this aspect is discussed in section 2.5.2). Additionally it provides a structuring of the design space that can be used to describe and model PPI.

Steimle's framework consists of a syntactic layer of *core interactions* and a semantic layer of *conceptual activities*. Interaction techniques are combinations of core interactions to perform conceptual activities. Described core activities include inking, clicking, moving, altering shape, combining and associating paper artifacts. Conceptual activities are functionality offered by the system, e.g., annotating, linking or tagging. This relates to the triggering of functionality as used in the *RSL* model described above, but extends it by a domain vocabulary. Both, core interactions and conceptual activities, were derived by observing users in various collaborative knowledge work tasks.

This structuring covers part of the design space for PPI and provides a basic vocabulary aiding its exploration (R3.1). Chaining of interaction techniques is an integral part of the framework and thus supported at the conceptual level (R3.2). However,

[37] although of course resulting toolkits can be extended through various selectors, links and supported resources

Requirement		RSL	Steimle's Framework
Suitability for Toolkit	R3.1 Design Vocabulary	-	(x)
	R3.2 Composition	x	x
	R3.3 Openness and Extensibility	-	-
	R3.4 Machine Understandable	x	-

Table 2.2: Suitability of Conceptual Frameworks of PPI for Toolkit Design

the vocabulary is highly influenced by the domain of knowledge work, which section 2.2 shows to be only a sub portion of the PPI design space. For instance, how would gestures be expressed by the framework? The closest match is *inking*, which would also refer to annotations made on paper, or even selections. Therefore the presented vocabulary can be considered as specialized. Also, the framework does not support openness and extensibility, as core interactions and conceptual activities are fixed (R3.3) Furthermore, it lacks formal semantics required for a machine understandable representation that is required to enable it to serve as foundation for a toolkit (R3.4).

2.4.4 Deficits in Existing Conceptual Frameworks

Existing approaches do not offer a comprehensive design vocabulary to interaction designers allowing to express all existing classes of interaction techniques (R3.1), let alone offer extensibility to reflect novel classes of interaction techniques (R3.3). More gravely, table 2.2 shows that existing approaches offering a limited design vocabulary lack machine understandability (R3.4) and vice versa. However, both are required for a conceptual framework of PPI to serve as basis for toolkit (c.f., section 2.4.2).

RSL, [Signer and Norrie, 2007a], provides formal execution semantics and hence machine understandability based on the hyperlinking model. However, it does not provide a domain vocabulary for designing interaction techniques, as this can be mapped only using the selector concept. This makes it to generic to serve as a basis for toolkit design. In contrast to this, Steimle's theoretical framework for PPI, [Steimle, 2009a], provides a first valuable step towards this direction. It lacks, however, machine understandability. Hence it does not lend itself to serve as basis of toolkit design as it remains too vague; digital systems cannot easily map the described concepts. Furthermore, its offered vocabulary grounds on the domain of knowledge work and therefore introduces a conceptual limitation with respect to other domains. The provided vocabulary is limited to simple interaction chains formed out of the limited set of actions. Modeling simultaneous actions or usage of gestures is not possible without extending the model. This fact is further aggravated, as both existing conceptual frameworks lack openness and extensibility at the conceptual level (R3.3).

In summary, existing conceptual frameworks fail to address all requirements of toolkit design. The invocation of actions and composition (R3.2) has been modeled successfully and initial steps toward a design vocabulary exist (R3.1). However, this needs to be combined with machine understandability (R3.4). As such, existing conceptual frameworks might serve as inspiration for a conceptual framework suitable as basis of a PPI / mPPI toolkit; provided such a framework can meet the demand for openness and extensibility (R3.3).

2.5 Theories of Mobile Pen-and-Paper Interaction

Toward the goal of developing usable and engaging PPI based applications, the designer and developer needs more than just the infrastructure and methods to map interaction techniques in infrastructures. The question arises, how should the actual interaction with a digital system by means of digital pen and paper be designed? Numerous interaction techniques for PPI exist to choose from. Yet, which of those prove to be adequate given the mobile setting?

To answer these questions, interaction theories play a crucial role. Interaction theories allow interaction designers to understand why certain interaction techniques work where others fail. Here, theories allow balancing conflicting design choices in hybrid mPPI ensembles and to design interaction between user and digital system in an informed way, based on understanding the impact of choosing one setup over another. They provide a set of design considerations to interaction designers which allow for precisely targeting successful techniques. Thereby, theories aid answering the question regarding adequate interaction techniques for a specific scenario.

This section describes existing theories for PPI and related scenarios. Thereby it demonstrates why these do not suffice with respect to the design of mobile PPI. First, it analyzes related work on the use of individual components of hybrid mPPI ensembles with respect to mobile settings thereby highlighting theoretic insights reported. Then, it lays out related work directly aiming at theories in hybrid mPPI ensembles and related fields. Finally, it discusses the limitations of existing approaches and outlines the need for theoretical research in the domain.

2.5.1 Theoretical Impact of Hybrid mPPI Ensemble Components

This section reviews related work on interaction using the components of hybrid mPPI ensembles before embarking on an analysis of theories of interaction in the ensemble as a whole. For the most part, related work in this area took an explorative approach, where theoretic insights were derived from evaluating novel, creative approaches toward interaction. Thereby, interaction with all components of hybrid mPPI ensembles had been explored individually and in varying combinations: digital pens, paper artifacts and mobile devices (c.f., definition on page 2). However, the issues addressed with respect to individual components also highlight concepts and challenges of the domain as a whole.

Interaction Using Digital Pens

The digital pen presents the primary input device in the context of PPI. Its main capability is to enable input directly on paper. However, in order to broaden the design space for PPI, additional input mechanisms on the pen itself have been proposed: Song et al. describe the concept of using the pen surface to issue multi-touch gestures on the pen barrel, [Song et al., 2011]. This supports for example mode switching operations, or similarly chained interaction techniques while writing or drawing on the paper surface. Thereby, the results reported do not offer any theoretic insights, e.g., whether such additional control operations are disruptive with respect to the flow of interaction.

Feedback on the pen. The lack of direct feedback on the digital pen besides physical inking of paper has been emphasized as a problem to the design of interaction techniques by some: in order to alleviate this lack of additional feedback, Liao et al. designed a prototypical extension to the digital pen, which is capable to provide visual, acoustic and haptic feedback, [Liao et al., 2006]. A similar approach has also been used in the *LiveScribe* pen family, which employs several feedback mechanisms, [Schreiner, 2008]. These pen models are equipped with a small LCD display area attached to the side of the pen itself. This allows a tiny visual display of some limited feedback or information at the side of the pen. In addition to this, the pen provides acoustic feedback via a built-in speaker.

However, it remains unclear in these approaches, how distracting feedback on the pen might become to the user. Furthermore, it is not explored when to give feedback or how much of the feedback can actually be digested by users. For instance, if the pen vibrates while writing this might prove distracting to the user and information displayed at the side of the pen may go unnoticed. Concrete theoretic insights with respect to the role of feedback have not been reported.

Projection Feedback. Providing feedback, not on the pen itself but on the paper medium, has been explored. Song et al. introduced *PenLight*, [Song et al., 2009a], a system using projection on paper to visualize digital feedback. This approach was subsequently adapted by other systems, e.g., *Strip'TIC*, [Letondal et al., 2013] and *AR Lamp*, [Kim et al., 2014]. In Song's original approach, an enlightened area in front of the pen is used to visualize data and allow for direct manipulation menu selection (using radial menus). This enables interaction techniques known from pen based GUIs, e.g., lasso selection and pressing projected buttons.

MouseLight, [Song et al., 2011], further explores the concept of projected information on paper surfaces. Here the projection is not coupled to the pen itself, but to a secondary device similar to a computer mouse. This allows for pen based interaction with a decoupled area where digital information is projected, similar to the paper information lens, [Mackay et al., 2002]. *PenBook*, [Winkler et al., 2013], takes a slightly different approach in order to preserve the fine-grained tactile feedback offered by the digital pen moving over a paper-like surface. Here a projection area is attached to the mobile device and users interact with the system using digital pens on this surface. The pen itself does not produce physical ink, digital ink is recorded and projected onto the surface [Winkler et al., 2013].

Discussion. In these approaches, the strategy of *overlaid information* is used to ease the *media transition* between the paper medium and digital devices (which typically provide dynamic feedback). Although the authors claim to present findings applicable for mobile projection settings, e.g., in [Song et al., 2009a], the actual experimental setting of these prototypes consists of a stationary top-projection unit in combination with digital pens and paper. As a result, characteristics of the mobile setting, e.g., the re-arranging of paper artifacts on a surface, have not been considered. Furthermore, the distinguished role of mobile devices in conjunction with paper and digital pens are not part of their investigations. However, several approaches raised questions with respect to the role of *feedback* in hybrid mPPI ensembles.

Interaction using Paper Artifacts

The interaction with mobile systems using paper artifacts without employing a digital pen as input medium has also been examined in the literature. Liao et al. propose a gesture based system called *PACER*, [Liao et al., 2010a]. It supports embodied interaction with paper documents through a mobile phone, using the camera and visual features of the paper documents. Here the user can issue selection and other manipulation gestures on document content by using the mobile device itself. This allows for a new class of user interfaces as argued by Liu and Liao: the so called *PaperUI*[38], [Liu and Liao, 2012].

A similar approach is taken in *Hotpaper*, [Erol et al., 2008], which offers a recognition mechanism to link paper maps with digital information. In these novel interaction techniques the mobile phone is used like a tangible device. This has, e.g., been studied in [Hudson et al., 2010], where low-attention interaction techniques are proposed, and in [Edge and Blackwell, 2009] with a focus on bi-manual interaction. However, for hybrid mPPI ensembles, there are no theoretic insights reported with respect to bi-manual interaction.

Touch on Paper. Recent technological advances allow digitally enhanced glasses to be used in this context [Zhou et al., 2014]. Those glasses thereby offer *augmented reality* (AR) features, essentially projecting digital information on the visual image perceived by the user. This was used by Zhou et al. in their prototype to further augment paper documents without additional projectors while added tracking capabilities support touch input on paper documents, [Zhou et al., 2014]. However, none of the existing studies from that area take into account the pen as an additional interaction device and support fine grained input generation, such as handwriting or drawings. The approach rather focus on adding the paper surface and paper artifacts as interaction device.

An extension to *PACER* named *FACT*, [Liao et al., 2010b], introduces a first step toward this direction by supporting (non-digital) pen based input and interaction with document contents for nomadic settings, however, using a Laptop computer. Thereby, this approach follows the vision of a nomadic interactive desktop environment using mobile projection to overlay information on paper documents and camera based input recognition, an approach that has already matured in the stationary domain, e.g., as in *Strip'TIC*, [Letondal et al., 2013]. Extensions to *FACT* support, e.g., interaction with paper documents via mouse and keyboard in the mobile domain by means of mobile projection, [Liao and Liu, 2011].

Discussion. Approaches targeting mobile interaction with paper artifacts without digital pens highlight the role of paper itself as an interaction device. Thereby, the position, orientation and content of paper documents becomes important. Although this does not offer any direct insight into the question how to adequately design PPI in hybrid mPPI ensembles; it shows that paper documents itself have to be considered as active components in any theory of interaction for such ensembles. Additionally, the approaches here quite naturally employ the mobile device as embodied interaction

[38] not to be confused with *Pen-and-Paper User Interface* (PPUI), as defined on page 2; a *PaperUI* does not include the use of digital pens

device, i.e., the user can interact with the system by physically moving the mobile device to various positions related to the paper artifacts involved in the setting. This raises the question, how to integrate this form of interaction with PPI? The relative positioning of the ensemble components must be carefully investigated and integrated into related interaction theories.

Interaction Using Mobile Devices

Existing systems employing PPI in the mobile domain, e.g., by using smartphones or other mobile hand-held devices in conjunction with digital pen and paper, mostly constrain the use of the mobile device to mere remote functionality access and minimal feedback. For instance, in *EdFest*, [Signer et al., 2006], a paper map interface is used to access multi-media content via tipping with the digital pen on certain map locations. This corresponds to Pidget based interaction as described in section 2.4.1, with the mobile device only serving as a platform to present multimedia content. Likewise, the ”hotspot association gesture” from *ButterflyNet*, [Yeh et al., 2006a], where the user draws two edge marks as a placeholder for a photo to be inserted later, is designed in a way that avoids simultaneous and tightly integrated interaction with pen, paper and mobile device (a camera instead of a mobile phone here).

Pen and Touch. In the approaches above, the mobile phone itself is not used in integration with pen and paper to provide input to an application. Recent approaches therefore began exploring interaction techniques spanning paper and the mobile device in the same course of action generating valuable insights with respect to the combination of pen and touch in the mobile domain [Pietrzak et al., 2012, Tsandilas, 2012], a widespread combination of modalities on tabletop systems, e.g., in [Matulic and Norrie, 2013]. In *UbiSketch*, [Weibel et al., 2010a, Cowan et al., 2011], the user draws sketches on paper by using the digital pen and publishes these sketches via touch on the mobile device to a social network. Here, output on the mobile phone is used only to confirm that an action was performed [Weibel et al., 2010a].

Toward tighter integration, *S-Notebook*, [Pietrzak et al., 2012], offers a system for mobile note-taking (c.f., section 2.2.1) that extends the interaction surface of a smartphone through paper documents, while supporting links between digital content and written content on paper. The system introduces a temporal association similar to *ButterflyNet*, [Yeh et al., 2006a], in addition to a tap using two fingers on the mobile device. However, it does not report theoretic insights on the interaction device switch between pen and touch.

Tsandilas reports on a set of four interaction techniques spanning mobile device, digital pen and paper [Tsandilas, 2012]. Theses techniques aim to improve recognition results of digital ink. Results of a comparative study showed that some techniques were preferred by users over others. However, no further theoretical insights why users prefer some techniques while disliking others were presented. Furthermore, both approaches are hindered by supporting infrastructures as explained in section 2.5.3. Theoretical insights in how to actually design mobile PPI are strictly limited and concrete design considerations are not reported.

Discussion. Recent approaches toward mPPI show that integrating the mobile device as interaction device in the same course of action, e.g., by employing the touch modality as part of an pen-and-paper interaction technique, yields usable, tightly integrated setups empowering users to leverage the full potential of hybrid mPPI ensembles. For instance, such a tightly integrated setup allows improving online recognition results of digital ink, [Tsandilas, 2012]. Thereby, *media transitions* in the same course of action form the central characteristic of such interaction techniques. However, existing approaches fail to report on theoretic insights how to actually design such tightly integrated interaction techniques. What are the key challenges designers have to consider? What relationships do exist between conflicting design choices?

2.5.2 Existing Theories of PPI and mPPI

As shown above, numerous approaches describe the usage (c.f., section 2.2) of PPI and design of concrete interaction techniques (c.f. section 2.4.1). These approaches offer little theoretical insight into the domain as theoretical approaches mostly focus exclusively on the document life-cycle, e.g., as in *PADD*, [Guimbretière, 2003], or *iPaper*, [Weibel et al., 2007]. Others, such as *PaperWindows*, [Holman et al., 2005], focus on theoretical aspects of interaction design with paper-like displays, neglecting interaction by means of a digital pen. These approaches do not provide insights in the theoretic foundations of interaction design in hybrid mPPI ensembles.

However, theoretical analysis of interaction plays an important role in the interaction design live cycle [Rogers et al., 2011]. Interaction theories and frameworks provide the theoretical foundation offering interaction designers guidance and emphasizing particular problems of the domain. Here theories need to describe important concepts of the domain and need to offer predictive relations between these concepts. This ultimately results in a well founded set of design guidelines the interaction designer can use to determine which interaction techniques work in which settings.

PPI Theory Fundamentals

Traditionally, research in HCI theory applies theories from the fields of psychology, sociology or ergonomics to the domain of HCI. These theories include a huge spectrum ranging from cognitive theories of the human memory, e.g., Baddeley's theory of the *working memory*, [Baddeley, 1992], over the role of social interactions in the context of knowledge processing, e.g., Hollan's theory of *distributed cognition*, [Hollan et al., 2000], to theoretical models of perception motor coordination, e.g., the famous Fitts' law modeling the dynamics of pointing tasks, [Fitts, 1954]. However, the domain of PPI, let alone mPPI itself, has been systematically neglected by theoretical research.

In order to satisfy the need for interaction design guidance as outlined above, existing HCI theories need to take the specific requirements of PPI and mPPI into account. This view has also been supported by Steimle, who emphasized the lack of theory in the domain of PPI [Steimle, 2009a]. In PPI and mPPI based settings, concepts such as chunking, phrasing and chaining, [Buxton, 1986] (c.f., section 2.4.1) have to be used in a different context compared to traditional (read: purely digital) HCI. Well established interaction concepts used in contemporary GUIs such as direct manipulation, [Shneiderman, 1983], can only be used with several serious limitations in the context of PPI based settings.

Adding the factor mobility further complicates this situation: when using pen, paper and mobile devices in conjunction, switching between paper and digital devices during tasks might require compensation strategies, e.g., the use of imprecise gestures [Hudson et al., 2010]. Additionally, the designer might have to consider how the user interacts in a bi-manual way, e.g., by examining the bi- manual frame of reference [Hinckley et al., 1997], with pen and mobile device being held in the hand during interaction.

Steimle's Framework

Steimle's framework, [Steimle, 2009a], recently has gained some popularity as it presents the only theoretical framework directly targeting PPI. As described in section 2.4.3, it consists of a set of frequently occurring core interactions and conceptual activities. These were derived in a series of empirical studies in the context of knowledge work using pen, paper and tabletop computers. These core interactions describe important concepts with respect to user actions. However, other important domain concepts, e.g., attention switching between different media, are not included in the framework and no analysis results toward this have been reported.

In Steimle's framework, core interactions are combined using interaction chains, with the goal to perform certain contextual activities. However, the framework only enumerates possible core interactions and contextual activities, without explaining the relation between these two sets of concepts regarding the question of where to apply which. Therefore the practical relevance of the gained theoretical insights, i.e., the guidance offered to interaction designers, remains limited. It lacks concrete interaction design guidelines that could be derived out of the presented concepts.

Additionally, the considered stationary setting with a tabletop does not take the specific characteristics of mobile settings into account, e.g., the varying placement of the involved components. Hence the theoretical relevance to the design of mobile PPI remains limited.

Pen and Touch

Recently, Hinckley et. al. presented results of an extensive exploratory study on interaction in device ensembles comprising multi-touch enabled tabletop computers and pens [Hinckley et al., 2010]. Although not directly aimed at PPI, this work presents another important theoretical approach, as the investigated modalities, i.e., pen and touch, also occur in the domain of mobile PPI. Besides introducing novel interaction techniques for the combined use of pen and touch, they presented concrete design considerations for simultaneous use of these modalities.

These design considerations mostly apply to the domain of PPI also, both mobile and non-mobile. They offer insight in the design of interaction when switching between modalities (here: digital device with touch, WIMP or pen-based interface). However, the mobile setting and the physical properties of mobile device and paper introduce additional aspects: the (relative) physical placement and the limited input / output capabilities of the encompassed components, as well as switching between a physical medium (paper) and a digital medium (mobile device) during interaction. Furthermore, the interaction on paper documents typically differs from interaction on digital media, as established interaction concepts, e.g., direct manipulation [Shneiderman, 1983], cannot be applied in the same manner.

2.5.3 Deficits in Existing Theories

So far interaction in hybrid mPPI ensembles largely remains unexplored from a theoretical perspective. Interaction research focused on individual ensemble components, however, highlights several important concepts and challenges without explaining or further investigating them. For instance, the role of *feedback* in hybrid mPPI ensembles has been underlined in particular, e.g., in [Tsandilas, 2012], as well as the role of *media transitions*, e.g, in [Steimle, 2009a]. Current theoretic approaches thereby fail to offer any explanation of how these concepts relate to each other. In addition, there are no guidelines available on how to design interaction in the face of the aforementioned challenges, e.g., media transitions.

Related work has highlighted the combination of pen and touch input modalities in hybrid mPPI ensembles, e.g., in [Pietrzak et al., 2012], [Tsandilas, 2012] or [Weibel et al., 2010a]. This concept needs to be investigated further from a theoretical perspective, e.g., as done by [Hinckley et al., 2010], however, in this context with a focus on hybrid mPPI ensembles. In particular the impact of media transitions within a single course of action has to be explored further as this is a critical factor in tight integration of ensemble components, [Tsandilas, 2012]. Furthermore, the role of relative spatial positioning of paper and non paper ensemble components as well as where and when to provide feedback to the user requires deeper understanding.

Holistic theories of interaction in hybrid mPPI ensembles have not been suggested so far. Even for stationary PPI applications, the theoretic landscape remains scarcely populated. The only existing genuine theoretical PPI framework presented by Steimle, [Steimle, 2009a], describes several important concepts of the domain regarding interaction. However, it lacks description of relations between these concepts and hence does not offer more concrete design guidelines when it comes to choosing appropriate interaction techniques. Also, it focuses on a stationary setting excluding particular phenomenons of hybrid mPPI ensembles, e.g., the switching between small screen mobile devices and paper documents, as well as the combined use of paper and mobile device. This limits its applicability to mobile PPI. Similar limitations apply to Hinckley's work regarding the combination of pen and touch [Hinckley et al., 2010]; although the latter offers a set of concrete design guidelines, it targets stationary settings exclusively and does not include paper artifacts.

2.6 Summary and Conclusion

This chapter provided background information on basic domain concepts and technology, as well as a state of the art review of pen and paper interaction (PPI) with a focus on mobile PPI (mPPI) and hybrid mPPI ensembles. Thereby, it established the Anoto pen and paper technology as the most reliable technology currently available to realize mPPI. As such, Anoto will serve as the base technology used in the reference implementation of concepts introduced throughout this thesis. However, the assumptions for concepts and findings presented throughout this thesis are not restricted to any technology: the findings apply to any absolute positioning technique (c.f., section 2.1.2).

Furthermore, this chapter provided a survey of application domains for PPI. It thereby outlined the lack of support for mobile usage practices in existing approaches. Further analyzing *why* mobile use is not supported on a broad scale, this chapter embarked on an in-depth analysis of infrastructures, conceptual frameworks and interaction theories.

Starting with infrastructures (as being the core of enabling mPPI), a set of requirements was derived which such infrastructures need to satisfy in order to support mobile usage characteristics real of pen and paper. Based on these requirements, existing infrastructures were analyzed and their shortcomings were demonstrated. Following the approach outlined in chapter 1, the analysis was then extended to conceptual frameworks of PPI, as those form the connecting element between infrastructures and interaction. Here, current interaction techniques were reviewed and a set of requirements for conceptual frameworks was derived. Subsequently, existing conceptual frameworks were analyzed and their shortcomings were illustrated. Finally, existing theoretical insights and holistic theories of interaction for PPI were analyzed and the lack of a theory for interaction in hybrid mPPI ensembles was demonstrated.

Infrastructures. Existing infrastructures fail in supporting mobile usage practices of pen and paper, such as *user mobility* and *document mobility*. Thereby user mobility refers to practices where the user is on the move, while document mobility refers to practices where documents are passed on between users or organizations. These two mobile usage practices entail a set of requirements toward mPPI infrastructures that no existing approach satisfies in its entirety: support for interactive mode in mobile settings (R1.1), a platform specifically designed for processing of mPPI on mobile devices (R1.2) and flexible deployment schemes (R1.3) in the case of user mobility; sharing of resources between applications (R2.1) and the discovery of interactive regions (R2.2) in the case of document mobility. In particular, no current approach supports flexible deployment schemes (R1.3). Infrastructures rely on monolithic deployments impeding support for a flexible mobile approach toward mPPI processing.

This lack of infrastructures specifically specifically designed to support mobile PPI obviates development and research of mobile PPI based solutions. In order to alleviate this impediment and ultimately enable mobile PPI, such a mobile PPI infrastructure is required. It needs to be designed to address the specific requirements of mobile application of PPI, while at the same time, allowing PPI to be employed in non-mobile use cases.

Conceptual Frameworks of Interaction. Conceptual frameworks of interaction form the basis of toolkits. As such, they provide the connecting element between infrastructure and interaction. However, existing conceptual frameworks of PPI fail to address the basic requirements toward serving as foundation for toolkits: provide an adequate design vocabulary (R3.1), allow composition of interaction techniques (R3.2), remain open and extensible (R3.3) and finally be machine understandable (R3.4). This results in existing toolkits for PPI (there are none for mPPI) basing on conceptual frameworks not matching the concepts of PPI / mPPI, i.e., the traditional GUI inspired WIMP paradigm.

Here, a genuine conceptual framework for PPI is needed. Such a framework needs to support describing interaction through an adequate design vocabulary, allow composition and remain open and extensible. It also needs to provide machine understandable descriptions of interaction techniques: descriptions the infrastructure can recognize. Furthermore, it has to be shown that such a conceptual framework is able to serve as basis of a PPI or mPPI toolkit.

Theory of Interaction. Theories of interaction help interaction designers in understanding important concepts, phenomena and their interrelations. Through a set of derived guidelines, a theory enables designers to make informed choices with respect to interaction design in hybrid mPPI ensembles. Research with respect to mobile PPI has highlighted some important characteristics, e.g., the role of feedback in hybrid mPPI ensembles and the combination of pen and touch interaction. However, these approaches do not offer holistic theories with explanatory power entailing concrete interaction design guidelines. The only holistic theory of PPI does not address a mobile setting and its applicability therefore remains limited to a stationary context. Thus, basic domain concepts, e.g., the combination of pen and touch interaction, and their interrelations have so far only been analyzed in stationary settings.

This raises the need for a theoretical framework explaining the dynamics of mobile PPI. A comprehensive theory needs to be derived, particularly targeting interaction in hybrid mPPI ensembles. This theory needs to explain the important concepts of the domain and their interrelations. Furthermore, it has to allow deriving a set of concrete design guidelines aiding interaction design in the domain.

3 Infrastructure for Mobile Pen-and-Paper Interaction

Synopsis: *This chapter introduces a novel infrastructural approach supporting mobile PPI: the* distributed PPI processing pipeline*. It describes its basic architecture and demonstrates how this approach enables infrastructures to satisfy the requirements for mobile PPI support established in chapter 2. It outlines detailed design considerations for required components and exemplifies these concepts by means of its reference implementation* Letras*. It concludes with an evaluation of the novel infrastructure using the case study of the digital grocery list application (c.f., chapter 1) and a set of prototypical applications. Thereby, it demonstrates and assesses the practical relevance of concepts introduced.*

The conceptualization of an mPPI infrastructure laid out in this chapter follows a systematic stepwise approach. First, the basic paradigm employed towards mPPI processing is derived in section 3.1. In an initial step, the foundation is laid with a concise generic model for the processing steps incurring in all PPI and mPPI settings. This model, called the *Generic PPI processing pipeline*, informs the conceptualization of PPI processing for the mobile use case in the next step. Here, the generic PPI processing pipeline is adapted to satisfy the requirements toward infrastructural support for mobile PPI (c.f., chapter 2, section 2.3.1). This yields the novel *distributed PPI processing pipeline* supporting mPPI.

Second, a concrete infrastructure for mPPI is developed based on the distributed PPI processing pipeline and introduced throughout section 3.2. Detailed design considerations are outlined and required components discussed. Thereby, as introduced in chapter 2, section 2.3, this infrastructure provides the environment offering common functionality and services to applications reducing the amount functionality replicated in each application, while at the same time influencing the user interface as little as possible [Edwards et al., 2010]. It is capable of supporting both, PPI and mPPI, however unique in that it provides the first infrastructure specifically designed to preserve mobile usage characteristics of pen and paper.

Third, *Letras*, a reference implementation and generic application development platform based on the novel mPPI infrastructure is introduced in section 3.3. It exemplifies how to apply the concepts introduced in section 3.2 in a real-world system.

Thereby, it provides a distributed, highly flexible approach toward system support for PPI and mPPI, particularly targeting mobile settings[1]. It is available for free download under the Mozilla Public License[2].

Finally, the mPPI infrastructure is evaluated in section 3.4. First, the presented infrastructure is analytically evaluated at the conceptual level with respect to fulfilling the requirements established in chapter 2. Second, a *proof-of-concept* evaluation is given through the case study and implementation of the digital grocery list (c.f., chapter 1, section 1.1.2) in combination with a set of mobile applications developed on top of the infrastructure.

3.1 Mobile PPI Processing

As shown in chapter 2, existing infrastructures are designed to predominantly support stationary, non-mobile use. Most approaches thereby focus on a single application deployed along with the infrastructure. If a multi-application scenario is targeted, e.g., as in iServer/iPaper based approaches [Norrie et al., 2006a], a single server hosts the application logic and functions as central authority managing all interaction resources.

This approach yields a monolithic deployment, i.e., resources in use are attached to a central processing environment and cannot be shared between multiple, heterogeneous environments. Fundamental to this approach, is the assumption that all parts of the system (pen, paper artifacts, computer hardware, application software components) are deployed and controlled by a single authority. However, this assumption does not hold in realistic, distributed PPI and mPPI scenarios [Signer et al., 2014].

Existing infrastructures based on such a monolithic deployment show severe limitations with respect to mobile PPI processing as shown in chapter 2, section 2.3.2. Mobile usage concepts such as *user mobility* and *document mobility* are not (completely)supported. Therefore a novel approach for organizing the PPI / mPPI infrastructure is required that supports (c.f., section 2.3.1)

R1: User Mobility Interaction with pen and paper while being in a mobile or nomadic setting; this encompasses full support for the *interactive mode* (R1.1), support for execution on resource constraint *mobile platforms* (R1.2) and *flexible deployment* of different processing components (R1.3).

R2: Document Mobility Passing of documents between users and organizations both intra-organizational and inter-organizational, encountering of new documents and applications on the fly, switching between different applications

[1] although, as stated above, stationary settings are supported also

[2] https://github.com/fheinrichs/letras (accessed: July 2015)

and simultaneous use; this encompasses *resource sharing* between applications (R2.1) and *interactive region discovery* to associate data with newly discovered applications (R2.2).

In order to develop such an infrastructural approach aimed at supporting mobility, this section derives the basic architectural paradigm enabling infrastructure to satisfy these requirements.

Approach. As laid out at the beginning of this chapter, the basic idea here is to initially derive a generic abstraction of the process of PPI processing in general, both in mobile and non-mobile contexts. This generic abstraction, the *generic PPI processing pipeline*, determines the functional components of all PPI and mPPI processing infrastructures. In its original, unmodified form, it also corresponds to the basic architectural paradigm implicitly employed in existing, monolithic infrastructures.

Subsequently, this architectural paradigm is extended into the *distributed PPI processing pipeline*. In contrast to the generic PPI processing pipeline, this novel approach allows supporting user mobility and document mobility at the architectural level due to its inherent deployment flexibility. Hence, it provides a suitable set of basic abstractions and concepts to serve as foundation for an infrastructural approach specifically targeted at mobile PPI; as a side effect, infrastructure based on this concept supports the non-mobile use case also.

3.1.1 The PPI Processing Pipeline

The role of infrastructure is to provide common functionality and services to applications thereby reducing the amount functionality replicated in each application [Edwards et al., 2010]. In the context of mobile and ubiquitous computing, the infrastructure can be regarded as the logical driver layer of an assumed ubiquitous computing operating system, the so-called *Meta Operating System* [Román et al., 2002]. In this role, it abstracts from the environment and provides a common interface for applications to access all resources and offer all functionalities as needed. As such it serves as the basis of any system employing PPI or mPPI.

In order to achieve this, the infrastructure has to provide a common set of interfaces and abstractions to the PPI based application. Basically the infrastructure accesses the digital pen hardware and prepares generated data, i.e., *digital ink* as defined in chapter 2, section 2.1.1, into a form usable at the application level; subsequently dispatching relevant data to the application. *Relevant* here refers to pen data recorded at the interactive regions an application is interested in. Thereby applications define these interactive regions of interest according to their application model, e.g., a paper sheet

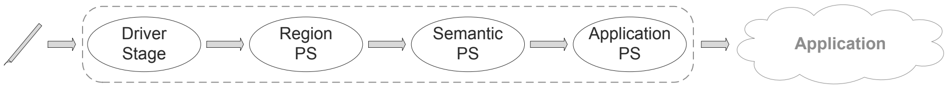


Figure 3.1: The Generic PPI Processing Pipeline: Successive processing stages (PS) transform digital ink data deployed in a monolithic scheme (as in traditional approaches)

with several Pidget interactors on top of it. Upon receiving digital ink, applications invoke functionality as defined by the application logic.

Digital Ink Processing. Processing of PPI requires access of movement data captured by digital pens on a surface. As defined in section 2.1.1, areas on which movement data can be captured are referred to as *interactive regions*. These could be paper documents (and parts thereof) or other media, e.g., a specially prepared foil on a digital screen. The infrastructure then successively transforms this data, from raw sensory information to higher-level data constructs, dispatching it to interested applications. Interactive regions form the backbone of the application model, as they map regions in physical space to regions in the global coordinate system used by the digital pen.

As an example consider a form filling application where the user needs to confirm an action using a check mark gesture in a predefined check mark box. Thereby, the check mark box corresponds to an interactive region the application is interested in. First, the infrastructure needs to access the digital pen operated by the user. It records its movement data, e.g. pen tip at position x, y. Any recorded data within the interactive region is of interest to the application and has to be dispatched accordingly. However, only check mark gestures are important, so the infrastructure takes the raw data and transforms it into a pen stroke vector, a so called *trace*. This is then interpreted, e.g., by a gesture recognizer and the application finally receives meaningful events, e.g., ”check mark” gesture performed.

Processing Pipeline. Successive transformation and aggregation of data implies a pipeline architecture, consisting of a sequence of *processing stages* (PS). A processing stage here refers to a certain part of the pipeline with a clearly defined task with respect to the data traveling through the pipeline. Data flows through the pipeline starting at a source, e.g., the digital pen, and ultimately reaching a sink, e.g., the application using this data to invoke a certain functionality. Thereby each processing stage takes a certain type of data as input, applies its functionality and produces a certain data as output.

In the domain of PPI processing four elementary transformation steps for the processing of digital ink can be distinguished. These are common to the tasks performed in existing approaches. Therefore these steps yield elementary processing stages constituting the elements of the generic pipeline architecture:

Driver Stage The driver stage connects the digital pen hardware to the rest of the pipeline. It provides appropriate hardware abstractions to higher-level processing stages, establishes data connections via bus or network protocols and transforms data into low-level data structures. As such it encapsulates the drivers for digital pens (hence its name). The challenge for the infrastructure lies here mainly in accessing and decoding the data generated by digital pens. Typically the data produced in this stage consists of *samples* and *events*. Samples describe sensory information provided by the digital pen when determining the position of the pen on paper, e.g., pen tip position in 2-dimensional coordinates associated with recording time. Events encapsulate elementary state information of pen usage such as ”pen tip down”.

Region Processing Stage Interactive regions provide the basic link between digital functionality and physical paper and thus constitute the most important building blocks of PPI based applications. Each PPI based application defines at least one interactive region as area of interest, although typical applications use more complex region hierarchies (e.g. printed buttons contained within a writing region). The region processing stage relates digital ink data to the interactive regions defined by the applications and channels it to successive processing components. It uses knowledge of defined interactive regions and checks whether received data lies within one of these regions. If the data is relevant it can be further processed by the pipeline.

Semantic Processing Stage The next step transforms relevant digital ink into common higher-level semantic structures, adding meaning and interpretation. Such transformation depends on the specific needs of a PPI based application. Examples for semantic processing are segmentation of digital ink into text and drawings, form filling data, free notes, gesture recognition or even handwriting recognition. Here specialized frameworks exist, e.g., the segmentation framework introduced by Ispas et al. [Ispas et al., 2011] or the iGesture framework introduced by Signer et al. [Signer et al., 2007b]. These steps are commonly executed by the infrastructure in order to avoid replicating the implementation of complex recognition algorithms.

Application-Level Processing Stage Finally, the digital ink data and the results of its semantic interpretation are processed and interpreted according to the ap-

> plication logic. Depending on the type of application and its degree of interactivity, appropriate actions are triggered. For example, a digital notebook might render the digital ink data to produce a facsimile of the physical ink written, while a dictionary lookup application might obtain a *selected* (e.g., as result of a gesture) *word* (e.g., as result of the handwriting recognition) and perform the dictionary lookup. However, a set of common tasks at the application level remains, which should be supported by the infrastructure. This includes, but is not limited to, rendering of digital ink and the persistent storage of recorded data structures, as well as structured querying of pen data.

This processing pipeline forms the architectural basis of PPI processing infrastructure components. Existing, monolithic approaches implicitly employ this pipeline design. For example, in the *PaperToolkit*, [Yeh et al., 2008], a set of subject / observer abstractions are provided that essentially form the pipeline when assembled into an application. In these approaches all components are deployed in a monolithic way, i.e., a single software stack within a single runtime environment, typically packed together with a single application. This means that each application replicates the necessary processing stages, a setup depicted in Fig. 3.1 by the gray rectangle around the interaction processing pipeline (which denotes the processing infrastructure).

As demonstrated in section 2.3, existing infrastructures basing on the monolithic deployment of the generic PPI processing pipeline fail to provide adequate support of mobile use of PPI. This is due to the fact that the monolithic deployment introduces a set of limitations with respect to mobile PPI, especially with respect to supporting *user mobility* and *document mobility*, as resources are locked into a single pipeline instance (for a detailed analysis please refer to section 3.4.1).

However, In order to overcome these limitations, and lay the foundation for a mobile PPI processing infrastructure, the generic pipeline architecture can be extended into the *distributed PPI processing pipeline*.

3.1.2 Distributed PPI Processing

The basic idea behind the distributed PPI processing pipeline is to decouple the processing stages in order to allow applications to share pipeline components and hence to provide shared access to resources. This setup allows for a distributed deployment, as opposed to the monolithic deployment described in section 3.1.1.

To facilitate this, processing stages are conceptually and functionally decoupled and successive processing stages are connected exclusively via so called *processing stage interfaces* (PSI). Processing stage interfaces are clearly defined interfaces describing data format and available services which interconnect processing stages as depicted in Fig. 3.2.

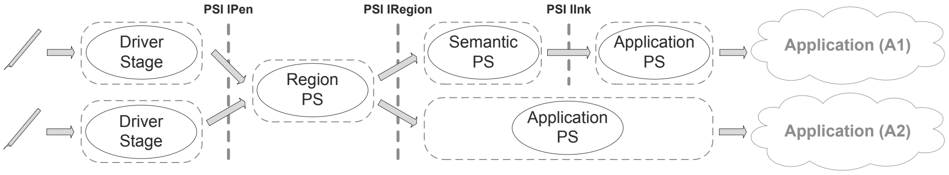


Figure 3.2: The distributed PPI Processing Pipeline: Successive processing stages (PS) transform digital ink data deployed in distributed scheme to allow for sharing of resources between applications (example setup)

Each processing stage *consumes* data in a form defined by its input PSI and *produces* data in a form defined by its output PSI. It also offers a set of services allowing to obtain this data. Thus, processing stage interfaces define stages of data, communication channels used and services offered; whereas processing stages generate data and offer defined services.

In the distributed processing pipeline, processing stages can be deployed independently and thus physically distributed within the runtime environment. This adds the deployment flexibility required in order to lay the foundation for mobile PPI processing support while at the same time supporting non-mobile use cases (as the generic processing pipeline becomes *one* of the possible deployment options). A gray box around each processing stage indicates this concept of decoupling and physical distribution in Fig. 3.2.

There are three processing stage interfaces defined in the distributed PPI processing pipeline. The name of each PSI indicates the main data construct or service playing a role in the processing of PPI at this stage. The processing stage interfaces are

IPen At this stage in the pipeline the main data construct is referred to as *raw digital ink*, i.e., digital ink data that has not yet been further processed. It consists of samples and events. Thereby, samples describe the position of the pen tip, while events indicate interaction events, e.g., putting the pen to the surface. However, the most important abstraction at this interface is the generic pen service: a software service, or more precise a collection of services, that hides connection and pen model specific methods and provides a uniform resource access to all available digital pens. Hence the name of this PSI. This PSI decouples all sources of pen data from the rest of the pipeline.

IRegion Here the raw digital ink has been related to its enclosing region(s) and been processed accordingly. The main data constructs therefore are *region related digital ink* and the *interactive regions* themselves. Region related digital ink has been clustered into traces which describe movement sequences with the pen on the surface. This digital ink consists also of samples and events. These are now dispatched to components interested in the interactive region the interaction occurred on. This is achieved by offering services encapsulating the data on specific interactive regions only, thus giving this PSI its name. Here uniform access to all defined interactive regions and their data is provided.

IInk Here the digital ink on a certain interactive region has been semantically processed. This adds meaning and interpretation to the data, thus further classifying the data structures obtained in the previous stages. Now the digital ink has been transformed into its final data structure, the *processed digital ink*. In this PSI, a set of services offer classification results obtained by semantic processing components, basically injecting these results into the pipelined data stream. The obtained digital ink is now ready to be used at the application level.

Although these PSIs introduce additional complexity into the pipeline design, they offer numerous advantages in the context of mobile PPI processing. Fig. 3.2 shows how the distributed pipeline design allows sharing pipeline stages, both between instances of the pipeline and applications. This provides for an intrinsic non-exclusive access to resources, e.g., pens and interactive regions can now be shared by applications. Resources are no longer locked into a specific pipeline deployment. This lays the foundation for infrastructures supporting flexible deployment (R1.3) and resource sharing (R2.1).

Furthermore, the connections of two successive processing stages via PSIs can be realized using a networking middleware. This makes it possible to distribute the processing stages physically, depending on the available computing resources. Ultimately it enables flexible deployments (R1.3) and operation on the mobile platform (R1.2), as resource intensive processing tasks, e.g., as part of the semantic processing stage, can be delegated to backend servers (assuming existing connectivity etc.).

Essentially, connecting successive processing stages via PSIs over communication channels offered by a networking middleware further increases the flexibility of the concept of distribution. It also allows for a wide spectrum of so called *deployment schemes*.

Deployment Schemes

Deployment schemes describe the deployment of the pipeline stages, their physical and or logical distribution and their interconnection at runtime. In the narrower sense, a deployment scheme describes where a processing stage will be executed and which processing stages will be shared by multiple instances of the pipeline. In the broader sense a deployment scheme describes the general processing paradigm employed by

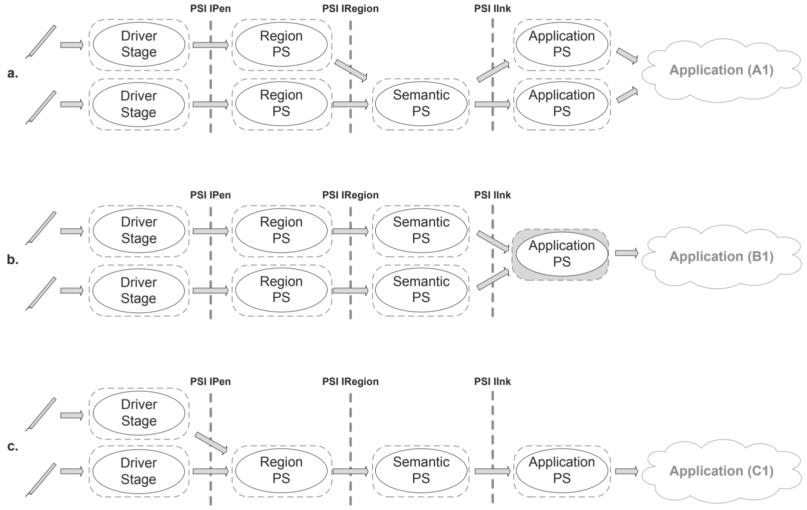


Figure 3.3: Examples of deployment schemes in the distributed processing pipeline: a. note taking application with a desktop and mobile client, b. the shared whiteboard application with personal pens, c. the borrowed pen setting

the pipeline. The distributed processing pipeline enables a broad variety of different deployment schemes. This is a crucial advantage over a fixed monolithic deployment when it comes to support mobile usage practices as it allows the pipeline adapting to the current use case as defined by the application (or the set of applications).

In order to illustrate the flexibility of the concept of deployment schemes in the distributed processing pipeline, consider the three example schemes depicted in Fig. 3.3.

The first example corresponds to a note taking application allowing the user to employ PPI in order to edit notes. It features a desktop and a mobile client, both supporting input via digital pens. The mobile client delegates processing intensive tasks of the semantic processing stage, e.g., handwriting recognition, to the desktop pipeline in order to save battery. Its deployment scheme is depicted in Fig. 3.3 a., the mobile client being the top row.

The second example depicted in Fig. 3.3 b. is a shared whiteboard application where users can interact with their personal pens on a collaboratively used whiteboard. In this deployment scheme, the first processing stages are deployed on the individual mobile clients connecting with the whiteboard. Only the rendering and data storage, i.e., the application processing stage, is handled on the whiteboard server (indicated by a gray background of the application PS in 3.3 b.).

The third example is a mobile PPI based application supporting sharing of a pen between different users. Here, the cumbersome manual setup of the bluetooth pen connection is abstracted away by the flexible pipeline design. The driver stage remains deployed on the mobile device of the first user lending out her pen. All further stages are deployed on the mobile device of the user actually using the pen for interaction. Fig. 3.3 b. depicts this setting.

In this broad range of heterogeneous deployment schemes, the monolithic deployment itself can be considered a degenerate case in the context of the distributed processing pipeline: Even with the distributed pipeline architecture, it is possible to deploy all processing stages alongside the application. This then yields the monolithic deployment found in most existing infrastructures (c.f., section 3.1.1).

Another deployment scheme emulating the approaches taken by existing infrastructures is to host the driver stage on a client device, while the other stages are hosted at a central server. This then yields the client / server approach, e.g., as taken by *iServer / iPaper*, [Norrie et al., 2006a]. Thus the flexibility of the distributed approach enables deployment as in existing architectures while at the same time adding support for alternate deployment schemes supporting the mobile use case.

Sharing of PS. Sharing of processing stages (PS) between applications is thereby facilitated through two concepts: *services and interfaces*, as described above, and the *pipeline base architecture*. While services and interfaces ensure that decoupling and physical distribution is possible, the pipeline base architecture, where data continuously moves forward through the processing pipeline, ensures it is feasible. Services thereby are not required to be completely state-less, e.g., recognition services will track and change their state according to the samples and events received through the pipeline. Re- entrant usage of services as such, however, does not occur as data successively travels through the pipeline.

It is important to note here, that back channeling of data has to be possible: Consider, for instance the semantic stage as recognizing a certain gesture, that would then trigger handwriting recognition of previously recorded digital ink. In this case, infrastructures based on the paradigm of the distributed PPI processing pipeline need provide adequate means to enable such concepts, e.g., through a flexible *Micro Service* approach in the semantic and applications stages (c.f., section 3.2).

Required and Optional Stages

Not all processing stages are essential to all applications. Consider for example a simple drawing application for multiple users, which does not require any semantic processing of PPI. Also the rendering is done by the application itself and it stores only a bitmap version of the resulting drawing. Such an application requires only the Driver

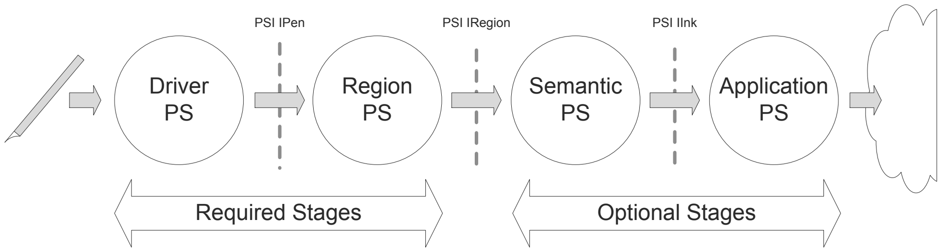


Figure 3.4: The distributed PPI Processing Pipeline: required and optional processing stages

PS to connect the pen hardware and the Region PS to identify whether the interaction did occur on the drawing region. In the context of the distributed processing pipeline, this concept can be used to further classify available processing stages

As processing of digital ink at the very basis means connecting a digital pen as input source and relating ink to an interactive region, e.g., a paper document, the Driver PS and the Region PS form the *required processing stages*, whereas the Semantic PS and the Application PS form the *optional processing stages*[3]. As trivial as this distinction might sound at first, it is important: focusing on the required stages and optimizing their implementation yields better resource utilization. This enables the pipeline to adapt more flexibly to mobile settings and to the needs of applications. Thus, it reduces the deployment size and provides a lower footprint of required software components. Ultimately, this supports operation on mobile platforms better.

3.1.3 Interactive Region Publishing and Discovery

In order to fully support the interaction with encountered documents as characterized by document mobility, the infrastructure must be able to cope with previously unknown interactive regions. This is the task of the *interactive region discovery* component (IR discovery). This problem has two facets: first, applications require a mechanism to publish digital representations of their interactive regions so that they can be found by the infrastructure; second, pipeline instances require a mechanism to actually discover these interactive region representations.

[3] This concept of required and optional components can even be broken down on a sub-stage level: For instance, an application might not need multiple drivers for different digital pen models, or might only want to support a trivial interactive region discovery mechanism. In this case not all services of the required PS might be relevant and a minimal deployment might cherry pick those which are. Please refer to section 3.2 and 3.3 for a discussion of concepts enabling these micro-deployments.

In the distributed interaction processing pipeline, the region processing stage dispatches digital ink to components interested in particular interactive regions. Thereby, it needs to utilize the mechanism for discovery in order to associate digital ink with new, previously unknown interactive regions. Following the principle of high cohesion, i.e., grouping related functionality in a single component, the region PS thus also serves as entry point for publishing of interactive regions by applications.

An application registers its interactive regions through digital representations of interactive regions at a region stage instance. The region stage then stores all known interactive regions in its *interactive region model*. This model is subsequently shared with other regions stage instances in order to allow for interactive region discovery. The publishing and discovery of interactive regions thereby boils down to construction and search of the shared interactive region model.

Here the distributed processing pipeline already provides an inherent architectural advantage. Several deployments can share the same region stage, hence these already share local knowledge of interactive regions. However, this alone does not suffice: in order to provide high performance processing of digital ink as required by interactive systems employing PPI with instant feedback, the dispatching of digital ink cannot be handled on a single, central component. Round-trip times for requests and scalability issues render this approach inadequate. Thus, sharing of interactive region models between different instances of the region stage needs to be supported.

In the distributed processing pipeline this sharing of interactive region models follows a 2-level approach as depicted in Fig. 3.5. This bases on the principle that the infrastructure essentially provides the logical driver layer in an assumed meta operating system for smart spaces [Román et al., 2002], where the publishing and discovery of interactive regions follows the locality principle suggested by Hartl et al. [Hartl et al., 2002]: cascading levels of interoperation between components are described starting from components in the direct vicinity of the user to interoperation on global scale with backend connectivity. The two levels of sharing employed in the distributed interaction processing pipeline are

1st Level: Local IR discovery This level refers to local sharing of interactive region models between all region stage instances in a certain fully-meshed network, e.g., in a local area network. The model is fully synchronized between all instances. This facilitates intra-organizational publishing and discovery of interactive regions as required in order to satisfy R2.2 (c.f., chapter 2, section 2.3.1). Not all interactive regions visible at this level need to be also visible on a global level, i.e., an application might re-use an interactive region belonging to another application. This mechanism supports document mobility in the sense of the same document being re-used in a different application, e.g., the phone call notes scribbled at the side of a drawing sheet.

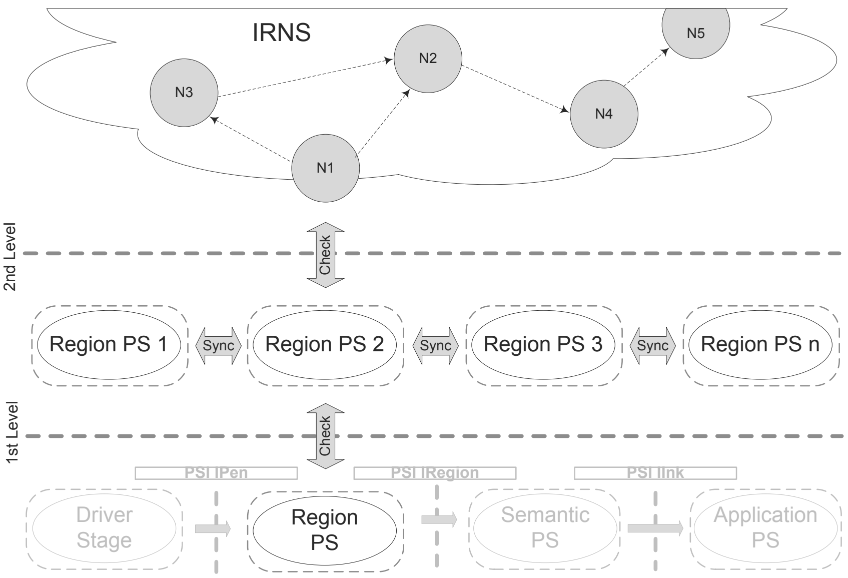
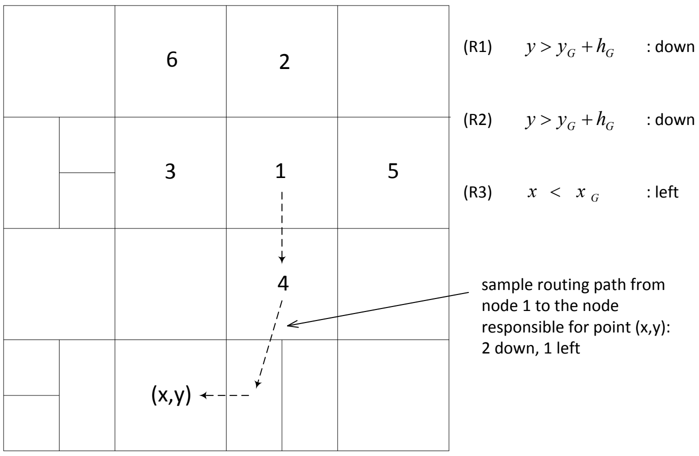


Figure 3.5: 2-Level approach to publishing and discovery of interactive regions

2nd Level: Global IR discovery Publishing and discovery of interactive regions on a global scale corresponds to construction and search of a global interactive region model. This global IR model is thereby constituted of all local interactive region models. The system allowing for search and construction of this global model is referred to as the *Interactive Region Name System* (IRNS). The IRNS allows the inter-organizational sharing of knowledge on interactive regions required to satisfy R2.2 (c.f., chapter 2, section 2.3.1) and thus supports interaction with previously unknown interactive regions crossing organizational boundaries. Consider for example an interactive, PPI enabled leaflet distributed by a retailer to its customers (as in the *DGL* case study, c.f., section 3.4.3). Here, the PPI / mPPI processing infrastructure at the customer side uses the IRNS to discover the application adding interactive functionality to the leaflet.

Thereby, lookup for an unknown interactive region is initiated by the region stage. The approach is as follows: First the local region model is checked, whether it knows the region. If the region cannot be found there, the lookup request would be delegated to the IRNS to retrieve the part of the global region model that is associated with the particular unknown interactive region.



sample routing path from node 1 to the node responsible for point (x,y): 2 down, 1 left

Figure 3.6: Sample region discovery in the CAN based IRNS (source: adapted from Ratnasamy et al. [Ratnasamy et al., 2001])

Interactive Region Name System

Guimbretière [Guimbretière, 2003] compared the lookup system for interactive regions (or in his narrower interpretation, for interactive paper documents) to DNS for paper. This is based on the fact that in terms of the problem size it corresponds to, if not exceeds, the *Domain Name System* (DNS)[4]. However, because of the size of the resulting global model and its inherently high degree of change due to different organizations controlling its interactive regions, as well as highly dynamic definition of interactive regions (e.g., compared to the less dynamic assignment of hostnames to IP addresses in the DNS), centralized storage of this global model cannot be achieved.

However, *Peer-to-Peer* (P2P) systems offer decentralized data storage solutions to this problem: *distributed hash tables* (DHT) [Balakrishnan et al., 2003]. Here a network of nodes store the data. For any lookup in the shared data, it is sufficient to know a single participating node. Then, just like in a hash table, these systems map a key via a hash function to its associated value. The employed mapping function also provides optimized routing within the network of nodes storing the values. Typical systems utilize an *Overlay Network* of notes, a term that refers to a logical network of nodes on top of an existing network, e.g., the Internet. Numerous approaches for realizing efficient construction and routing (which corresponds to search or discovery) of DHTs exist [Stoica et al., 2001, Ratnasamy et al., 2001, Rowstron and Druschel, 2001, Zhao et al., 2001].

[4] http://tools.ietf.org/html/rfc1591 (accessed: July 2015)

In the domain of PPI, Weibel suggested a P2P based approach for the lookup of paper documents [Weibel, 2009, p. 199]. However, the virtual binary search tree used in the underlying P-Grid system [Aberer et al., 2002] does not utilize a key characteristic of PPI: the perfect 2-dimensional hash function provided by the global positioning mechanism used in the underlying digital pen technology. Therefore a DHT is much more promising regarding the lookup and construction of IRNS. With a perfect 2-dimensional hash function given, a very efficient routing and overlay network construction can be achieved using a *Content Addressable Network* (CAN) [Ratnasamy et al., 2001].

The CAN for the 2-dimensional case allows directly associating nodes with certain macro regions in the underlying pattern space, while the partitioning of the space can still remain very flexible to avoid waste of pattern space. Each node participating in the IRNS overlay network manages a rectangular area of the global coordinate space. It stores the part of the region model that encompasses all regions within this area. If a region stage instance performs a lookup in the IRNS, it needs to inquire at a single previously known IRNS node providing a point in the global pattern space coordinate system as reference, i.e., a coordinate it had received to which the region stage cannot relate any region within its local region model. Now the IRNS node checks whether the reference point lies within its rectangular region. If this is the case, it returns the relevant part of the global region model. If not, it delegates the lookup along either the x or the y axis to its neighboring nodes in the CAN based IRNS. An example for such a lookup is shown in Fig. 3.6.

3.2 Distributed mPPI Processing Infrastructure

Section 3.1 laid the foundation for developing a novel infrastructure for mobile PPI processing by introducing the distributed PPI processing pipeline as a fundamental architectural paradigm enabling support for mobile usage practices. Now a concrete infrastructure for mobile PPI can be derived, specifically designed to satisfy the requirements of mobile PPI as laid out in chapter 2. Toward this end, this section elaborates on design considerations and basic concepts, gradually introducing a novel, generic platform for mPPI processing based on a distributed processing pipeline. Although the platform allows supporting non-mobile scenarios as well, i.e., PPI as opposed to mPPI, its main focus lies on mobile usage concepts as outlined throughout section 3.1.

3.2.1 Basic Platform Concepts

The basic idea of the distributed PPI processing pipeline as described in section 3.1 is to decouple processing stages (and hence interaction resources), both from particular applications and deployment locations. The deployment becomes *flexible*, enabling a variety of different deployment schemes and approaches to share existing interaction resources.

The platform supporting PPI in the mobile domain based on the distributed pipeline architecture follows this principle of flexibility. It provides a flexible, yet light-weight approach toward interaction processing: Successive processing stages with clearly defined separated interfaces channel digital ink data to applications and transform it appropriately. The two basic building blocks of the infrastructure, as derived in section 3.1, hence are

Processing Stages (PS) The individual components in the pipeline transforming the data, i.e., the *Driver Stage* accessing the pen hardware, the *Region Stage* relating digital ink to interactive regions, the *Semantic Stage* interpreting the digital ink and the *Application Stage* preparing it for processing at the application level

Processing Stage Interfaces (PSI) The interfaces between processing stages as points of distribution consisting of data definitions and services, i.e., the *IPen* interface describing raw digital ink and services abstracting from the pen hardware, the *IRegion* interface relating this to interactive regions and organizing data into movement sequences and the *IInk* interface describing the full tree-like data structure of digital ink with attached interpretation

In order to provide the flexible and light-weight platform the concept of required and optional stages needs to be supported within the platform (c.f. section 3.1.2). This enables the infrastructure to cope with changing environment conditions, i.e., the infrastructure can ensure that the required stages remain accessible, while the optional stages can be deployed if needed and if resources permit. Additionally, a runtime re-deployment needs to be possible, including flexible resource sharing via appropriate handover mechanisms.

In a platform for mobile PPI processing based on the distributed PPI processing pipeline, this is achieved by employing a *plug and play* interoperability scheme between the stages of the distributed PPI processing pipeline: stages can discover other stages by using standard service discovery methods and start interoperating within a flexible publish / subscribe based infrastructure. Following this approach, stages are deployed independently and have the ability to ”plug together”, i.e., stages can discover other stages and interoperate requiring minimal configuration at deployment time, by subscribing to, or publishing at pre-defined communication channels.

On the one hand this facilitates the physical distribution required to support PPI in the mobile domain, e.g., to allow the processing intensive handwriting recognition to be executed on a backend system and not on the mobile client itself. On the other hand, it allows for redistribution according to the current needs of an application: Imagine a hybrid mobile / nomadic scenario, where a user interacts with her paper based note-taking application on the move and then arrives at her office in order to sort through and archive the notes. Here the mobile scenario might utilize a different deployment than the nomadic scenario in the office. Additionally, the plug and play like interoperability increases the resilience of such a system: shutdown or connection loss between components can be handled much more flexibly.

Services, Data and Dataflow

At the core of the plug and play like interoperation of components are *services*, *communication channels* and *data structures* defined at the processing stage interfaces. Thereby, the mechanisms to exchange data between components need to support the pipeline based processing of digital ink, i.e., data successively travels through the pipeline. Thereby, the direction of dataflow plays an important role. While digital ink is typically pushed through the pipeline, the data on defined interactive regions, i.e., the digital representations of interactive regions, must be pulled through the pipeline by the processing components in order to allow relating digital ink to interested applications.

In order to allow processing stages *plugging* together, the mPPI processing infrastructure employs a *micro service architecture* (MSA), similar to emerging architectures for distributed, light-weight web-applications[5]. This architectural approach bases on patterns and principles commonly found in *Service Oriented Architectures* (SOA), e.g., service discovery and explicit interfaces. However, its services are much more fine-grained and employ a high degree of autonomy.

In the MSA employed in the mPPI infrastructure, a set of discoverable services forms the backbone of the system. This is combined with a *publish / subscribe* (pub / sub) system defining the decoupled, asynchronous communication channels between components to support the push based data flow characteristic for pipeline processing approaches. Thereby, services do not necessarily correspond to entire processing stages, e.g., the driver stage exposes individual services for connected pens.

Typically, exposed services provide information on available communication channels, e.g., on which channel to obtain the data of a particular digital pen. However, they also offer state data regarding particular pipeline components, or can be used to inquire about defined interactive regions or data collected within the pipeline. Predefined data structures associated with PSIs describe the data exchanged on these communication channels in order to allow for easy and convenient interoperation. Hence the processing stage interfaces consist of

[5] http://martinfowler.com/articles/microservices.html (accessed: July 2015)

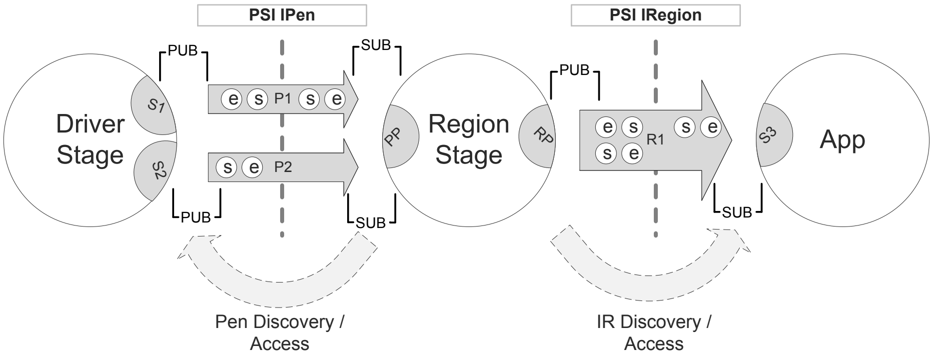


Figure 3.7: Dataflow and Services in the distributed PPI processing pipeline (example setup): Data consisting of events (e) and samples (s) is *pushed* through the pipeline on communication channels P_1, P_2 and R_1 (pub / sub). At the same time, information about active pens (S_1 and S_2), as well as defined IRs (S_3) is *pulled* as required.

- communication channels (push based), used to access data and establish data flow within the pipeline, this follows the pub / sub communication paradigm
- discoverable services (pull based), used to identify communication channels and inquire about state of interaction resources
- data structures describing the exchange format of data traveling on the communication channels

Fig. 3.7 illustrates how services, communication channels and data structures form the interaction processing pipeline.

Channels. Data is pushed through a set of communication channels using pub /sub. This dataflow starts at the pen as source to one or several applications as sinks, i.e., the applications interested in a particular interactive region on which interaction with the pen occurs. While traveling through the pipeline, the data is enriched with additional information, e.g., recognition results or clustering information. However, the original data can still be traced, it is just wrapped with additional information. This employs a topic based pub / sub communication paradigm, where the different topics are certain stages of processed data (from raw to application level), corresponding to the processing stages of the pipeline architecture.

Services. At the same time, a set of services offer pull based information on interaction resources present in the current pipeline setup, e.g., which pens are available or which interactive regions are currently defined. These services typically allow to inquire about state data of the associated interaction resources, e.g., whether a given digital pen is currently moving on an interactive region. Their main objective, however, is to identify where the data generated by these resources can be obtained, i.e., at which communication channel it will be published and thus where an interested component needs to subscribe to.

Data Structures. As described above, the basic data structures are events and samples. Events describe state changes in the course of interaction. Depending on the processing stage where they emerge, this can either be state changes regarding the pen movement, e.g., pen tip put down, or state changes regarding the recognition of continuous traces or even semantic recognition results, e.g., when a gesture recognizer detects a certain gesture. In contrast to this, samples form the actual digital ink generated by the digital pen. A sample typically consist of the x and y coordinates of the pen tip position at a certain point in time t. Depending on the underlying digital pen hardware, additional information can be included, e.g., in the reference implementation, *Letras*, the pressure applied to the pen tip while (f) is also included in the sample information (c.f., section 3.3). These data structures constitute the data that is pushed through the pipeline, i.e., the data traveling along the communication channels.

Deployment Schemes

Following the concepts outlined above, the derived mPPI infrastructure allows a plug and play like interoperability of pipeline stages at the processing stage interfaces using standard service discovery and topic based publish / subscribe communication. This enables flexible and easy distribution: As channels can provide both local and network connections, the distribution decision can be made at deployment time, or even at run time. The deployment layout, i.e., which stage is hosted on which nodes and the communication links between them (local / network) corresponds to the deployment scheme being used (c.f., section 3.1.2, *Deployment Schemes*).

Each processing stage in the resulting platform consists of a main service, that will start or stop services for its encapsulated resources as required, e.g., the services wrapping digital pen resources. A processing stage will initiate a continuous service discovery for the services encapsulating resources of interest, e.g., digital pens or interactive regions, of adjacent processing stages.

Upon discovery of such a service, e.g., a service representing a digital pen resource, the processing stage will inquire the communication channels of this resource, e.g., the channel the pen streams its data on. It will automatically subscribe to these channels

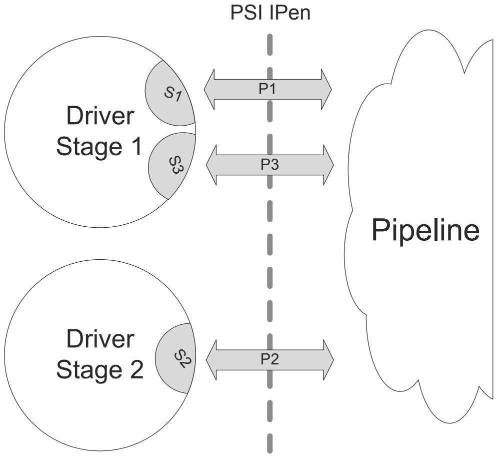


Figure 3.8: Service abstraction in the Driver Stage

and thus establish the connection between processing stages. After that it will commence operation immediately during runtime. This mechanism is therefore referred to as a *hotplug* mechanism.

With an ongoing, continuous service discovery and a hotplug connection mechanism, each pipeline stage is ready to connect to its adjacent stages during runtime. However, one further abstraction is required in order to fully utilize the deployment schemes described in section 3.1.2: processing stages require support for *multiple fan-out* and *multiple fan-in* with respect to other processing stages.

In the mPPI infrastructure this is achieved exclusively using pub / sub channels as means of communication between stages, combined with the aforementioned mechanism to detect new channels via wrapping services, e.g., a pen service offering information where its encapsulated pen resource streams its digital ink on (at which channel). An advantage of the pub / sub paradigm is thereby that it naturally supports multiple subscribers observing a particular channel and processing data received on that channel. This enables multiple fan-out.

Furthermore, using a continuous service discovery in combination with the service abstraction at the PSI level, allows for dynamic discovery of available channels at the consuming processing stage (c.f., Fig. 3.8). This, in combination with a mechanism supporting subscriptions to multiple channels simultaneously allows for multiple fan-in at the consuming side: the consuming services receives data on multiple channels (subscriptions) and processes the combined data as needed.

Following this paradigm, different deployment schemes are supported as described in section 3.1.2. Pipelines can be constructed as required by the current setting, a concept similar to the pipeline concept employed in *OpenInterface*, [Lawson et al., 2009], a generic component model to construct pipelines for multimodal interaction processing. The flexible hotplug mechanism introduced above supports a broad variety of different deployment schemes, as well as runtime re-deployment of components, tailored to the requirements of the current setting.

3.2.2 Digital Ink Processing

Following the approach outlined in section 3.2.1 for interoperability and deployment schemes, the four processing stages perform the actual processing of digital ink, coupled by their respective interfaces (PSIs). Here, digital ink is acquired, processed and transformed successively as introduced in section 3.1.

This section lays out the processing stages and their respective interfaces within the mPPI infrastructure, discusses their components and highlights the central concepts employed. The focus thereby lies on building an efficient processing pipeline as during processing of user interaction speed (responsiveness) becomes critical. Additionally, the PSIs are explained in more detail where needed.

Driver Stage: Pen Data Access

The driver processing stage accesses the digital pen hardware and provides the entry point into the pipeline. Here, the mPPI platform connects to the digital pen, receives digital ink streamed by the pen (including elementary state events) and streams that data into the pipeline. Its main objective is to provide scalable support for multiple pens and multiple heterogeneous pen models.

Hardware Abstraction The driver stage has to abstract from different pen models, i.e., different hardware and vendors, in order to allow heterogeneous pens to be used within the same pipeline - an important aspect when dealing with dynamic environments which are typical for mobile settings. This is achieved by a dynamic *driver subsystem* and a common *pen service* abstraction, essentially a generic interface for pen hardware, offering required functionality and defining a common data format for digital pen hardware. This abstraction also supports the hotplug mechanism introduced in section 3.2.1, as successive processing stages will connect to the driver stage only via its exposed pen services.

In order to actually support different pen models, the driver stages needs to employ drivers supporting the protocols required to access a particular pen model. Toward that goal, the mPPI platform employs a flexible plugin mechanism: pen drivers can be

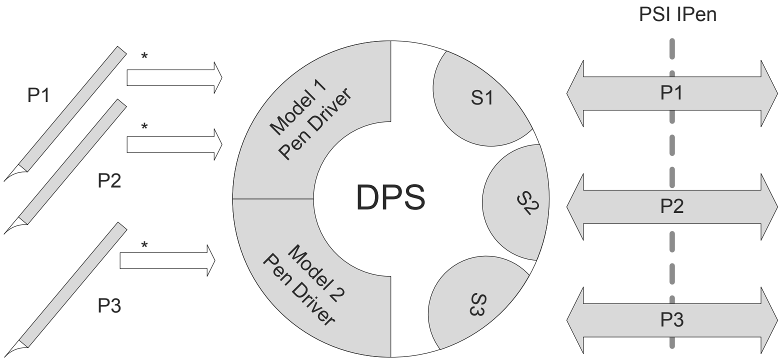


Figure 3.9: The Driver Stage supporting multiple heterogeneous pens by using the pen driver abstraction

loaded dynamically, allowing installation of a new digital pen into a running system, at the same time reducing the burden of a painful manual installation.

Pen drivers within the driver stage of the pipeline are either wrappers for drivers of the underlying operating system, or *native pen drivers* accessing and connecting to the pen hardware directly from within the platform; without requiring drivers for the pen hardware installed on the underlying operating system.

Pen Services and Scalability Consider the example setup shown in Fig. 3.9. Each pen driver instantiates a pen service for each connected pen. The Model 1 Pen Driver (e.g., Nokia SU1-B) connects to pens P_1 and P_2, and spawns the pen services S_1 and S_2. Similarly, the Model 2 Pen Driver (e.g., Logitech IO2) connects to P_3 and spawns pen service S_3. Thereby, pen services define a generic abstraction of the underlying pen hardware. This enables the infrastructure to treat different pen models exactly the same. They dispatch data collected by the digital pen to a dedicated channel for the pen, enabling topic based publish / subscribe to access recorded digital ink data as laid out in section 3.2.1. Interested entities then simply subscribe to the appropriate pen channel and listen for digital ink data published on that channel.

Following the hotplug mechanism introduced in section 3.2.1, services allow other components, e.g., succeeding processing stages, querying the connection channel and other information about pens using the pen service. At the processing stage interface (PSI *IPen*), only pen services and their associated channels are visible. Thereby, multiple pens of multiple models can be hosted by the same Driver Stage.

However, since the successive stages only notice the provided pen services and channels, it is also easily possible to deploy several Driver Stages in the environment as shown in Fig. 3.8 on page 102, while other pipeline stages still see only the sum of their connected pens. This also allows for more flexible scaling with respect to the application. For instance, this mechanism allows building applications that employ many pens simultaneously, even exceeding the limit of 8 devices imposed by the bluetooth *Personal Area Network* (PAN) specification: multiple driver stage instances could access different PANs and stream their pen's data to a single application.

Data Structures The data produced by this stage is referred to as *raw digital ink*, indicating that no further processing has yet been performed on it. Data consists of a set of events indicating the current pen state (e.g. *pen set down* or *pen lifted*) and a series of samples describing the movements of the pen on a surface. Both are published by the driver stage on a specific, unique channel associated with a particular pen. This channel can be obtained from the service wrapping that particular pen via RMC as laid out above.

Events and samples emitted vary among different pen models. Therefore each pen driver has to adapt them to the generic pen service data model used to abstract from a particular pen model. In the generic pen service data model, samples include the following information

Pen tip position (x, y) The tip position is provided as 2-dimensional coordinates x, y. In this stage, coordinates used within samples are given in *pattern space coordinates* (PSC): pen samples have absolute coordinates within the global coordinate system, i.e., the pattern space. Some pens, e.g., the Logitech IO 2 (Bluetooth), stream their sample coordinates with respect to a virtual interactive region. In these cases, the driver has to transform the coordinates of the samples back to PSC. Values are given as positive floating point values, e.g., 284.75 or 180.0.

Pen tip force (f) The tip force is provided as single, 1-dimensional value indicating the pressure applied to the pen tip while writing on paper as sensed by a pressure sensor. This is an example for additional sensoric values that might be useful during interaction. Although a broad variety of additional sensor information might theoretically be included (e.g., tilt angle) and thus is reflected in the W3C standard *InkML*[6], the pen tip force reflects the only additional data available by contemporary digital pen models.

Time stamp (t) provided as a single long integer (64 Bit) in milliseconds since the Unix epoch on midnight UTC, January, 1st 1970. This time stamp is required, as

[6] http://www.w3.org/TR/InkML/ (accessed: July 2015)

Event	Description	IPen	Nokia	Logitech
ON	Pen connected (or re-connected)	Y	N	N
OFF	Pen disconnected	Y	N	N
UP	Pen tip lifted from paper	Y	Y	Y
DWN	Pen tip put on paper	Y	N	Y
ERR	Communication error	Y	N	N
OOR	Pen out of reach	Y	N	N

Table 3.1: Events emitted by a *Letras* Pen Service

different pen models will stream data with differing sampling rates (c.f., section 2.1.2). In some cases this has to be synthesized in software, e.g., the Logitech pen does not stream any time stamps.

The pen events published by a pen service follow the pattern described in table 3.1. It lists two different pen models supported in the reference implementation of the mPPI infrastructure, *Letras* (c.f., section 3.3). Note that it abstracts from the differences between the supported pen models and synthesizes events that are not supported by the pen hardware in software.

Region Stage: Dispatching to interested Components

After having provided the raw digital ink data in the Driver Stage, the samples and pen events need to be transformed into higher-level data structures and – most importantly – related to interactive regions. This is handled in the Region Stage. This stage subscribes to input from a configurable set of pen services. In addition, it maintains a model of all currently known interactive regions as published by available PPI based applications, the *local interactive region model* (LIRM) described below (c.f., section 3.1.3 and section 3.2.3).

The processing stage interface (PSI *IRegion*) following after the Region Stage therefore describes how descriptions of interactive regions can be published and defines the interface of services allowing to retrieve the dispatch information. Interested components can obtain data per region by accessing the services wrapping interactive regions. Besides information about the layout and shape of the interactive region, these services offer a channel allowing to obtain interaction occurring on a particular region.

This concept also enables the sharing of interactive regions among different applications, as required to satisfy the requirement for resource sharing with respect to interactive regions R2.1 (c.f., chapter 2, section 2.3.1): multiple applications can connect to a single channel of an interactive region, thereby interpret digital ink according to their respective application context.

Local Interactive Region Model The region stage uses the two stage region discovery approach described in section 3.1.3. A detailed description of the mechanism is given below in section 3.2.3. Upon discovery, all available interactive regions provide details as required by the regions processing stage. This includes

- the position of the region, given as x, y coordinates of its *upper left corner* (ULC) in the global pattern space
- the dimension of the rectangular, axis-aligned bounding box of the region given as w, y (width, height)
- its geometric shape
- and the channel defined for streaming digital ink on that region

The region stage then compiles spatial relations of discovered IR into the *Local Interactive Region Model* (LIRM). It consists of a containment hierarchy tree of the regions bounding boxes as depicted in Fig. 3.10. The *bounding box* is thereby defined as the smallest axis-aligned rectangle fully containing the interactive region. The constructed tree of bounding boxes enables the regions stage to efficiently compute the interactive region in which a data sample is located by applying range checks on sample coordinates along the x, y axis. This technique is a simplified, 2-dimensional application of *bounding volume hierarchies* (BVH) which are typically used in collision detection tasks in computer graphics, e.g., in [Gottschalk et al., 1996].

Thereby, rationale behind choosing the rectangular bounding box over arbitrary shaped 2-dimensional regions in the LIRM is increased performance with respect to containment testing of pen tip positions. Most interactive regions, e.g., paper documents or screen areas, are intrinsically rectangular. Here, the geometric shape of the IR itself can be used. In special cases, where applications require differently shaped IRs, providing the bounding-box enables the pipeline to identify candidate data for containment, rather than actually contained digital ink. For these corner cases, final containment checking can easily be realized at the application side (where knowledge on the actual geometric shape of contained IRs is available).

However, fast traversal also requires the design decision to exclude overlapping regions. The rationale behind this is that typical containment trees of interactive regions

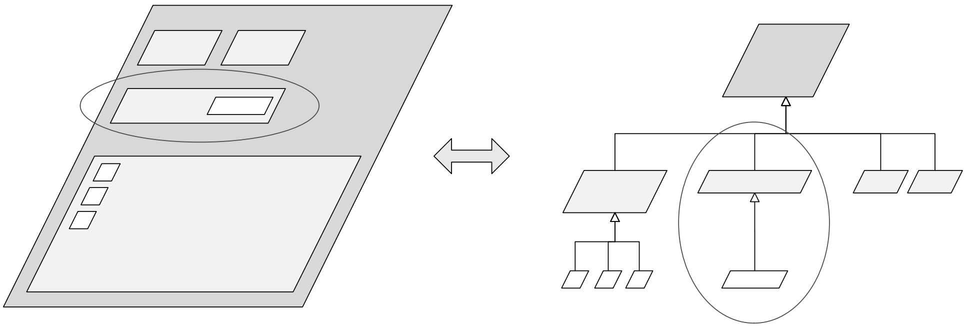


Figure 3.10: The region model: containment hierarchy of interactive regions on paper (left) is translated into a tree representing the *Bounding Volume Hierarchy* BVH (right)

are broad and shallow. First, interactive regions in most cases come with optical clues on paper, e.g., a printed button. Because of this, deep hierarchies would visually occlude the paper document. Second, there are many interactive regions besides each other in many applications, e.g., the pages in a book. So accelerating traversal at the sibling level becomes important.

Additionally, despite having worked in the field of PPI based application development for nearly 5 years, the author could not come up with a single real- world use case exemplifying overlapping interactive regions. While this statement of course does not add scientific credibility to the argument, it still indicates that overlapping regions are an academic corner case at best. It is however possible to handle these cases: the application simply defines the top-level interactive region and handles the dispatching to contained (overlapping) interactive regions in a custom implementation[7].

Digital Ink Dispatch The region stage receives raw digital ink and dispatches it to the channels of the interactive region(s) containing the samples using the bounding volume hierarchy of the LIRM. In order to compute the interactive region containing the x, y coordinates of a given sample, the LIRM tree is traversed in a pre-order, fail-fast scheme. On the one hand, *fail-fast* here refers to the fact, that the containment relation between parent and child nodes, i.e., every child node is fully contained within it's parent node, allows ceasing traversal on the parent level iff no node containing the sample is discovered. On the other hand, it also also refers to the fact that whenever an interactive region has been found containing a sample, the traversal of siblings can be stopped at this point. This considerably accelerates the traversal process.

[7] should the reader ever encounter such a case, the author would be delighted to learn about this

Upon identification of the interactive region containing digital ink, the region stage dispatches the digital ink to the channel associated with that region. Because of the high spatial proximity of sample series (the samples on one pen stroke obviously lie close together), a simple *least recently used* (LRU) caching system allows boosting the dispatch performance. This is required to be able to handle the data rates of digital pens which can be up to ∼100Hz (c.f., section 2.1.2).

In nested region hierarchies, there are cases where more than one region might be interested in the digital ink. Consider as an example an area on a paper sheet allowing the user to enter gestures in a sub-portion of the region. This setup is depicted as the highlighted part of the region model shown in Fig. 3.10. Here, not only the region itself, but also its parent region(s) need to receive digital ink streamed on the region, e.g., for rendering purposes. In order to support the dispatch to interested parent regions, the mPPI platform introduces a `HUNGRY` flag. The parent region indicates its interest in digital ink on its child regions by setting this flag. The region stage then dispatches to all regions upward in the containment hierarchy of the LIRM having set this flag.

Should no interactive region in the LIRM contain a given sample, e.g., if the user starts interacting with an unknown document, the discovery mechanism for unknown interactive regions as described in section 3.2.3 will be triggered.

Data Structures Data at this stage is *region related digital ink*. Here, samples received in the driver stage are aggregated into *traces* on the pattern space surface. These can later be further aggregated. The organization of digital ink data structures in the platform follows the *InkML* standard: digital ink essential forms a tree. Samples constitute the tree leafs. Traces consisting of samples form the next higher level. Ink structures defined in later processing stages group traces or other ink structures, reflecting semantic relations among recorded data, with one ink structure (the top-level structure) forming the root of the tree.

Traces are basically movements of the pen on paper. Generally speaking, a trace is everything that the pen records between putting the pen tip on the surface, i.e., a `DWN` event being emitted by the driver stage, and lifting the pen from the surface or any other event canceling the interaction, i.e., a `UP`, `OFF` or `ERR` event. After having detected these ink data structures at the trace level, the region stage dispatches samples along with selected pen events and events indicating the trace data structures to the interested parties using a dedicated channel for each region (again employing topic based publish / subscribe approach).

In contrast to the driver stage which emits all data on channels associated with specific pens, the region stage emits data on channels associated with interactive regions: each IR description defines a channel where the digital ink generated by all connected digital pens on this particular region will be published. Thereby, the channel is defined by the region service advertising that region's information, i.e., an application can define which channel it wants to use to receive data for a particular interactive region. Usually, the region stage will dispatch data only on to a single channel. However, in the case of parent regions having set the `HUNGRY` flag, it can be dispatched to multiple cascading regions (and thus to multiple channels).

The region stage emits specific events that indicate the start (`TST`) or end (`TND`) of traces, as well as `UP` and `DWN` events. If a trace crosses the boundary of an interactive region, the trace itself might not end. However, as the current region cannot "see" the trace anymore[8], a `TRACE_END` event is emitted even if the pen is not lifted from the surface. This is then followed by a `TRACE_START` event on the new region.

In order to make it possible for the application to identify both traces and their events as a single trace crossing two regions, these events bear a special flag, the so called `CONTINUES` flag. This flag indicates that the trace ended only on a given region, but actually does continue further. To ease identification, the trace events on both regions and the samples share a common *universally unique identifier* (UUID). This allows associating two traces even if their samples are received on different interactive regions, i.e., on different channels.

Samples carry the same data as in the *driver stage*. Coordinates within samples, however, in this stage are given in *normalized region coordinates* (NRC). This means that $\forall x, y . x, y \in [0, 1]$, normalized against the enclosing (rectangular) interactive region, where the original coordinates in PSC can be obtained as

$$x_{PSC} = x_{IR} + x * w_{IR}$$
$$y_{PSC} = y_{IR} + y * h_{IR}$$

With $x_{IR}, y_{IR}, w_{IR}, h_{IR}$ being the x, y position, the width and height respectively, of the interactive region. This makes rendering of samples and other task regarding scaling etc. less complicated.

Additionally, all samples contain their original coordinates within the global 2D coordinate system, i.e., their pattern space coordinates. This is required, because otherwise the non-uniform scaling that might arise out of non-quadratic interactive regions, might skew recognition results at a later stage where the interactive region data itself might not available, e.g., at the semantic stage.

Finally, each sample and event carries an identifier of the digital pen originally producing the data. This is important, as digital ink from all connected pens will be received on the channel of the IR it occurred on. Without a pen identifier, applications supporting input by multiple digital pens could not distinguish the data sources.

[8] unless the `HUNGRY` flag is set and the second region is a child region to the first region

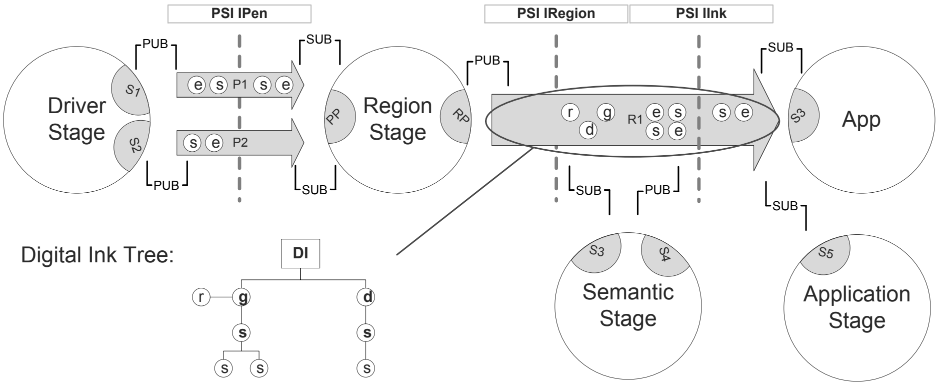


Figure 3.11: Publish / Subscribe Channels in the Pipeline: The Semantic and Application Stages subscribe to the Channel of an Interactive Region and inject Recognition Events

Optional Stages: Semantic Processing and Application Services

The next stages in the processing of digital ink are adding interpretation to samples and events occurring on interactive regions (*Semantic Stage*) and providing applications with the means to offer PPI based user interfaces (*Application Stage*). Thereby, applications themselves define which components they need, e.g., a simple drawing application mainly using Pidget interactors will not need a gesture recognizer, whereas a PPUI based word processor as in [Liao et al., 2008] will require a sophisticated segmentation and gesture recognition engine.

To reflect this diversity and flexibility, both optional stages, i.e., the semantic stage and the application stage employ a service oriented, highly configurable component model based on a *micro service architecture* (MSA). Combining services as needed, applications can construct individual low- level pipelines according to their specific requirements.

Toward this end, it is possible to discover available services in the environment, or to instantiate required services in case such a service is missing (or exclusive access to a service's functionality is required); processing stages thereby serve as deployment containers for required services and provide common abstractions employed by the services, e.g., connections to receive and process digital ink. Instantiated services can be configured to restrict their processing either to a single application, or allow sharing of their processing capabilities. However, services shared at this stage that are not inherently *stateless* must be carefully designed to avoid side-effects.

Semantic Stage In this stage, each service is capable of taking the channel of an interactive region as input to receive samples and events as defined by the *IRegion* interface. It then performs its recognition and injects the recognition results into that channel. For instance, a gesture recognition service receives events and samples produced on an interactive region IR on its associated channel C_{IR}. Triggered by trace events it stores pen traces and upon trace completion (or even during drawing depending on its implementation) triggers recognition. Recognition results are sent as *recognition events* on C_{IR}, e.g., as n-tuple containing probabilities for a gesture set. This mechanism of injecting events into C_{IR} is shown in figure 3.11.

Thereby, the Semantic Stage consumes data structures and services as defined by the *IRegion* interface. This interfaces revolves around dedicated communication channels for all interactive regions and a set of data structures for digital ink that describes the digital ink tree at the level of *trace* nodes. The interface offered by the Semantic Stage, i.e., the PSI *IInk*, builds on the same principle.

However, it extends the digital ink tree in a similar bottom-up approach as applied in the Region Stage (which clustered the received samples into trace elements): the semantic stage clusters the received *traces* into *digital ink* elements. It thereby attaches descriptors, e.g., XML or JSON Strings, to those elements. These descriptors specify semantic for digital ink elements. For instance, a segmentation service would label a digital ink element as ”drawing”, ”text” or ”gesture”, whereas a gesture recognition service would add concrete recognition results, e.g., the n-tuple mentioned above. The recognition events thereby essentially serialize these data structures on the region's channel C_{IR}.

This flexible approach allows offering a broad variety of services. Although a broad variety of semantic processing services is imaginable, highly recommended and in the reference implementation described in section 3.3 actually implemented services at this stage are

- a generic and highly configurable segmentation service used to split digital ink structures into semantically related units (based on clustering algorithms)
- a generic interaction processing service based on the conceptual framework introduced in chapter 4
- gesture and handwriting recognition services

Wrappers for external services, e.g., the iGesture toolkit [Signer et al., 2007a], the segmentation framework introduced by Ispas et al. [Ispas et al., 2011] or the Microsoft handwriting recognition service can be easily integrated into the pipeline at this stage.

Application Stage Digital ink data and the results of semantic processing are interpreted by an application and mapped to the application's functionality. This cannot be part of the infrastructure. However, some tasks are common among applications, e.g., the storage of digital ink in a document database, or the rendering of digital ink onto a GUI. Hence the application stage provides support for this common type of functionality and offers some convenience services. Suggested services at this stage include

- a persistence service capable of storing digital ink data
- a generic digital ink rendering service
- an interaction buffer for temporary connection loss with the application in mobile settings (c.f., section 3.3.2)

Thereby, the persistence service, allows storing digital ink in a tree mapping to document structure as described above and in section 3.2.2 (based on the *InkML* standard). Samples form the leaves of the tree, grouped by traces. These are grouped by arbitrary digital ink structures that can possess attributes to attach semantic interpretation, e.g., gestures or clustered parts of a document. The root of the tree is typically a node corresponding to a document (or a set of multiple documents). The tree is stored in an object database and can be retrieved by applications for further use.

Similarly, the rendering service supports being attached to region channels in order to render the digital ink on a digital display. It uses standard spline interpolation employing the sample points as support points, i.e., Catmull-Rom spline interpolation [Catmull and Rom, 1974]. This allows drawing smooth curves fitting around the samples that resemble natural digital ink.

3.2.3 Interactive Region Publishing and Discovery

Publishing and discovery of interactive regions is a crucial aspect of supporting document mobility. As described in section 3.1.3, publishing and discovery of interactive regions essentially forms the problem of constructing and sharing the interactive region model. In the distributed pipeline, this follows a 2-level approach treating local and remote knowledge of interactive regions differently[9]. Thereby, the 1st level corresponds to construction and sharing of the *Local Interactive Region Model* (LIRM) within the region stage, while the 2nd level provides maintenance of and access to the *Interactive Region Naming System* (IRNS).

[9] local means here either on the same logical node or on another node in the local subnet, whereas remote refers to knowledge only available accessing the Internet

1st level: LIRM construction and sharing

Locally available knowledge of interactive regions is fully synchronized between instances of the region stage. This means, every instance of the regions stage constructs its own LIRM containing all locally available interactive regions. As described in section 3.2.2, the region stage issues a continuous service discovery to detect interactive regions advertised in the vicinity of the user. Then, it constructs the containment hierarchy forming the LIRM out of these regions. The continuous service discovery being used for all available interactive regions implies that local model construction is done proactively, i.e., without requiring any digital ink being generated on that particular region. This means, locally deployed applications will be usable without any additional delay.

The publishing process for interactive regions consists of advertising its wrapping service in the local subnet by using standard service discovery mechanisms, e.g., by broadcast advertisement in the local area network. Upon discovery, the region stage will inquire further details about the region as described above (bounding box, channel) via *remote method calls* (RMC). However, in case of applications defining a huge number of interactive regions, the round-trip times of the RMC will add up, which might result in increased network traffic and long delays, particularly if there are several region stage instances active within the same subnet.

Therefore, another concept is required to handle this: *bulk region transfer*. This concept enables an application to transfer not only a single interactive region, but to transfer a set of interactive region data in bulk to the region stage, i.e., a whole subportion of the region model. Here the round-trip time of the RMC will only occur once, while a larger amount of data will be transferred. Which method is best depends on implementation details, however, the bulk region transfer introduces additional flexibility to the individual components.

2nd level: The IRNS

Local applications, hosted in the vicinity of the user, essentially are applications hosted in an organization's intranet: publishing and discovery of these applications are handled intra-organizational, i.e., by synchronizing LIRM as explained above. However, for inter-organizational publishing and discovery, the infrastructure has to refer to the global 2nd level of publishing and discovery. As introduced in section 3.1.3, infrastructure based on the distributed PPI processing pipeline handles this via the *Interactive Region Naming System* (IRNS).

Knowledge on interactive regions that are not advertised locally, is fetched using the IRNS to provide bindings of interactive regions to applications. Toward that end, each region stage instance has a pre-configured reference to one node in the IRNS. As described in section 3.1.3, the IRNS is designed as a 2-dimensional P2P distributed hashtable, i.e., a *content addressable network* or CAN [Ratnasamy et al., 2001], to provide fast and reliable publishing and discovery of interactive regions.

Whenever the region stage receives samples which cannot be related to any interactive region in the LIRM, it triggers global *discovery* over the IRNS. The goal thereby is to determine which, if any, interactive regions sign responsible for receiving the digital ink and to which channels it should be streamed. To ease future application uses, discovered interactive regions are integrated into the local LIRM upon discovery in order to accelerate the dispatch process.

This approach requires a handler mechanism to ensure that the LIRM does not turn stale, i.e., has an outdated part of the region hierarchy. For instance, a new application using a sub-partition of an interactive region formerly used by another application might advertise this new IR globally; however, as long as the former IR remains part of the LIRM, the new IR cannot be discovered. In the mPPI infrastructure this this problem is overcome by using *time-to-live* (TTL) annotations in the region model for non-local regions, similar to the approach in the DNS. If the TTL is exceeded, the region stage drops that part of the LIRM and issues another discovery if required.

This approach toward remote application access implies an application paradigm similar to the Internet. Remote applications can only process digital ink as long as the client is connected. In mobile settings, however, connection loss might render remote access temporarily unavailable. While this is unproblematic for some applications, e.g., simple access to remote information by clicking on a paper map as in *EdFest*, [Signer et al., 2006]; other applications, e.g., mobile note-taking solutions, might lose important data. In these cases, it is therefore possible to dynamically deploy a remote application locally, e.g., by automatically downloading application code upon discovery and instantiating the application.

Publishing in the IRNS follows a straight-forward approach: local interactive regions are handed over to the responsible node in the IRNS by copying the relevant sub-tree of the LIRM. However, in practice, some applications might choose to support local access only, or to re-use and thus re-define interactive regions associated to globally available applications. In these cases, an IR can be flagged as `local-only`. This will cause that particular part of the LIRM to not be forwarded to the IRNS, while keeping it for local use. This re- use of interactive regions is an important feature and unique to the mPPI infrastructure introduced here.

3.3 Reference Implementation: The Letras Platform

This section introduces the *Letras* platform for mPPI processing. It demonstrates how the concepts introduced in section 3.2 can be employed in a generic platform supporting PPI and mPPI. By offering a *reference implementation* of the approach, it presents the first step toward a *proof-of-concept* evaluation of the concepts introduced throughout this chapter. Thereby, the foundation of the infrastructure is discussed first. This foundation supports distributed deployment and is designed to work on both, mobile and non-mobile platforms. Subsequently, a special deployment of the system for mobile platforms is introduced: *MobiLetras*. It adds functionality specifically designed to solve issues occurring in mobile settings, e.g., connection loss and resource management.

3.3.1 Generic Distributed Deployment

Letras provides an extensible platform for rapid development of (mobile) PPI based applications. It empowers application developers to base on a common infrastructure and processing model. *Letras* is designed to support all pipeline stages, from acquiring digital pen data at the hardware interface (driver stage) to higher-level semantic processing needed by specific applications (application stage), supporting lightweight protocols and communication mechanisms. *Letras* also offers an application model based on interactive regions and associated functionality, that supports both local and remote applications.

Thereby, *Letras* employs a gray-box framework approach: application developers can construct instances of the mPPI processing pipeline using a set of predefined components which can ”plug together”; or developers can choose to extend and customize components if required by the targeted setting. In addition, highly flexible deployment schemes as described in section 3.2.1, allow for easy adaption to the specific requirements of the environment at hand.

A ready to use, platform independent implementation of *Letras* is available for download[10]. The implementation supports the development of mobile PPI based applications on Microsoft Windows, Mac OS X and Linux operating systems. Due to the flexibility and small footprint of the underlying ubiquitous computing middleware, it is also possible for *Letras* to integrate applications and processing stages on resource constraint devices, such as smartphones, tablets etc.

Required adaptions of the base architecture for the mobile platform use case are described en detail in section 3.3.2. As reference implementation of *MobiLetras*, the mobile prototype of the *Letras* platform has been developed for the *Android*[11] operating system for smartphones (version 2.1 and above). It has been deployed and tested on a Motorola Milestone mobile phone as well as Samsung Galaxy S and Nexus S smart- phones.

[10] https://github.com/fheinrichs/letras (accessed: July 2015)

[11] http://www.android.com/ (accessed: July 2015)

Underlying Middleware

Implementing a *plug and play* approach toward flexible connection of pipeline stages, as laid out in section 3.2.1, is ideally based on a flexible and lightweight middleware. This reduces the amount of custom development for problems not specific to mPPI infrastructures, e.g., network transparent communication and service discovery. Such a middleware needs to provide suitable communication abstractions and supporting network transparent coupling of processing stages. Several such approaches exist in the domain of pervasive computing, e.g., *Gaia* [Román et al., 2002] or *MundoCore* [Aitenbichler et al., 2007][12]. As such, the middleware is not part of the actual mPPI infrastructure, it rather exists as an independent foundation.

Multiple ready-made middleware solutions exist. The criteria a middleware needs to support in order to serve as foundation for the mobile PPI infrastructure based on the distributed processing pipeline are

- network transparent communication both *Remote Method Call* (RMC) based to construct client / server links, as well as *Publish/Subscribe* (pub / sub)
- topic based pub / sub communication using unique, named communication channels (topics)
- services and a service discovery mechanism allowing for construction of *service oriented architectures* (SOA) and *micro service architectures* (MSA)

Based on such a middleware, applications in *Letras* can then either detect deployed processing stages in their environment using service discovery, or provide their own stages. Connections between processing stages, as defined by the respective processing stage interface, rely on a completely transparent communication link: it does not matter, whether the connection is local, or on a remote device.

In order to realize this, *Letras* is constructed on top of the ubiquitous computing middleware *MundoCore*, [Aitenbichler et al., 2007]. It offers topic based *publish / subscribe* (pub / sub) and *remote method call* (RMC) communication. In addition, *MundoCore* offers a highly flexible component model based on services: each logical node in the *MundoCore* overlay network can host a number of services, while other nodes can discover these services, either deployed on the same logical node, or on any other node in the currently configured overlay [Aitenbichler et al., 2007]. *Letras* bases on this highly flexible, service oriented component model of *MundoCore*. Following this approach top-level components (processing stages) can simply be (re-)used or replaced according to the needs of the application at hand.

[12] for a survey of middleware approaches refer to [Raychoudhury et al., 2013]

MundoCore supports language independent communication channels between logical nodes in the *Mundo* overlay network. Ports for multiple programming languages exist in *MundoCore*, making it available for broad variety of platforms. This flexibility holds for *Letras* also. It is easily possible, for example, to develop a C# based application, which uses a Java based processing pipeline. *Letras* itself is developed in the Java programming language, although, due to the flexibility of the underlying *MundoCore* middleware, any stage in the pipeline can be replaced by an appropriate stage in any other programming language for which a *MundoCore* port exists[13].

Deployment Schemes

The underlying *MundoCore* middleware offers the concept of *nodes*, each of which hosts a set of *services* communicating via pub / sub *channels*. Effectively, even the provided RMC functionality is emulated over pub / sub channels. Service discovery is available, both on the same node, i.e., the same physical machine, or within the overlay network of *Mundo* nodes. Communication is handled network transparent, i.e., the middleware enables nodes to communicate either over a network transport or locally without exposing any additional complexity to the PPI infrastructure. In the local case, special optimized communication is used. A more detailed overview of the *Mundo* architecture can be found in [Aitenbichler et al., 2007]. For details regarding concepts and implementation please refer to [Aitenbichler, 2006].

Plug and play interoperability as described in section 3.2.1 to realize different deployment schemes uses the *MundoCore* continuous service discovery function to implement the hotplug interoperability between processing stages. Here, services wrap resources, e.g., digital pens, and offer communication channels to receive data generated by these resources, e.g., digital ink. Communication channels are realized via *MundoCore* channels, allowing for topic based pub / sub. Additionally, the *MundoCore* pub / sub channels allow multiple fan-out and multiple fan-in via subscription to a certain channel. This makes a scalable and flexible composition of interaction processing pipelines possible.

Each processing stage is wrapped in special *Mundo* service that starts all required services and monitoring processes of that stage. As such, it is only possible to run one instance of each stage per node. Therefore, the deployment scheme is determined by which *Mundo* nodes are started in the local network and by which processing stages are configured to be run on each node. *MundoCore* nodes can be configured to run certain services in a specific XML configuration file, the `node.conf.xml`. Hence, deployment schemes are determined by settings in the configuration files of the nodes available in the *Mundo* overlay network.

[13] Including Java, C++, C# and various flavors of JavaME

Distributed mPPI Processing Pipeline

As described above, each processing stage consists of a wrapping service, that starts or stops services for associated resources such as digital pens, interactive regions, semantic processors and data stores or rendering components. These services are themselves not discoverable, only the resource wrapping services they expose are discoverable by other pipeline stages.

Driver Stage The driver stage wraps *MundoCore* plugins containing drivers for different pen models. Each driver handles a type of digital pen and will spawn a new *PenService* wrapping each digital pen that is discovered. Drivers can be deployed in a *hotswap* fashion, that is, bundled plugins for new drivers can be installed into a running driver stage. Drivers could be wrappers accessing OS drivers or native drivers, i.e., drivers developed directly within *Letras*. As an example, *Letras* supports three native pen drivers:

- the *Nokia SU-1B*
- the *Logitech IO2 (Bluetooth)*
- and the *Anoto ADP-301 A*

The Nokia and Logitech drivers are cross platform drivers developed in pure Java where communication with these pen models is handled via bluetooth [14]. This is possible, as both pen models stream their data using the bluetooth *Serial Port Profile* (SPP) [15] on dedicated bluetooth channels as soon as the pen has been paired with a device. They differ only in the channels they stream on and in the reference coordinate system of the digital ink provided, so their drivers have to provide an appropriate abstraction here.

The ADP-301 pen, however, uses the *Human Interface Device* bluetooth profile (HID), and requires a handshake to setup the pen connection before starting to stream its data. This involves operating system specific code as most operating systems intercept HID device connections, e.g., to support wireless keyboards and mice. As a result, part of the driver is C++ based and thus different ports for different operating systems exist[16].

[14] based on Bluecove, a JSR-82 wrapper library for various native bluetooth stacks (http://bluecove.org/)

[15] https://www.bluetooth.org/en-us/specification/adopted-specifications (accessed: July 2015)

[16] MS Windows, Mac OS X and Linux

Region Stage The implementation of the region stage follows the principles outlined in section 3.2.2. It uses a *MundoCore* continuous service discovery to discover all pen services independent on which driver stage they are hosted. At the same time, it uses a continuous service discovery to detect available interactive region services (i.e., services advertising IR descriptions) in the local *Mundo* overlay and constructs the bounding volume based LIRM out of these. Whenever digital ink is received on any of the channels of discovered pen services, this digital ink is dispatched to the channels of interactive regions containing it, if any.

Implementation is available in pure Java, as well as in the Android subset of Java. Here, it was important to not base geometry computation on the AWT[17] classes, as these are not existing on the Android platform. Hence, geometry functionality for rectangles, containment testing etc. had to be implemented within *Letras* itself to avoid different branches of code for the mobile / desktop versions within the pipeline itself.

Semantic and Application Stage Both semantic and application stage are also written in pure Java. The semantic stage of the reference implementation includes a configurable segmentation service based on simple geometric as well as temporal proximity, a generic interaction processing service to interpret interaction techniques based on the conceptual framework of interaction introduced in chapter 4 allowing to quickly implement different interaction techniques, as well as several gesture recognition services, e.g., the 1$ gesture recognizer [Wobbrock et al., 2007] and a custom geometric recognizer for pie based gestures.

The application stage of the reference implementation stores the digital ink tree as produced by the region stage and the semantic stage (e.g., through clustering). It offers a persistence service based on *MongoDB*, a no-sql database[18], to store the digital ink data tree. On Android devices, a SQLite based implementation is used instead. Additionally, it includes two different rendering services: a Java Swing based implementation for desktop applications and an Android version for use with the mobile platform based on Catmull-Rom spline interpolation [Catmull and Rom, 1974].

Interactive Region Publishing and Discovery

As described in sections 3.1.3, 3.2.3 and 3.2.2, *Letras* uses the two level interactive region and discovery approach. This is implemented using the *MundoCore* service discovery functionality within the local *Mundo* overlay network in the region stage of the pipeline. Currently, the reference implementation of *Letras* supports only local

[17] http://docs.oracle.com/javase/7/docs/technotes/guides/awt/index.html (accessed: July 2015)
[18] http://www.mongodb.org/ (accessed: July 2015)

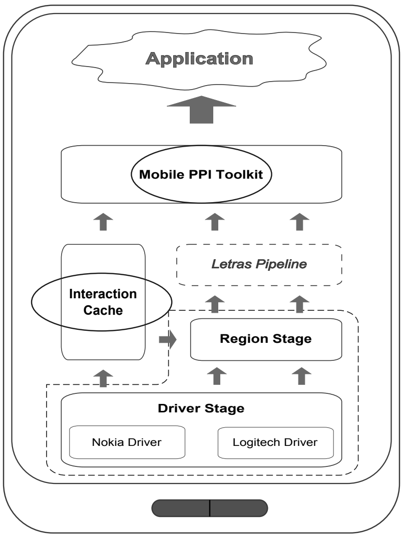


Figure 3.12: Mobile platform support for the distributed PPI processing pipeline

region publishing and discovery to the full extent. However, this allows deploying and testing all PPI based applications on a local scale for real-world test cases.

Although the IRNS has been conceptualized as a CAN based DHT (c.f., section 3.1.3), *Letras* skips actual implementation of this concept as the underlying *MundoCore* middleware offers a ready-made P2P service discovery functionality: a Tapestry [Zhao et al., 2001] based DHT. This mechanism can be used to discover remote region stage instances (and thus their LIRMs), which maintain non-local parts of the global region model.

3.3.2 Mobile Deployment: MobiLetras

As demonstrated in chapter 2, section 2.3.1, supporting a *mobile platform* constitutes an essential requirement to the mPPI infrastructure in order to fully support mobile usage practices. On the one hand, it should be possible to interact with paper documents on the move, on the other hand, it should be possible to use the mobile device for displaying dynamic information related to mPPI and access required backend functionality. Furthermore, the mobile device could be used in interaction techniques integrating mPPI with its own interaction capabilities.

Thus, tailoring the distributed interaction processing pipeline to allow deployments on mobile platforms becomes crucial. Fig. 3.12 depicts the generic architecture of mobile deployment of *Letras* on mobile platforms. It builds on top of the distributed PPI processing pipeline and extends it for supporting mobile applications. Red circles in Fig. 3.12 show the extensions to *Letras*, the dashed line contains the specific deployment of *Letras* for mobile devices.

Deployment considerations. In the mobile platform setting, deployment of the first two processing stages on the mobile device saves bandwith: as samples can be streamed directly from the driver stage to the region stage, the amount of messages send over network channels will be reduced by half. Deployment schemes using the mobile platform setting should always strive use this mechanism as much as possible to save battery and network bandwith. For other processing stages, e.g., the computation intensive semantic stage, the application developer has to consider the tradeoff between network bandwith utilization and computational complexity.

In order to support user mobility, i.e. users taking paper documents with them and interacting on the move, the infrastructures also needs to handle temporary connection loss. Depending on the application type and its degree of interactivity, this could render remote applications temporarily unusable. The mobile deployment solves this for applications demanding a high degree of interactivity by deploying the applications, or *application proxies*, directly on the phone after discovery of the application. This is part of the *Mobile PPI Toolkit*[19].

However, in case of applications that have not yet been discovered, e.g. in situations where the user encounters an unknown document, or applications that do not require a high degree of interactivity, e.g., a simple paper based note- taking application, the pen data cannot respectively does not have to be processed immediately. In these cases the data is stored in an *Interaction Cache* and can be fed back into the pipeline as soon as the application has been discovered or becomes available, i.e. when the network connection is re- established.

3.4 Evaluation

The concepts reported in the present chapter were evaluated in order to asses the benefits of the mPPI infrastructure presented, its support for mobile usage practices, modular distributed design and the mPPI processing model used as conceptual underpinning. Toward this end, a qualitative analytical approach in combination with a *proof of concept* evaluation was chosen. The evaluation comprised a detailed analysis of the concept of a distributed PPI processing pipeline in the context of supporting mobile PPI. Thereby, the approach was compared with the monolithic pipeline design employed in existing approaches. The analysis was based on the requirements established in chapter 2, section 2.3.1. It demonstrated that the distributed processing

[19] for the other parts supporting interaction techniques see chapter 4 and interactors defined in the conceptual framework

pipeline allows supporting both, *user mobility* and *document mobility*, at the conceptual level better than existing approaches. It thus provides a suitable underpinning for mobile PPI infrastructures.

Furthermore, the concept of the distributed PPI processing pipeline as enabling factor for mobile PPI and as infrastructural support for PPI in general was established using a *proof-of-concept* approach: a reference implementation of the mPPI infrastructure based on the distributed PPI processing pipeline, *Letras*, in combination with a set of prototype applications developed on-top of the infrastructure. This also demonstrated practical feasibility of the approach.

Applications thereby span various use cases, including mobile and stationary settings. On the one hand this demonstrates that the novel mPPI infrastructure based on the distributed PPI processing pipeline enables both, mobile PPI and stationary PPI. On the other hand, it exemplifies the ease of developing different applications using the infrastructure. This validates the argument of the presented infrastructure concept *enabling mobile PPI*.

Exemplifying the practical relevance of established concepts, a case study of the implementation of the *digital grocery list* application introduced in chapter 1 concludes the evaluation.

3.4.1 Analytic Comparison: Monolithic vs. Distributed Pipeline

This section provides a detailed analysis and demonstrates why the standard, monolithic processing pipeline used in existing infrastructures introduces conceptual problems with respect to mobility requirements derived in section 2.3.1. In comparison to this, it analyses how the novel mPPI infrastructure introduced throughout this chapter addresses these requirements and shows that using the distributed PPI processing pipeline as conceptual underpinning enables use of PPI in the mobile domain.

R1: Support for User Mobility

Support for user mobility (R1) refers to support for PPI in situations where the user is on the move. This includes nomadic settings, e.g., at the office desk, and mobile situations, e.g., in the train or while walking. As derived in chapter 2, section 2.3.1, supporting user mobility requires the infrastructure to support the interactive mode in mobile settings (R1.1), a mobile platform (R1.2) and a flexible deployment to adapt to changing environment conditions (R1.3).

R1.1: Interactive Mode While the monolithic processing pipeline in principal supports using the *interactive mode*, the lack of means for resource sharing and interactive region discovery forces applications to use a client/ server based approach for the mobile domain (as described in the next section). Here mobile devices can only serve as proxies accessing the digital pen hardware, while sending all data to an application server hosting the pipeline deployment. This prevents the interactive mode from being used on mobile devices where instant feedback on user actions is a central requirement.

The distributed pipeline supports the *interactive mode* just as does the monolithic pipeline. However, the flexible concept of resource sharing among multiple pipeline instances allows using fully distributed systems instead of either monolithic deployments on the mobile device, or client server based approaches. This makes it possible to deploy required stages, i.e., those components critical to provide instant feedback to the user, on the mobile device without adding to the distribution complexity. As a result, employing the interactive mode on mobile devices becomes easily possible via the distributed processing pipeline and its concept of shared processing resources.

R1.2: Mobile Platform Existing infrastructures are designed for the stationary use case. As a result, the sheer size of the monolithic deployment using the generic pipeline does not allow for a deployment on some mobile devices, i.e., it inherently does not satisfy the *mobile platform* requirement. However, with increasingly powerful mobile devices available, this limitation becomes purely a matter of implementation. The only conceptual limitation of the monolithic approach is the distribution of applications, i.e., the mobile device would need to host all applications and the user would have to install them manually. This cumbersome manual step contradicts the flexible and light weight style of interaction that makes paper such a convenient medium.

The concept of required and optional processing stages enables applications to only use those processing components which are actually needed, while remaining flexible if additional services are required at runtime. This is supported by the PSIs, as required data can be obtained at any processing stage interface and thus at any stage in the pipeline. This allows considerably reducing the footprint of the PPI processing infrastructure providing ease for the development of *mobile platform* ports of the infrastructure. Furthermore, the distribution of applications becomes convenient and flexible due to the plug and play like interconnectivity of processing stages via PSIs. This avoids the cumbersome manual setup of new applications on the mobile device and allows a flexible and lightweight style of interaction with paper documents on mobile platforms.

R1.3: Flexible Deployment The most important problem of the monolithic deployment of the generic PPI processing pipeline with respect to user mobility is the lack of support for *flexible deployment*. The monolithic deployment does not allow for distribution of processing intensive tasks, e.g., handwriting recognition. However, such options are required in the mobile domain as discussed in section 2.3.1, for instance to save battery. Furthermore, the monolithic deployment forces all components to be hosted on a single processing entity. There is no dynamic redeployment that could cope with problems arising out of the mobile setting, e.g., temporary connection loss. Hence the monolithic set-up severely limits the deployment flexibility, required to support user mobility in changing environments. A more flexible approach is required.

In contrast to this, *Flexible deployment* and sharing of resources form the basic concepts of the distributed processing pipeline. This enables mPPI based applications to distribute processing intensive tasks, e.g., handwriting recognition. Distributions thereby bases on flexible deployment schemes, as shown in the first example in Fig. 3.3 a. on page 91. This mechanism can be used to save battery. Moreover, the deployment flexibility inherent to the distributed pipeline allows for dynamic redeployment of processing stages, e.g., in the case of temporary connection loss to a backend server.

The distributed PPI processing pipeline supports this by defining clear processing stage interfaces and flexible ”wirings” between the processing stages. Using the *hotplug* mechanism based on the service discovery mechanism of the underlying middleware it allows discovering available processing stages and adapting the pipeline layout dynamically. Connections are handled completely network transparent, that is: it is not important whether a processing stage is hosted locally, or is distributed in the network. Hence truly flexible deployments can be achieved.

R2: Support for Document Mobility

Support for document mobility (R2) refers to support for PPI in situations where the paper artifacts are mobile. This includes encountered documents, e.g., where the user receives a document, and passed on, or disseminated documents, e.g., interactive leaflets send out by a supermarket chain. As derived in chapter 2, section 2.3.1, supporting document mobility requires the infrastructure to support resource sharing (R2.1) and interactive region discovery (R2.2).

R2.1: Resource Sharing The most severe limitation of the monolithic deployment is its intrinsic lack of *resource sharing*. In a monolithic approach, a resource is associated with a single pipeline instance. Apart from being a waste of resources, this setup makes it difficult to support sharing of resources. Hardware resources become blocked by a pipeline instance. This prevents, e.g., using the same digital pen in multiple applications simultaneously. A pen connection usually requires manual set-up (i.e., bluetooth pairing of the pen). Therefore switching a pen to another pipeline instance is slow, cumbersome and error prone in the monolithic setup. Although manual setup cannot be circumvented, the monolithic deployment aggravates this situation by ”blocking” the pen resource, thus effectively locking the pen within a single pipeline deployment, i.e., an application or a collection of applications from the same distributor.

To alleviate this problem, some monolithic infrastructures, e.g., *iServer / iPaper*, [Norrie et al., 2006a], and *PADD*, [Guimbretière, 2003] (with respect to interactive region discovery), allow using a client / server based approach. In this approach all applications are essentially hosted on a central server and the pen data is simply dispatched to that server. However, this approach introduces limitations with respect to scalability as the central server provides a performance bottleneck. In purely document centric applications with a low degree of interactivity, this solution might hold. However, the performance penalty obviates using the interactive mode of the pen in mobile settings (c.f., discussion of *user mobility* above).

In contrast to this, *sharing resources* between processing pipeline instances is one of the fundamental concepts in the distributed PPI processing pipeline. Here it is possible to share interaction resources, e.g., digital pens, as well as processing resources, e.g., gesture recognizers. This allows for better resource utilizations and prevents pipeline instances from blocking resources. For instance, the distributed pipeline supports using the same pen on different applications by sharing the driver stage between pipeline instances. The cumbersome manual setup, i.e., the bluetooth pairing of the pen with a device, has to be done only once.

This non-exclusive resource access also considerably broadens the spectrum of possible use cases. For instance, it is now also possible to share the same paper document among applications via shared use of the region stage. This allows for more natural interaction with paper documents, e.g., a sheet of paper used for a drawing application can simultaneously serve as interaction medium for a note-taking application; in this setup, both applications simply receive digital ink of the same interactive region.

R2.2: IR Discovery As described in section 2.3.1, local storage of all interactive regions cannot be achieved. In order to support the interaction with encountered documents, the infrastructure needs to support *interactive region discovery*. In the monolithic approach, the processing infrastructure is deployed on a single device. This neither helps nor hinders the discovery of interactive regions. However, the monolithic pipeline design prevents deployments from sharing knowledge of interactive regions. This puts a lot more stress on the discovery mechanism compared to a setting where this knowledge would be shared across pipeline instances, e.g., if two applications are run on the same device. Hence the monolithic setup implies an unnecessary performance penalty.

By contrast, the dynamic *discovery of interactive regions* forms a central mechanism in the mPPI infrastructure based on the distributed PPI processing pipeline. Here it is essential to support interaction with encountered documents at the conceptual level.

On the one hand, components of the infrastructure can be shared across pipeline instances In the distributed PPI processing pipeline. Although this in itself does not provide a discovery mechanism for interactive regions, the possibility of sharing the region stage across pipeline instances reduces the stress put on the discovery mechanism as knowledge of existing interactive regions can already be shared here. As a consequence, the distributed approach decreases the performance penalty introduced by the monolithic approach, e.g., if two pipeline instances are hosted on the same device and share their knowledge on interactive regions.

On the other hand, support for interaction with encountered documents demands for distributed and highly scalable mapping of interactive regions to applications. At the same time, the dispatching of digital ink to interested parties must be handled efficiently and the solution must scale to a global dimension. In the mPPI infrastructure presented throughout this chapter, this is solved using the 2-level peer-to-peer based approach combined with local sharing of the interactive region model, i.e., the LIRM. Integrated into the pipeline's region processing stage, this 2-level Peer-to-Peer approach allows distributing the knowledge on regions efficiently in the local network by sharing the LIRM between instances of the region stage. As this is backed by a global overlay net for region publishing, the IRNS, it supports the dynamic definition of interactive regions needed to integrate interactive surfaces for both paper-like and other interactive surfaces, as well as the publishing of interactive paper documents.

Summary

In summary, mPPI infrastructure based on the distributed PPI processing pipeline inherently supports the requirements associated with user mobility and document mobility. This stands in contrast to the monolithic deployment of the generic PPI pipeline commonly found in existing PPI infrastructures that faces conceptual challenges when tackling these requirements. As a consequence, the distributed processing pipeline allows supporting mobile usage practices better and thus enables infrastructures to provide support for mobile PPI as demonstrated.

Requirement		Anoto SDK	LiveScribe / SP3 SDK	MIL SDK	PADD + PapierCraft	PaperToolkit	iServer + iPaper	***Letras***
User Mobility	R1.1 Interactive Mode	x	x	x	x	x	x	**x**
	R1.2 Mobile Platform	-	x	-	-	-	(x)	**x**
	R1.3 Flex. Deployment	-	-	-	-	-	-	**x**
Doc. Mobility	R2.1 Resource Sharing	-	(x)	(x)	-	-	(x)	**x**
	R2.2 IR Discovery	(x)	-	-	(x)	-	x	**x**

Table 3.2: Mobile PPI Support Comparison

Table 3.2 summarizes the analysis given above. It depicts the support for mobile PPI requirements as provided by the novel mPPI infrastructure through its reference implementation *Letras* in comparison to the architectures of existing PPI infrastructures (c.f., section 2.3). In summary, *Letras* provides support for mobile pen-and-paper interaction through the introduced distributed processing pipeline, enabling developers of PPI based applications to provide interactivity to physical paper, without compromising its inherent mobility.

3.4.2 Prototype Applications

Several prototype applications were developed in order to demonstrate that the distributed interaction processing pipeline allows supporting mobile PPI. Besides demonstrating feasability, they also aim to show practical relevance of the approach and outline how different deployment schemes contribute to the overall flexibility of the approach in various domains. This serves as the second part of the *proof-of-concept* evaluation (the reference implementation *Letras* constitutes the first part, c.f., section 3.3).

Mobile Applications

Mobile application prototypes form the central element of the proof-of-concept evaluation as they show how mPPI infrastructure based on the distributed PPI processing pipeline actually enables mobile use of PPI. In order to evaluate this claim of *enabling*, a total of six applications were developed based on *Letras*. Applications share

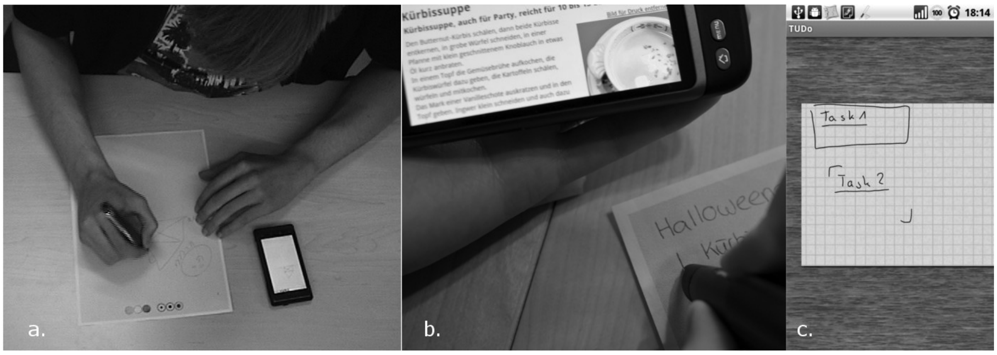


Figure 3.13: Prototype applications: a. the magic drawing application, b. cross-media bookmark application, c. the digital / physical task list

the same processing pipeline which bases on the adaptions for mobile platform usage as described in section 3.3.2.

Digital / Physical Task List This application combines the communication facilities of a mobile phone with traditional paper as input medium in a todo application, or task list. It is designed to allow noting tasks on any (Anoto enabled) paper document. As shown in Fig. 3.13 (c.), this can be achieved by simply drawing two opposite corners on any paper artifact where the user wants to note down a task. Handwritten tasks can then be shared with and delegated to other people. It is also possible to interact with tasks, e.g., complete, delete or edit them, using either mPPI or interaction on the smartphone. Thereby mPPI functionality enables users to mark a task as done using a simple check mark gesture, cross out a task in order to delete it, or edit the task on paper.

This application's implementation predates the *MobiLetras* mobile platform. Although it also hosts the driver and region stages directly on the mobile platform, no parts of the semantic and application stages are used. Gesture recognition and digital ink storage are implemented directly within the application. Thereby, gesture recognition bases on a simple implementation of the $1 Gesture recognizer [Wobbrock et al., 2007]. This application demonstrates how the distributed processing pipeline allows creating and disseminating new interactive regions (here: paper areas where the user writes down tasks) on the fly, i.e., during runtime. This exemplifies its power regarding resource sharing, where not only the digital pens, but also the paper are itself can be shared between applications and demonstrates a scenario requiring document mobility support at the infrastructural level.

Magic Drawing Application This application allows the user drawing on a sheet of paper (or write notes and other information), while the digital ink is recorded by the smartphone. Notes are represented using a digital ink facsimile on a digital representation of the paper sheet in the user's smartphone. The user can instantly interact with this digital version using the phone itself as an interaction device. She can change to various pencil stroke widths and colors, however, those will only be visible in the digital facsimile. The paper version is considered only a draft. In this application, Pidget interactors on special pre-printed paper documents allow using commands directly on paper as shown in Fig. 3.13 (a.). It is possible to flush away and store the digital facsimile by shaking the mobile phone. This simple application derives its name from the magic drawing table kids use, where sliding over the drawn contents erases them instantly.

The Magic Drawing application uses the *MobiLetras* infrastructure in combination with the interaction toolkit based on the conceptual framework for PPI introduced in chapter 4. It interprets digital ink stemming from interaction occurring on its interactive regions within the semantic stage of the pipeline (through the recognizers encompassed in the toolkit, c.f., section 4.3.3). Thereby, interactive regions are the specially prepared paper documents, as well as the Pidget interactors printed on them. This applications demonstrates the utility of the mPPI infrastructure with respect to user mobility: it employs the interactive mode of the pen, bases on the mobile platform and uses a specialized deployment scheme to adapt to this setting.

Hybrid Photo Scrapbook The hybrid photo scrapbook was modeled after applications described in the literature, i.e., *ButterflyNet*, [Yeh et al., 2006a], *Memento*, [West et al., 2007], and *Prism*, [Tabard et al., 2008]. It enables users to take notes on paper and enrich those with digital photos captured using a smartphone. Users can capture their notes using the digital pen on paper, while a facsimile of the digital ink is stored on the mobile device. In addition to digital ink, they can insert photos into their notes by drawing placeholders on the paper. This application also allows triggering the mobile device's camera directly when the user draws the placeholders to demonstrate the increased level of interactivity through supporting the interactive mode during mobile use.

Similarly to the magic drawing application, this application bases on *MobiLetras* and the interaction toolkit. Gesture recognition uses the $1 Gesture recognizer [Wobbrock et al., 2007] integrated into the pipelines semantic stage, storage of digital ink bases on a shared SQLite based data store being part of the application stage. It demonstrates that in this setting, infrastructural support for user mobility enables convenient use of mPPI.

Cross-media Bookmarks The cross-media bookmark application was modeled partly after *CoScribe*, [Steimle, 2009a]. It demonstrates in a limited prototype, how the distributed PPI processing pipeline enables a stationary application to be used in the mobile domain. As shown in Fig. 3.13 (b.), it enables users to take notes and link those notes with digital documents using cross-media links. Thereby, the user can mark certain regions of physical documents and thus connect these documents with web pages. Whenever the mark is subsequently touched with the digital pen, that particular web page opens on the smartphone's browser. Several interaction techniques exist to establish the bookmarks and will be introduced in detail in chapter 5.

As the magic drawing application and the hybrid photo scrapbook, this application bases on *MobiLetras* and the interaction toolkit. Gesture recognition uses a custom implementation for bookmarking gestures. Whenever bookmarks are established, an interactive region around the mark is created to serve as Pidget interactor for accessing a particular web page.

Integrated Note-taking Application This application combines digital and physical note-taking applications into a cohesive system. It enables the user to take notes on arbitrary (Anoto enabled) paper artifacts while being on the move and to integrate a digital facsimile of these notes with a digital note-taking system, i.e., the *Evernote* cloud based note-taking application[20]. Notes can be taken on any paper artifact by drawing two opposing corners of the note's area, a gesture similar to the one used in the digital / physical task list or in the hybrid scrapbook. Notes can later be simply re-accessed, i.e., their digital facsimile displayed on the mobile device, by tipping the pen on them. Additionally, the user can control the stroke width and color of generated digital ink through the mobile device as in the magic drawing application. Inserting photos into notes as in the hybrid scrapbook is also supported. As such, this application can be seen as a comprehensive successor to the aforementioned prototypes, integrating their capabilities with the *Evernote* system.

This application bases on *MobiLetras* where both driver and region stage are deployed on the mobile device itself. It also uses an SQLite based storage in the application stage to store digital ink in its original form on the mobile device, as well as a remote store where it synchronizes notes with the user's *Evernote* account. This makes their facsimile accessible on other devices, e.g., the user's desktop computer and demonstrates a slightly differing deployment scheme, where two independent application stage storages are used.

[20] https://evernote.com/ (accessed: July 2015)

Mobile SVG Editor The mobile SVG editor is another successor to the magic drawing application. It supports the same functionality, the drawing of diagrams or notes in mobile settings extended by the capability to export any drawings directly to the *Scalable Vector Graphics* (SVG) format[21]. Attributes of the digital facsimile, e.g., pen stroke width and color, can be controlled using the mobile device and paper. In contrast to the magic drawing application, the SVG editor does not require any pre-printed paper documents for applying controls on paper. The user can convert any (Anoto enabled) surface into a a drawing area by using the corner gesture. These regions, i.e., notes, can be re-accessed when tipping on them later. Digital ink properties can be controlled on the phone. Additionally, the user can draw pie-formed interaction proxies, that allow to bind a certain functionality to them. The size of the pie determines which functionality will be selected by a linear mapping of circumference to menu item on the mobile device. Whenever those pies are tipped, the bound functionality is invoked, e.g., the pen stroke color is changed.

This application also bases on *MobiLetras*. It uses a custom-built gesture recognizer for pie-based gestures and handles interaction proxy registration by creating new interactive regions bound to certain functionality. This exemplifies the flexibility of the publishing and discovery system for interactive regions within *Letras*. Additionally, a SQLite based data store is used to store drawings.

Digital Grocery List (DGL) The DGL supports grocery shopping in a mobile use case and requires both, user and document mobility support in the underlying infrastructure. It combines a mobile and desktop client for collaboratively editing and sharing grocery lists on paper and in the digital domain. This application will be discussed in detail throughout section 3.4.3, as it constitutes the case study reported in this section.

This set of applications shows that the distributed PPI processing pipeline enables the infrastructure to support the mobile use of PPI. It shows that several heterogeneous applications can easily be developed given that the infrastructure provides adequate support. Bringing stationary applications of PPI to the mobile domain, e.g., as with the hybrid photo scrapbook, becomes possible through to infrastructure satisfying the requirements for mobile usage.

Desktop and Smart-Environment Applications

Besides the mobile use case, PPI infrastructure also has to support stationary applications or applications being executed in a smart environment. Toward this end, it is

[21] http://www.w3.org/Graphics/SVG/ (accessed: July 2015)

important to use the same pipeline otherwise the advantage gained by sharing interaction resources through the distributed processing pipeline would be lost. All in all, the same pipeline need to be able to support stationary applications also. In order to validate this and to complete the proof of concept evaluation, a set of stationary applications was developed. It demonstrates that the distributed PPI processing pipeline supports stationary applications in parallel to mobile applications.

Table-Top Control The use case of this application is control of a table-top display by means of a digital pen in a smart environment. In this environment, the user can use the same digital pen to interact with several other applications and resources to facilitate seamless interaction. This application was initially implemented as an adaption of an earlier prototype developed by Steimle, [Steimle, 2009a], to evaluate *Letras* capabilities as supporting infrastructure for legacy systems. In Steimle's application the pen is used as direct pointing device on a rear-projection table-top display. This had been rebuild based on *Letras*.

Thereby, it was possible to considerably reduce the footprint of the overall application by using *Letras* instead of the original infrastructure (based on an adapted *PaperToolkit*, [Yeh et al., 2007], branch). The deployment scheme here hosts driver and region stages on the table-top. It does not require semantic processing or storage of digital ink, as this is handled within the legacy application. However, it defines an interactive region corresponding to the table-top screen and this wrapper delegates digital ink to the application itself.

Emergency Response Workflow Support As paper centric workflows still persist in control and enactment of emergency response processes, this application developed by Döweling et al., [Doeweling et al., 2013], supports the integration of such workflows into the digital world by augmenting a tabletop display in order to allow for interaction with digital pens. At the same time it includes interaction with paper artifacts in a smart control room.

This applications uses a deployment scheme where pen connection and region management are hosted in the smart control room on a central server. The user can control a table-top screen by means of a digital pen, while the same digital pen can be used to interact with a variety of applications including notes and emergency plans in the surrounding environment.

Collaborative Drawing and MindMapping Support This application aims to support group discussion in an augmented meeting room by offering a big central drawing surface (A0 format) onto which users can draw and write by means of digital pens. It supports simultaneous collaboration of multiple[22]

[22] it was tested with up to 12 users with two workstations hosting a driver stage each to overcome the

users. Thereby, each user employs a pen producing different colors in the generated, digital facsimile. After the group discussion, the contents of the discussion surface, e.g., contained drawings and mindmaps, can be exported to SVG and published either on a web server, or used in personal notes.

This application is deployed with region and application stage on one workstation per room, while the driver stage will be hosted on a set of other nodes, e.g., to allow for more digital pens being used simultaneously. The application stage uses a *Mongo DB*[23] backed data store for persistently storing digital ink.

PPI Based SVG Editor This application presents a stationary version of the SVG editor developed as a prototypical mobile application as described above in section 3.4.2. It shares its features and enables users to interact with the system by means of pie based interactors. Users can draw these interactors on paper in order to select pen stroke width and color. Drawings can be exported to SVG. It's gesture recognizer is a custom-built recognizer that allows calibrating and fine-tuning recognition properties (as used in the mobile version).

Deployment here is a simple flexible deployment, that only requires one version of the pipeline and its services somewhere in the vicinity[24].

Besides the applications presented above, various further systems base on *Letras* and its associated components. *Letras* has also been adopted by other researchers in the context of their scientific work, e.g., Lisserman et al. used it in order to add pen interaction to their research prototype [Lissermann et al., 2012]. For the sake of brevity, this section cannot enumerate all systems basing on *Letras*.

Despite initially being designed for stationary contexts, the applications presented above have successfully been tested in integration with personal digital pens connected to a user's personal mobile device hosting the driver stage, e.g., as part of one of the mobile prototypes. Here the digital pen resource can be shared and the same digital pen can be used, e.g., for collaborative drawing and other applications.

This shows that although all these applications were developed separately, they allow to share the same interaction resources through the *Letras* distributed processing pipeline. It is possible for example, for the user to easily control the graphical user interface to her shopping and task list with the same digital pen she uses to write on the collaborative drawing surface of the mindmapping application and the same pen she uses to interact with the tabletop system.

bluetooth personal area network (PAN) limitation of max. 8 devices

[23] http://www.mongodb.org/ (accessed: July 2015)

[24] this application has also been while creating some of the figures in this text, e.g., Fig. 5.4 in chapter 3

3.4.3 Case Study: The Digital Grocery List

The hybrid digital / physical grocery list was introduced in chapter 1 as promising application scenario for mobile PPI. This section demonstrates the development of a mobile PPI grocery list application based on the infrastructure for mobile PPI introduced throughout this chapter. Goal of this application is to support the grocery shopping process through a combination of traditional, paper-based practices, e.g., handwritten grocery lists and printed leaflets, with digital functionality offered by contemporary mobile devices. Thereby, the *Digital Grocery List* (DGL) application offers mPPI based creation, management and distribution of grocery lists.

Thereby, this application demonstrates in a detailed case study how infrastructure based on the distributed PPI processing pipeline supports real world applications relying on both, *user mobility*, where users can interact with and edit their grocery lists in mobile settings, and *document mobility*, where users encounter leaflets and can interact with them immediately to fill their grocery lists.

Application Design and support for Mobility

In chapter 1, section 1.1.2, several design implications for a *digital grocery list* (DGL) application were derived from a field study. In summary, these design implications are:

- support collaborative creation and editing
- allow access in the store to inform the shopping process (manipulation in the store is less relevant)
- provide a household vocabulary
- employ hybrid mPPI based design and allow for handwritten creation
- support arbitrary paper artifacts
- provide access and links to additional resources and information

An overview of how the hybrid digital grocery list application addresses these design implications follows. Of particular interest here are the *hybrid paper-digital design / handwritten creation*, the *arbitrary paper artifacts* and the *additional resources and information* design implications. The former requires a mobile shopping list application employing PPI to support *user mobility*, while the latter two require support for *document mobility*, both in the sense of using a specific paper document in different applications and in the sense of accessing previously unknown PPI based applications.

Collaborative Creation and Editing The DGL application is designed for collaboration: Using a smartphone, everybody can manage items in a shared list stored on a home-server. In order to share a list, multiple users can connect their smartphones to a home-server storing the digital grocery list. Thereby, home-servers allow to host multiple lists. Changes in grocery lists are immediately propagated to all users sharing a list, e.g., all users belonging to a household, by forwarding them to all connected mobile devices of the server. The shopper can then see changes in the shared list while on the way to the supermarket.

Access to Inform Items in the list are displayed by their name and an icon if available, e.g., for items stemming from the store's leaflets. This allows the shopper accessing the list intuitively and thus informs the shopping process. At the same time, items can be marked as purchased. Marking items as purchased is also communicated instantly to the home-serve which informs the planner as to which items are already being purchased.

Household Vocabulary Additionally, the home-server hosts the shared household vocabulary. When the user starts to enter an item into the digital version of the list, an auto-complete list of matching items is provided. The user than can either select one of these or continue entering a new item. New items are automatically stored in the shared vocabulary.

Hybrid Paper-Digital Design / Handwritten Creation Users can add items to their grocery lists by writing them on paper. As described below, basically any paper artifact can serve as entry area to the grocery list. While writing items on paper, a facsimile UI shows the items in the list, as shown in Fig. 3.14 (b.). Instantly visualizing the written ink helps users to understand the input they are providing to the system. The written items are attached to the grocery list on the home-server by pressing a button on the mobile device. In addition, the application supports gesture based input on paper leaflets as described below.

Both in stationary and mobile settings are supported. For stationary settings, a PPI enabled desktop client exists. In the mobile domain, the *Android* and *MobiLetras* based client for smartphones can be used. This mobile application allows for mPPI based entry on the move, i.e., in a situation that requires support for *user mobility*. Here the interactive mode support enables reviewing the facsimile of the digital ink instantaneously on the smartphone to allow for additions, corrections and deletions. Furthermore, the flexible deployment enables processing of newly entered handwritten items directly on the home server.

Arbitrary Paper Artifacts Just as with traditional, paper-only grocery lists, items can be added by writing them on any (Anoto enabled) sheet of paper. In order to come as close to using arbitrary paper artifacts as possible, the DGL application does not require to introduce the paper artifact beforehand. To use a sheet of Anoto paper with the DGL application, the user *draws two corners spanning a rectangle on any paper containing the Anoto dot pattern*[25], similar to the hotspot association gesture described by Yeh et al., [Yeh et al., 2006a]. The DGL application will recognize any items written into the region specified thereby.

This acquisition of arbitrary paper regions in the global pattern space presents a special form of document mobility. Here, a document is mobile in the context of not being bound exclusively to one application. This can only be achieved through the 2 stage region publishing and discovery mechanism in the region stage. It basically unlocks paper artifacts from exclusive application contexts and utilizes sharing of resources.

Additional Resources and Information Other paper documents are involved in the grocery shopping planning process besides the grocery list; users browse leaflets and commercial brochures, etc. These documents satisfy information needs, e.g., telling users which items are available as bargain offer. DGL integrates these documents directly into the planning process. Leaflets are also augmented with the Anoto dot pattern. As depicted in Fig. 3.14 (a.), the user can add or remove items depicted in the leaflet to the list by drawing a plus or minus on them. This lets the user keep track of the items selected for a shopping list even while working with the leaflet only.

In order to obtain pricing information on the mobile device, the user can choose a store for the shopping trip. If this has been done, long-clicking a list item displays additional information. DGL matches the list item with the products offered in the chosen store and shows the matching products, along with their pricing, availability and packing size. The user can compare different stores by reviewing the same shopping list for all of the stores. Users can further benefit from functionality offered by their smartphones, e.g. by navigating to the store via a map application.

As not all leaflets will be issued by the same authority, e.g., super markets are likely to compete for customers and therefore might refuse to cooperate, this requires inter-organizational discovery of new interactive regions and associated

[25] This generic gesture for dynamically acquiring an interactive region on behalf of an application is not specific to the DGL application. It solves a general problem occurring wherever an application requires (temporary) dynamic allocation of IRs. Therefore, it has been used in a wide variety of applications, e.g., the note-taking application described above or the SVG editor. The DGL merely demonstrates an application of the concept.

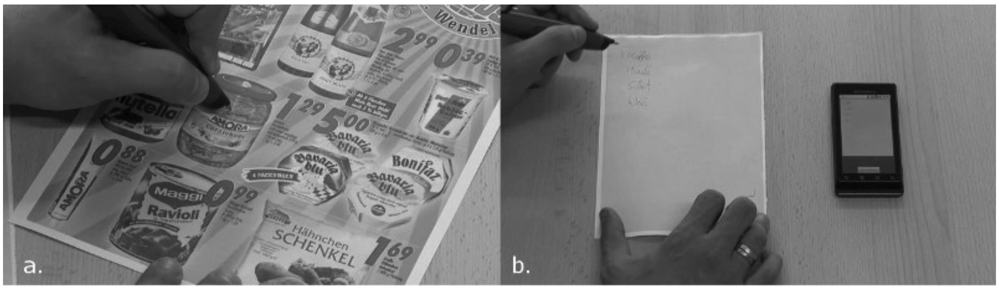


Figure 3.14: The DGL application: (a.) adding items directly in the leaflet with a digital pen, (b.) handwritten entry of list items

applications (c.f., requirement R2.2 as described in chapter 2, section 2.3.1). This corresponds to the second form of document mobility, where users encounter new previously unknown documents. Resource sharing here enables the user to employ the same digital pen in interacting with the leaflet as well as the DGL application. Additionally, the 2 stage region publishing and discovery system allows interacting instantaneously with encountered leaflets.

Architecture and Implementation

As the DGL application is based on the pub / sub paradigm of the mPPI infrastructure, various heterogeneous components can participate, such as multiple clients connected to a home server and $n : m$ connections between home servers and store (supermarket) servers. Thereby, the shared grocery list is stored on a home-server and accessed by several client applications. The same home-server hosts the household vocabulary. Client applications provide a GUI and optionally a PPUI. Each client application contains a local list to deal with connection loss.

Store servers thereby host a database containing an ontology of *products*. Products then can be mapped to items in the household vocabulary, e.g., "milk" can be mapped to specific brands and packing sizes. Additionally, this supports direct, specific entry, e.g., of goods on promotion. However, a sophisticated handwriting recognition as in [Liwicki et al., 2011] is currently not included. Store servers also enable access to additional information, e.g., store locations etc., and process interaction with leaflets.

This application employs an interaction concept, where users posses personal digital pens and connect them via personal mobile devices. The personal mobile devices then dispatch digital ink to interactive regions; thus driving applications, or discovering unknown applications by using the IRNS in case of unknown interactive regions.

Thereby, technically the paper *leaflet* contains interactive regions not defined within the DGL application, but within a *leaflet application* that allows adding items to digital grocery lists managed in the DGL application via a simple remote interface. As described above, upon receiving interactive leaflets, users might select special offers or even groceries with their pen out of the leaflet by interacting with their printed representations. Simultaneously, users might write grocery items to the handwritten-list. This actually encompasses seamlessly interacting with two different applications

Implementation The DGL application was developed in *Java* using the *MundoCore* [Aitenbichler et al., 2007] middleware for communication between its components. PPI support was realized via the reference implementation of *Letras*. Thereby, the mobile client application was developed on the Android platform, using API version 2.1 and above. mPPI support for the mobile version was based on *MobiLetras* where the driver and region stages are hosted on the mobile device in combination with an interaction cache (see section 3.3.2 for details).

It was tested and deployed with the *Nokia SU-1B* and *Logitech IO2 Bluetooth* digital pen models and their Android drivers for *Letras* on a *Motorola Milestone* and a *Samsung Galaxy S* smartphone.

3.5 Summary and Conclusion

Infrastructure plays a crucial role in supporting mobile PPI. Section 3.1 elaborated on conceptual issues of existing infrastructures with respect to supporting mobile usage practices and demonstrated their shortcomings. Subsequently, it introduced a novel concept for mPPI infrastructures based on the *distributed PPI processing pipeline* model. This infrastructure concept supports mobile usage practices by design while at the same time supporting the settings addressed by contemporary infrastructures.

Furthermore, section 3.2 described how this novel concept can serve as the architectural basis of mPPI infrastructures. In addition, section 3.3 demonstrated practical feasability and specific design considerations for implementation through a reference implementation of the concept: *Letras*, a novel infrastructure for mobile PPI processing. This also laid the foundation for an analytical and *proof-of-concept* evaluation as presented in section 3.4. Thereby, this section discussed how the distributed PPI processing pipeline supports mobile usage practices and demonstrated the practical relevance by reporting on a set of prototypical applications based on it. Finally, the evaluation took up the case study on grocery shopping introduced in chapter 1, section 1.1.2; thereby demonstrating in an in-depth example how the novel infrastructure supports the mobile use case.

Based on this novel concept of an mPPI infrastructure, interaction designers and application developers can for the first time target mobile applications employing the inherently mobile and convenient modality of pen and paper.

Thereby, the *distributed PPI processing pipeline* forms the conceptual underpinning of this approach: digital ink is successively transformed and enriched in a set of processing stages. Processing stages thereby constitute the driver stage connecting digital pen hardware, the region stage associating digital ink with interactive regions, the semantic stage attaching meaning and interpretation to digital ink and the application stage aiding storage and presentation. Furthermore, this approach enables flexible distribution and sharing of processing stages via clearly defined processing stage interfaces and network transparent links. Thereby, processing stage interface consist of services, communication channels and data structures enabling applications to construct flexible, shared pipelines following a micro service architecture.

As a consequences, resources can now be shared between different instances of the pipeline and applications. This enables, flexible, distributed deployments in both stationary and mobile environments. In combination with a 2-stage approach toward interactive region publishing and discovery based on local model synchronization combined with a global interactive region naming system (IRNS), this supports essential mobile usage characteristics of real pen and paper: *user mobility* and *document mobility*.

4 Conceptual Framework of (Mobile) Pen-and-Paper Interaction

Synopsis: *This chapter introduces a novel conceptual framework of PPI and mPPI. It enables formal description of interaction in the domain based on concepts adopted from logic programming and an open framework of conceptual dimensions, providing the foundation for a structured exploration of the PPI / mPPI design space. Subsequently, this chapter extends the basic conceptual framework by deriving a basic interaction vocabulary for PPI from exemplary interaction techniques described in the literature. In combination with machine understandable descriptions of interaction, this enables the conceptual framework to serve as foundation of toolkits for PPI / mPPI. Finally, practical relevance of the approach is demonstrated through a proof-of-concept implementation of an mPPI toolkit.*

Toward developing compelling and engaging interactive systems, designers should primarily focus on interaction between users and systems, not on technical aspects of user interfaces [Beaudouin-Lafon, 2004]. In particular, addressing technical peculiarities of interfaces too early on can hamper creativity and force adverse design choices. In this context, toolkits offered as part of the supporting infrastructure help interaction designers by providing means for rapid development and convenient abstractions, thus facilitating interaction design through exploration of the design space.

Thereby, as laid out in chapter 2, section 2.4, *conceptual frameworks* of interaction constitute the connecting element between interaction and infrastructure. On the one hand, conceptual frameworks serving as underpinning of toolkits determine the abstractions provided by these toolkits. As such, they ultimately determine what is available to the interaction designer. On the other hand, they also determine how interaction between users and systems can be expressed and perceived by the infrastructure. Thereby, it is expedient to employ a conceptual framework that actually matches the domain of concern, i.e., to use a conceptual framework of PPI as basis of a PPI or mPPI toolkit.

Essentially, a conceptual framework for the domain of PPI, or the more narrower scope of mPPI, aims to answer the question: How can pen-and-paper interaction techniques be described? Both flexible enough for human users, i.e., designers exploring

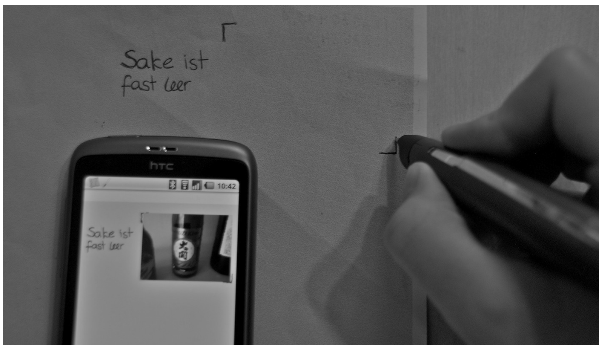


Figure 4.1: The crop mark selection technique in our prototype

the design space and formulating interaction techniques, and systems, i.e., the infrastructure interpreting inputs and triggering associated system functionality (c.f., section 2.4.1). Consider for example Fig. 4.1, where region on paper can be selected by drawing crop marks as suggested by Liao et al. [Liao et al., 2008]. How would we precisely describe this interaction technique in a way that a system, e.g., a toolkit, could understand it? How would the designer be able to explore similar, but potentially more suitable interaction techniques?

Toward answering these questions, this chapter introduces W^5 , a novel conceptual framework of PPI[1]. W^5 grounds on concepts introduced in prior work; however, it was specifically designed to provide a conceptual framework suitable for serving as basis of toolkits by addressing the requirements derived in chapter 2, section 2.4.2. As such it offers a basic design vocabulary (R3.1), supports composition of interaction techniques (R3.2), offers an open and extensible framework (R3.3) and provides machine understandable descriptions of interaction based on concepts adopted from logic programming (R3.4).

First, section 4.1 reviews the aforementioned requirements for conceptual frameworks for PPI as established in section 2.4.2. Then, it introduces the basic structure of W^5 and its fundamental concepts. This includes a set of conceptual core dimensions of PPI, fundamental principles of the framework, semantics and notation. Following this, section 4.2 empirically derives an initial design vocabulary through analysis of existing PPI techniques. Analysis thereby comprises central representatives of the three classes of PPI techniques derived in section 2.4.1. This empiric, analytical approach establishes a set of nine basic interaction predicates for interaction designers

[1] W^5 presents a conceptual framework of PPI in general as opposed to the narrower concept of mPPI. This allows W^5 serving as foundation for both, PPI and mPPI toolkits.

to build on. Finally, section 4.3 evaluates the theoretical concepts introduced. This includes a theoretic analysis of W^5 and the associated basic interaction vocabulary with respect to addressing the requirements raised toward conceptual frameworks serving as basis for PPI / mPPI toolkits; and a *proof-of-concept* implementation of a toolkit based on these concepts to demonstrate practical feasibility, i.e., that it actually enables constructing a toolkit for mPPI.

4.1 The W^5 Conceptual Framework

W^5 presents a conceptual framework for PPI, i.e., a way to describe, or express PPI. It focuses on describing the course of *interaction* between a user and a digital system through elementary aspects of user actions. As such it enables designers to precisely express how the user interacts with a digital system by means of pen and paper. On the one hand, this enables designers to explore the design space and select suitable interaction techniques for their problems at hand as they can now formalize what exactly constitutes an interaction technique. On the other hand, and in the context of this thesis more importantly, such a conceptual framework can serve as the basis for designing a PPI or mPPI toolkit; thus it provides the connecting element between interaction and the infrastructure supporting it.

This section embarks by reviewing the requirements toward conceptual frameworks aiming to serve as basis of toolkits, as derived in chapter 2, section 2.4.2. Then, this section introduces the basic structure of W^5 and its fundamental concepts, consisting of a set of conceptual core *dimensions* and *interaction predicates* enabling formulation of *expressions*. It also introduces the conceptual framework's semantics adopted from logic programming and discusses fundamental principles used in order to ensure compliance with the requirements toward conceptual frameworks.

4.1.1 Requirements Reviewed

As laid out above, the basic objective of a conceptual framework for PPI is describing interaction, i.e., PPI or mPPI, between users and digital systems. Here, *interaction techniques* form a central concept, as interaction techniques connect user actions, system responses and invoked digital functionality. As introduced in section 2.4.1, Hinckley defines an interaction technique as a combination of *input* with appropriate *feedback* triggering system *function*, [Hinckley, 2007]. In this context, an interaction technique forms a basic conversational unit of interaction between users and digital systems, an *expression* of interaction. Consequentially, addressing description of interaction techniques provides a natural starting point toward developing a conceptual framework of interaction.

Here it is important to consider further properties of interaction techniques. In addition to interaction techniques being combinations of their three basic elements, i.e., input, feedback and system function, interaction techniques themselves can be combined in order to form more complex, composite interaction techniques. This concept of *composition* to construct more complex techniques out of several basic techniques, e.g., as used to mark text for a copy command in *PapierCraft*, [Liao et al., 2008], has also been used as basis of *chunking and phrasing*, [Buxton, 1986].

Based on these characteristics of interaction techniques and the particular needs of toolkits, a set of requirements was derived toward conceptual frameworks of interaction as laid out in chapter 2, section 2.4.2. In the following, these requirements are repeated and initial design implications toward conceptual frameworks are introduced. Thereby, requirements are

R3.1: Design Vocabulary Conceptual frameworks need to offer a design vocabulary forming the basic building blocks offered to the interaction designer in order to describe interaction techniques; this then determines abstractions as provided by toolkits. Thereby, the conceptual framework needs to reflect the basic structure of interaction techniques, i.e., invoking functionality through actions coupled with (optional) feedback. At the same time, it has to offer a set of abstractions matching the domain of PPI, e.g., allow expressing existing classes of interaction techniques.

R3.2: Composition Conceptual frameworks need to support composition of interaction techniques. On the one hand, this entails composition of interaction techniques out of basic building blocks defined by the design vocabulary, e.g., enabling designers to build new techniques out of existing abstractions. On the other hand, this includes composition of multiple interaction techniques into larger, composite interaction techniques, e.g., toward support for chaining of interaction techniques.

R3.3: Openness and Extensibility Conceptual frameworks need to support an open, extensible approach toward encompassed design vocabulary at the conceptual level. That is, although a framework might offer a concrete design vocabulary, it cannot assume that this vocabulary remains fixed. Novel approaches might introduce novel basic building blocks toward designing PPI or mPPI.

R3.4: Machine Understandable Conceptual frameworks of PPI serving as basis for PPI and mPPI toolkits need to offer machine understandable descriptions of interaction techniques; the infrastructure needs to be able to determine when exactly to trigger system functionality as response to user input. Here, clear cut semantics for expressions describing interaction techniques are needed.

4.1.2 Basic Framework Structure

Based on the requirements reviewed in section 4.1.1, a conceptual framework of PPI was conceived. It enables interaction designers to express precisely which interactions are supported by a PPUI. Thereby, interaction is described through elementary aspects of user actions along an open set of conceptual core *dimensions* of interaction. Initially, five core dimensions constitute the framework, hence its name W^5. Thereby, W^5 not only facilitates description of interaction, it is specifically designed to serve as foundation for PPI or mPPI toolkits and thus offers semantics allowing to map conceptual abstractions to system responses.

W^5 describes interaction, i.e., PPI or mPPI, between a user and a digital system at the abstraction level of interaction techniques. As such, it allows designers expressing interaction techniques through formulating *expressions*. An expression thereby focuses on describing the course of interaction encompassed in an interaction technique through an operative description of *input*, i.e., atomic, observable aspects of user actions along conceptual dimensions. These observable aspects are called *interaction predicates*. Expressions correspond to composite chains of interaction predicates. In addition, expressions support composite chains of interaction predicates and other (sub-)expressions in order to form new expressions. This enables composition of interaction techniques, i.e., an interaction technique can consist of several other interaction techniques (see explanation of *composition* below).

This approach enables designers to formulate the input part of interaction techniques supported by an application. *Input* can then be bound to *function* and *feedback*. In W^5, both aspects are not explicitly modeled as they depend highly on the Hardware / Software composition of the targeted application. However, W^5 binds *function* to expressions; that is, whenever the system observes all *input* as described in an expression, e.g., when an interaction technique was performed by the user, the system *invokes* the associated function. Similarly, it binds *feedback* to interaction predicates; whenever the system observes a particular atomic aspect of interaction, it triggers feedback (if required).

In summary, the three basic structural constituents of the W^5 conceptual framework of PPI are

Expressions Expressions correspond to operative descriptions of the input part of interaction techniques. They consist of chains of elementary aspects of user actions, the so called *interaction predicates*. Chains consisting of varying combinations of interaction predicates and other (sub-)chains are possible in order to support composition of interaction techniques.

Interaction Predicates Interaction predicates are atomic, observable aspects of user actions along conceptual *dimensions*. As explained below, interaction pred-

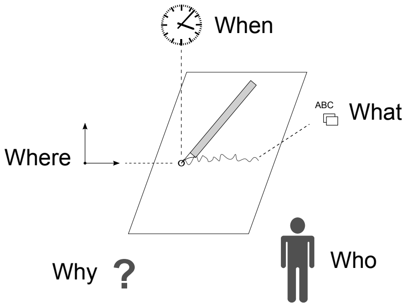


Figure 4.2: Dimensions of the $\mathbf{W^5}$ Framework

icates fully coincide with exactly one dimension. Thereby, multiple interaction predicates can describe a single, elementary user action, e.g., pointing with a pen on a Pidget. As shown below, interaction predicates correspond to the design vocabulary in W^5.

Dimensions Conceptual dimensions structure the design space and serve as containers for interaction predicates. As such, dimensions provide the structural frame of the framework and guide analysis toward an initial design vocabulary; additionally, dimensions can aid interaction designers in conceptualizing interaction techniques as they span the design space for PPI input.

Core Dimensions

As described above, conceptual *dimensions* classify interaction predicates. Classification is thereby based on conceptual properties of the interaction predicates, i.e., conceptual properties of atomic, observable aspects of user actions with a digital pen. Here it can be observed, that in PPI based systems user actions can be classified according to five central dimensions as illustrated in Fig. 4.2. These five dimensions constitute the *core dimensions* of the W^5 conceptual framework and lend its name. The five dimensions are:

$\mathbf{W_1}$ **Where?** : Spatial dimension

$\mathbf{W_2}$ **When?** : Temporal dimension

$\mathbf{W_3}$ **What?** : Content dimension

$\mathbf{W_4}$ **Why?** : Contextual dimension

$\mathbf{W_5}$ **Who?** : Originator dimension

All interaction predicates fully coincide with one of these dimensions. Spatial predicates (*Where*) describe aspects of the location of user actions, e.g., where the user touches the paper with a pen. Temporal predicates (*When*) describe aspects of the timing of user actions, e.g., when the user touches the paper with a pen. Predicates of the content dimension (*What*) refer to aspects of digital ink contents, e.g., what the user draws or writes with the pen on paper. Gestures or written commands belong to this dimension.

These first three dimensions directly relate to digital ink, i.e., the data observable by a PPI based system without additional information. However, as argued by Steimle, the perspective on the information ecology and the semantic dimension of interaction should not be neglected, [Steimle, 2009a]. Therefore, W^5additionally encompasses the contextual dimension describing associated purpose of actions (*Why*), e.g., currently executed tasks, and the originator dimension describing associated actors (*Who*), e.g., users. These five dimensions serve as core dimensions spanning the design space. However, completeness is not assumed as argued below, i.e., W^5 allows adding conceptual dimensions where required.

Examples. In order to illustrate how core dimensions structure the design space, consider *occurrence* as a simple spatial interaction predicate[2]. This interaction predicate describes *where* user actions occur through specifying whether system input can be observed at a certain place, e.g., when touching a certain paper region with the pen. As such, it can be directly used as an interaction technique triggering application functionality, i.e., as Pidget as in *PaperPoint*, [Signer and Norrie, 2007b], or *NiceBook*, [Brandl et al., 2010] (c.f., section 2.4.1).

Similarly, a simple interaction predicate from the temporal dimension could describe user action observed at a certain point in time, e.g., when the user has to touch the paper with the pen at a certain time. This might be used in a voting system, where the user has to mark a box with an X at the same time the desired choice (out of many) is shown on a screen. An example for a content predicate would be the aforementioned gesture predicate, e.g., when the system observes a *checkmark* gesture. Similar examples describe the originator and contextual dimensions. For instance, depending on the originator an application might, e.g., accept or reject a command. Furthermore, the current task can also influence available functionality, e.g., enabling modal interaction in PPUIs.

When forming expressions in order to describe interaction techniques, these predicates can be used standalone, as in the examples above, or predicates can be combined.

[2] This corresponds to the At_R predicate derived on page 157 in section 4.2. Here this and the following exemplary predicates are used only to illustrate the concept, not as a predefined constituent of the empirically derived basic domain vocabulary.

Combined predicates thereby reflect on multiple aspects of user actions along several dimensions. Additionally, combined predicates can express sequences and composite expressions as discussed below. For instance, an expression combining the spatial and temporal predicates above can express an interaction techniques where input has to be observed at a certain place at a certain time. In this case, application functionality would only be invoked if the user taps a certain paper region at a certain point in time.

Fundamental Concepts

As discussed above, W^5 allows forming expressions in order to describe input of interaction techniques through interaction predicates. Interaction predicates stemming from the five core dimensions introduced in the last section represent atomic, observable aspects of user actions. As the constituents of expressions, interaction predicates correspond to the domain vocabulary in W^5.

Domain Vocabulary. In W^5, there is no fixed domain vocabulary, i.e., no fixed set of interaction predicates that constitute the framework and claim to span the entire design space. Instead, W^5 applies the principle of *openness* as discussed below to both, dimensions and interaction predicates (i.e., the domain vocabulary) yielding an extensible set of core dimensions and core predicates. However, in contrast to core dimensions that are defined based on conceptual properties of interaction, core predicates are empirically derived from representatives of the three essential classes of PPI techniques laid out in chapter 2, section 2.4.1 (c.f., section 4.2). W^5 takes this approach in order to ensure a minimal, *relevant* basis interaction predicates.

Invocation. As laid out above, W^5 enables designers to formulate expressions describing the input of interaction techniques. Input is bound to functionality and feedback using the principle of *invocation*: whenever a single aspect of a user action (i.e., interaction predicate) contained in an expression occurs, the system *invokes* associated feedback[3]; similarly, whenever all aspects of user actions (and thus all user actions) contained within a particular expression occur, the system *invokes* associated functionality. This principle is based on the associative nature of the *RSL* model, [Signer and Norrie, 2007a]. *RSL* had been successfully applied to PPI, in particular in combination with Pidget interaction, [Signer and Norrie, 2007b]. Furthermore, the concept of invocation in combination with semantics of expressions lays the foundation for machine understandable representations of interaction as discussed below in section 4.1.3.

[3] In actual applications, feedback is of course optional. This can be conceptually modeled as ”invoking empty feedback”

Composition. Supporting composition of interaction techniques is a central requirement toward conceptual frameworks of PPI (R3.2). This entails two essential aspects of composition: on the one hand, conceptual frameworks need to support composing interaction techniques out of domain vocabulary; on the other hand, conceptual frameworks need to support composition of interaction techniques out of other interaction techniques. In W^5, expressions consists of composite chains of interaction predicates and other (sub-)expressions. As will be shown in section 4.1.3, *composition* thereby maps to logic operators connecting the interaction predicates. This enables composition at the syntactic level. However, composition at the semantic level of interaction predicates requires the additional concept of *relative predicates*.

Consider the examples of interaction predicates introduced on page 147. These predicates are *absolute*, i.e., they correspond to an absolute value observed in the digital ink produced by user actions. This concept becomes clear when looking at the example given above for the occurrence interaction predicate: here the system compares input with a pre-defined location (absolute value). However, specifying all input by means of absolute interaction predicates is problematic as they do not support conceptual relations of different aspects of user actions. However, this concept is needed toward enabling composition of expressions out of (sub-)expressions.

W^5 addresses this problem by introducing *relative* interaction predicates. As absolute interaction predicates, relative interaction predicates coincide with one of the framework's dimensions, e.g., one of the five core dimensions above. However, in contrast to absolute interaction predicates, relative interaction predicates describe relations of aspects of user interactions. An example for a relative spatial interaction predicate is ”above” meaning that one (aspect of a) user action must be performed above another, e.g., in the 2-dimensional pattern space. Another example for a relative interaction predicate is ”after”, stemming from the temporal dimension. It describes a relation where two (aspects of) user actions must be performed in a temporal sequence. For example, one can combine two spatial interaction predicates with the temporal sequence predicate (one shortly after the other) to describe, e.g., the double-click technique known from GUI systems.

Openness. Openness and extensibility at the conceptual level is another essential requirement toward conceptual frameworks of PPI (R3.3). Therefore, W^5 was designed following the principle of *openness*: completeness of dimensions or predicates is not required in the framework and cannot be assumed at any point. W^5 instead defines the structural constituents of the framework, i.e., expressions, interaction predicates and dimensions, as well as a relevant set of core dimensions and core predicates as part of the framework. Toolkits based on the framework need to reflect this by basing on the structural constituents, as well as offering core dimensions and core interaction predicates. Thereby, core dimensions essentially provide a relevant conceptual frame for analysis of aspects of elementary user actions (and potential structuring to toolkits based on W^5); while core interaction predicates provide relevant aspects of user actions, i.e., aspects that are actually used in existing interaction techniques.

Extensibility is then achieved by allowing additional dimensions and interaction predicates to be added to the toolkit. For instance, a toolkit based on W^5 needs to offer all the empirically derived core interaction predicates constituting the framework (c.f., section 4.2). However, it has to be designed extensible so that developers can easily add additional interaction predicates as required by their applications.

4.1.3 Semantics and Notation

Conceptual frameworks of PPI need to satisfy the requirement of providing machine understandable descriptions of interaction in order to form an eligible basis for PPI or mPPI toolkits (R3.4). Toward this end, W^5 derives its semantics from logic programming. This is based on the observation that interpretation of user actions and the invocation based approach of W^5 correspond well to term solving problems commonly addressed by logic programming approaches or production rule systems: given certain, observable input DI, the system needs to decide for each possible interaction technique whether its associated functionality shall be invoked or not. Thereby, expressions correspond to *rules*, while observed digital ink DI corresponds to *facts* in logic programming terminology.

Basics

As laid out above, W^5 defines expressions as composite chains of interaction predicates describing elementary aspects of user actions. Thereby, expressions correspond to the input part of interaction techniques, while feedback and function are invoked upon observing input. *Expressions* are thereby formed in a subset of first order predicate logic. As such, their structural constituents are

Logical Constituents This encompasses *conjunction* ($\wedge$), *disjunction* ($\vee$), *negation* ($\neg$), as well as the two *truth constants*: *true* (T) and *false* (F)

Algebraic Constituents This encompasses parentheses (”()”) and brackets (”[]”) identifying precedence in evaluating expressions, the *equality* relation and symbol ($=$) and an infinite set of variables denoted through lower case letters (indexed where need), e.g., x, y, …

Symbols This encompasses interaction predicates with differing valence or arity as laid out below (depending on what they exactly express). Interaction predicates are denoted as upper case letters or symbols (which are not structural constituents of expressions); predicates with arity ≥ 1 carry the variable(s) they apply to in parenthesis, e.g., $A(x)$, $B(x, y)$, $\ldots$[4]

This allows forming expressions. Note, that logical quantifiers ($\exists$, $\forall$) are not required in order to express interaction with W^5. However, they might be used while reasoning about W^5 expressions. The same holds for further logical symbols such as *implication* ($\rightarrow$) or *equivalence* ($\leftrightarrow$). This enables implementing systems to optimize recognition of expressions, i.e., allows for more efficient solving approaches, without limiting the expressive power.

Given a set of observable input (facts), interaction predicates and hence composite logic expressions either evaluate to *true* or evaluate to *false*. Thereby, facts are the actions carried out by the user with a digital pen as observed by the system (in combination with other relevant system input), here denoted as DI. Any interaction predicate, or expression which based on received DI evaluates to *true* is considered *invoked*. Invoked expressions trigger the associated functionality of the interaction technique; similarly, invoked predicates trigger associated feedback.

Bindings. In order to allow for detecting invocation of expressions, W^5 uses the concept of *bindings* which is commonly found in term solving approaches provided by logic programming languages [Lloyd, 1984] and relates to the logic concept of unification (see, e.g., [Baader and Snyder, 2001]). Bindings essentially describe an association of system input with an interaction predicate; that is, the binding B_A of an interaction predicate A is the subset of DI for which the interaction predicate evaluates to *true* (if any) as expressed in equation 4.1:

$$B_A = \{x \in DI : A(x) = T\} \tag{4.1}$$

Detection of invocation of such a predicate then determines whether $B_A \neq \emptyset$, in which case it is considered *invoked*. Detection of invocation of a composite expression thereby corresponds to evaluation of the boolean expression with respect to received facts, where all contained predicates are evaluated and the composite binding is determined. Thereby, the process of detecting invocation of expressions also encompasses unification of variables, i.e., bindings are returned in the variables and can be reused in the expression. Bindings for variables have to satisfy all predicates and logic state-

[4] As shown in equation 4.5 on page 154, the hierarchical shorthand notation omits the variable in cases where it is not further used in order to increase legibility.

ments containing these variables. This concept allows relating multiple aspects of user actions (along heterogeneous dimensions) to a single action.

Consider for instance an expression consisting of two interaction predicates A, B that apply to the same variable, i.e., that are required to be present in the same input: $E = A(x) \wedge B(x)$. As determined through unification, the binding for the composite expression E would be

$$B_E = \{x \in DI : (A(x) = T) \wedge (B(x) = T)\} \tag{4.2}$$

As stated above, detection of invocation thereby depends on the returned binding for the composite expression. Equation 4.2 shows that this binding consists of those facts for which both, A and B are true. Using the semantic of invocation as defined above yields that the functionality associated with this expression is triggered upon receiving input, i.e., user actions, for which both aspect A and aspect B are present. This demonstrates how user actions with elementary, observable aspects in multiple dimensions can be expressed through multiple interaction predicates relating to the same variable.

Domain of Discourse: Interaction Predicates

System input, and in particular user actions expressed through interaction predicates represent the domain of discourse in W^5. As laid out above, interaction predicates thereby correspond to atomic, observable aspects of user actions along conceptual dimensions. Interaction predicates can be applied to any system input DI and evaluated. Depending on presence of the atomic, observable aspect they refer to, interaction predicates then evaluate either to *true* (aspect is present in DI) or to *false* (aspect is not present in DI). Furthermore, relative interaction predicates express relations among other aspects of actions in order to enable semantic composition in addition to syntactic composition enabled by the underlying first order predicate logic.

Following the principle of openness, W^5 defines no closed domain of discourse: it encompasses basic, empirically derived *relevant* predicates (c.f., section 4.2), yet remains open to extensions through additional predicates (and associated semantics). However, as argued above, W^5 defines two semantically different classes of interaction predicates: *absolute interaction predicates* and *relative interaction predicates*. Thereby, classes can be distinguished by their arity.

Absolute Predicates Each *absolute* interaction predicate has an arity ≤ 1, i.e., is a nullary or unary predicate. Nullary predicates thereby correspond to constants or external, non PPI input (c.f., EXT predicate defined below). In contrast to this, unary predicates correspond to observable, atomic aspects directly relating

DI to absolute values or predefined conditions along the conceptual dimension of the respective predicates.

For instance, consider the spatial *occurrence* interaction predicate example discussed in section 4.1.2 on page 147. It describes input occurring at a particular place, e.g., on an interactive region. As introduced below in section 4.2, this predicate is denoted as $At_R(x)$ and evaluates to *true*, whenever it is applied to DI lying inside the region R. Thereby, DI is related to the region R, an absolute value in the conceptual *spatial* dimension (W_1). Here, the bindings returned in x for $At_R(x)$ would be all DI observed lying within R.

Relative Predicates Based on the concept of bindings, *relative* predicates relate (aspects of) user actions to other (aspects of) user actions. Relative interaction predicates thereby are n-ary predicates with $n \geq 2$. Bindings returned by relative predicates correspond to those elements of DI for which the relative predicate evaluates to *true* respecting contained (bound) variables, e.g., for a binary relative interaction predicate $P(x, y)$, the binding of its x variable is

$$B_{P_x} = \{x \in DI : \exists y \in DI \; . \; P(x, y) = T\} \tag{4.3}$$

As exemplified in equation 4.3, relative predicates correspond to observable, atomic relations of two or more sub-sets of DI along the conceptual dimension of the respective predicate. Relating two or more sub-sets of DI provides the basis for semantic *composition* as it allows for expressions reflecting *how* observable aspects of user actions are connected.

For instance, consider the binary temporal *sequence* interaction predicate example discussed in section 4.1.2 on page 149. It describes two actions, or aspects of actions, occurring in temporal sequence, i.e., *one* after *another*. As introduced below in section 4.2, this predicate is denoted as $\curvearrowright (x, y)$ and evaluates to *true*, whenever it is applied to two elements of DI where x occurred before y in the conceptual *temporal* dimension (W_2). As such this predicate relates two facts in *time* and $\curvearrowright (x, y)$ returns all bindings to x and y so that x occurred (or was observed) before y.

W^5 allows forming expressions consisting of composite chains of aspects of user actions through *absolute* and *relative* interaction predicates in combination with the mapping to logic programming.

Example. Consider the following example to illustrate how the concepts of absolute and relative predicates, in combination with binding of variables, enables expressing composite chains of user actions. Based on the example interaction predicates discussed above, we aim to describe an (arguably nonsensical) interaction technique, where the user has to ”double click” on three icons on paper in succession in order to invoke some functionality. Thereby, *”clicking on icons”* corresponds to observing DI at a place and can be expressed using the example At_R predicate, while *”double [. . .]”* and *”succession”* refers to temporal sequences and can be expressed using the example $\curvearrowright$ predicate (here used as both, binary and ternary relative predicate). In W^5, this interaction technique can then be expressed as

$$\begin{aligned} x &= At_{R_1}(x_1) \wedge At_{R_1}(x_2) \wedge \curvearrowright(x_1, x_2) \\ y &= At_{R_2}(y_1) \wedge At_{R_2}(y_2) \wedge \curvearrowright(y_1, y_2) \\ z &= At_{R_3}(z_1) \wedge At_{R_3}(z_2) \wedge \curvearrowright(z_1, z_2) \\ \boldsymbol{E} &= \curvearrowright(x, y, z) \end{aligned} \tag{4.4}$$

Thereby, equation 4.4 shows how the relative temporal interaction predicate $\curvearrowright$ expresses a relation between the spatial At_R interaction predicates. It also defines the expression $\boldsymbol{E}$ based on bindings to variables x, y and z, subsequently relating their respective sub-expressions in the temporal domain (*”succession”*). This demonstrates how expressions can be composed of sub-expressions by means of relative predicates in W^5.

Hierarchic Shorthand Notation. As can be seen in equation 4.4, expressing complete interaction techniques can result in lengthy systems of equations introducing a lot of variables simply in order to express relations among interaction predicates. While this is important for machine understandability, it does not aid legibility, e.g., for interaction designers conceptualizing interaction techniques. Therefore, W^5 introduces the *hierarchic shorthand notation*. It allows substituting interaction predicates in-situ for variables iff variables are not reused anywhere else in the term and the semantics of the term are not changed. Using the hierarchic shorthand notation, equation 4.4 can be rewritten as

$$\boldsymbol{E} = \curvearrowright(\curvearrowright(At_{R_1}, At_{R_1}), \curvearrowright(At_{R_2}, At_{R_2}), \curvearrowright(At_{R_3}, At_{R_3})) \tag{4.5}$$

Note that the in-situ substitution of sub-expressions for variables is used recursively in equation 4.5. As such, this abbreviated notation emphasizes the hierarchic composition of predicates and allows for considerable shortening of expressions.

The EXT Predicate Because W^5 as described up to know only addresses PPI input, it is impossible to combine non-PPI input with PPI within the framework. However, this is unrealistic: in real world systems other means to interact with the system exist, e.g., mouse and keyboard in the desktop setting or multi-touch interaction with a smartphone in the mobile setting. To remedy this limitation, W^5 introduces a special predicate called EXT, that evaluates to true when any relevant non-PPI input occurs. Relevant here refers to ”relevant as part of the interaction technique”. This could, e.g., be pressing a button on a keyboard to switch between inking mode and command mode for the UI as in [Liao et al., 2008].

4.2 Design Space Analysis: Design Vocabulary

Toward serving as foundation of toolkits, conceptual frameworks of interaction need to offer an interaction design vocabulary (R3.1), while at the same time remaining open and extensible (R3.3). W^5 addresses this by following the principle of openness. As such it defines no closed interaction vocabulary, it rather offers conceptual abstractions defining constituents of the interaction vocabulary as introduced in section 4.1. Additionally, it provides a basic, empirically derived *relevant* portion of the design vocabulary, the so called *core interaction predicates*. Each toolkit based on W^5 needs to offer this core interaction predicates. However, W^5 remains open to extensions at the conceptual level through additional interaction predicates and associated semantics: designers can define additional required predicates within W^5 as needed; implementing toolkits need to offer appropriate concepts allowing for extensibility with respect to interaction predicates.

This section empirically derives the W^5 core interaction predicates through analysis of existing PPI techniques in order to ensure a minimal, *relevant* basis. Analysis thereby comprises central representatives of the three classes of PPI techniques as laid out in chapter 2, section 2.4.1: Pidgets and Proxies, Gesture Systems and Cross-Media Links. Thereby, the representatives of the three classes were chosen according to complexity, i.e., systems employing the most sophisticated and complex interaction techniques of each class. Representatives chosen for analysis are

Pidgets and Proxies Interaction techniques stemming from two comprehensive systems built on the *iServer / iPaper* framework and its *RSL* model were chosen as representatives of this class: *PaperPoint*, [Signer and Norrie, 2007b], and *PaperProof*, [Weibel et al., 2008]. These systems provide a comprehensive example for rich and meaningful interaction mainly based on Pidgets. While *PaperPoint* only uses Pidgets, *PaperProof* combines Pidgets and gestures in a sophisticates approach and as such lays the foundation for the subsequent analysis of gesture systems.

Gesture Systems *PapierCraft*, [Liao et al., 2008], was chosen as generic representative for this class. *PapierCraft* provides an advanced example for gesture based input systems that combines gestures with several other interaction techniques. It therefore provides a convenient starting point for the analysis of gesture based PPI techniques, yielding more complex interaction techniques as compared to other, simpler gesture system.

Cross-Media Links *CoScribe*, [Steimle, 2009a], represents the cross-media linking interaction techniques. It combines advanced context based interaction techniques in the domain of PPI with existing techniques, e.g., Pidgets. Additionally, it contains cycles of actions spanning paper and digital artifacts. As such, it adds considerably to the coverage of derived interaction predicates.

In the following each of these systems will be analyzed aiming to derive a set of core interaction predicates for W^5. This empiric, analytical approach establishes an initial design vocabulary of nine *core interaction predicates*[5] for interaction designers to build on.

Thereby, as laid out above, interaction predicates correspond to atomic, observable aspects of user actions along conceptual *core dimensions* defined in the framework (c.f., section 4.1.2). Absolute interaction predicates describe aspects in relation to absolute values of their conceptual dimensions. Thus, these predicates are defined in parametric form and allow relating to different absolute values in order to increase their utility. In contrast to this, relative interaction predicates describe relations among aspects of user actions. Here, the utility of the definition is increased by employing an n-ary definition (for $n \geq 2$).

This section starts by analyzing *PaperPoint*, [Signer and Norrie, 2007b], and *PaperProof*, [Weibel et al., 2008], as traditional Pidget based approaches toward PPI. It thereby derives interaction predicates required to express the encompassed interaction techniques. It then analyzes *PapierCraft*, [Liao et al., 2008], a representative of sophisticated gesture based PPI and adds predicates required for expressing gesture based interaction. Subsequently, it analyzes interaction techniques described in *CoScribe*, [Steimle, 2009a], and adds predicates required for expressing cross-media linking techniques to the emerging vocabulary. Finally, this section provides an overview of the nine derived interaction predicates. Thereby, it offers a discussion of coverage taking other, additional interaction techniques described in the literature into account (besides the three representatives).

4.2.1 Pidget Interaction: PaperPoint and PaperProof

PaperPoint, [Signer and Norrie, 2007b], and *PaperProof*, [Weibel et al., 2008], employ the *iServer / iPaper* infrastructure for PPI. As such they represent an important

[5] To be precise, these are *interaction predicate classes*, as each of the nine interaction predicates can be used in parametric form (absolute predicates), or as n-ary definition (relative predicates).

class of systems basing interaction on the *RSL*, [Signer and Norrie, 2007a], conceptual framework of interaction which focuses mainly on invocation of PPI techniques. As discussed in chapter 2, section 2.4.3, *RSL* introduces the concept of link invocation through different selectors, e.g., regions on paper documents. This provides a natural basis for Pidget based interaction.

PaperPoint. *PaperPoint*, [Signer and Norrie, 2007b], represents an early approach toward PPI. It revolves around interaction through Pidgets, combined with free form drawing and annotating. In *PaperPoint*, users can control functionality offered by the *Microsoft PowerPoint* presentation software through a set of Pidgets. Users can print out an overview version of the slides contained in a presentation. The system then enables users to point the digital pen on icons representing slide show functionality, e.g., move one slide forward, start the presentation or continue at a particular slide number. Additionally, the system enables users to directly annotate print outs of slides and show a facsimile of the annotations in the presentation.

Here, two exemplary representatives of the interaction techniques offered by *PaperPoint* were chosen in order to facilitate analysis of interaction predicates required toward expressing these techniques in W^5. The first representative is the technique for starting a presentation at a particular slide by tipping on a printed ”show” button with the digital pen. Modeling this interaction techniques in W^5 requires an interaction predicate describing *occurrence* in the spatial dimension, i.e., the $At_R(x)$ predicate defined below. The second representative of interaction techniques offered by *PaperPoint*, enables users to draw or write on a particular slide. The system then switches to this slide and displays a facsimile of the digital ink recorded on this particular slide. This requires the aforementioned *occurrence* predicate, as well as an additional interaction predicate expressing reception of uninterpreted[6] *content* in the content dimension, i.e., the $C(x)$ interaction predicate.

The required interaction predicates are thereby defined as

$\mathbf{At_R(x)}$ The $At_R(x)$ predicate describes *occurrence* of digital ink in the spatial dimension W_1. Thereby, this absolute predicate is a parametric predicate relating to occurrence of digital ink at the interactive region defined by parameter R. As such, this predicate evaluates to *true* for all digital ink in x that lies within region R.

$\mathbf{C(x)}$ The $C(x)$ predicate is an auxiliary absolute predicate describing reception of uninterpreted *content* for further use in an interaction technique. It stems from

[6] Uninterpreted here means that the system does not evaluate the content per se, it merely records it and subsequently uses it in combination with other interaction predicates or triggers functionality based on content (e.g., displaying its facsimile)

the content dimension W_3 and characterizes the need for entering digital ink, i.e., handwriting or drawing, as part of an interaction technique. As such it always evaluates to *true* binding received digital ink without any additional conditions, e.g., when aiming to record digital ink for using it as facsimile.

Based on the defined predicates, the Pidget interaction technique (E_1) and the slide annotation interaction technique (E_2) can then be expressed in W^5 as

$$\begin{aligned} \boldsymbol{E_1} &= At_{S_1}(x) \\ \boldsymbol{E_2} &= At_{S_2}(x) \wedge C(x) \end{aligned} \tag{4.6}$$

Note that the parameter R of the absolute $At_R(x)$ predicate used in equation 4.6 relates to two different interactive regions in this example (with S_1 reflecting the "show" icon and S_2 reflecting the slide printout).

PaperProof. *PaperProof*, [Weibel et al., 2008], builds on the concepts introduced in *PaperPoint* and adds semantic interpretation of content, thus employing interaction predicates along the content dimension (W_3). *PaperProof* enables users to perform proof editing of documents directly on a paper printout. This includes inserting, deleting, replacing, moving and annotating text. Thereby it combines the spatial occurrence with gesture based interaction techniques (c.f., discussion of *PapierCraft*, [Liao et al., 2008]) in order to express more sophisticated, chained interaction techniques.

Here, the two most complex interaction techniques offered by *PaperProof* were chosen for analysis: *annotation* and *move*. Interestingly enough, the informal notation employed by Weibel et al. to describe the interaction techniques offered by *PaperProof* resembles the structure of expressions used in W^5, [Weibel et al., 2008]. Although it lacks some details, e.g., with respect to location of gestures, it can be readily transcribed. Thereby, annotation of text elements requires first enclosing the text in brackets (*gesture*), consisting of an opening followed by a closing bracket, and then writing digital ink, i.e., the actual annotation. Similarly, moving requires enclosing the text (as for annotation) and then entering a special line gesture.

Expressing these techniques in W^5 requires two additional interaction predicates in the W_2 and W_3 dimensions respectively

$\mathbf{G_S(x)}$ The absolute $G_S(x)$ predicate describes a *gesture* in the content dimension W_3. $G_S(x)$ is a parametric predicate relating to recognition of a particular gesture (i.e., one constituent of the gesture vocabulary). Thereby, the particular gesture is defined by parameter S. As such, this predicate evaluates to *true* for all digital ink in x that constitutes the gesture symbol S.

Technique	Formalization
annotation	$\curvearrowright(At_R(G_{cs1} \vee G_{cs2} \vee G_<),$ $At_R(G_{ce1} \vee G_{ce2} \vee G_>), DI)$
move	$\curvearrowright(At_R(G_{cs1} \vee G_{cs2} \vee G_<),$ $At_R(G_{ce1} \vee G_{ce2} \vee G_>), At_{R'}(G_N))$

Table 4.1: Interaction Techniques in *PaperProof*, [Weibel et al., 2008]

$\curvearrowright(\mathbf{x_1}, \mathbf{x_2}, \ldots)$ The relative $\curvearrowright$ predicate expresses a temporal *sequence* of its contained variables (W_2). W^5 defines it as n-ary predicate with $n \geq 2$, that is, it can be used to express a temporal sequence of an arbitrary number of constituents. As such, $\curvearrowright(x_1, \ldots, x_n)$ evaluates to *true* iff $\forall x_i, x_j.1 \leq i < j \leq n$ the digital ink x_i was received *before* the digital ink x_j. As such it allows relating sequences of (elements of) user actions.

Based on these predicates in combination with the predicates defined above, the *annotation* interaction technique (E_3) can then be expressed in W^5 as

$$\begin{aligned} \boldsymbol{E_3} = {} & At_R(x) \wedge (G_{CS_1}(x) \vee G_{CS_2}(x) \vee G{<}(x)) \\ & \wedge At_R(y) \wedge (G_{CE_1}(y) \vee G_{CE_2}(y) \vee G{>}(y)) \\ & \wedge C(z) \wedge \curvearrowright(x, y, z) \end{aligned} \tag{4.7}$$

Here G_{CS_i}, G_{CE_i}, $G_<$ and $G_>$ are different start and end delimiter gestures for marking text as defined in [Weibel et al., 2008]. The interactive region R used in the At_R predicate refers to the paper document, paragraph or sentence being marked. Bindings for x and y then express that both start and end gestures need to be executed sequentially in the same region, followed by an annotation that can be given anywhere. The produced annotation itself is bound to z in $C(z)$.

As can be seen in equation 4.7, the full formal notation of real world interaction techniques in W^5 can produce lengthy expressions. Therefore, the following examples employ the *hierarchic shorthand notation* defined on page 154. Table 4.1 presents the hierarchic shorthand notation of expressions for both interaction techniques discussed, *annotation* and *move*.

4.2.2 Gesture Systems: PapierCraft

Toward gesture based command systems, Liao et al. describe a set of interaction techniques employed in *PapierCraft*, [Liao et al., 2008]. These techniques rely on a sophisticated gesture set enabling users to invoke complex document editing operations, e.g., copy and paste, deletion and content editing. In the *PapierCraft* system, the user annotates and edits paper printouts of digital documents using a digital pen. Thereby, changes in the printout are reflected back to the respective digital document.

When interacting with the *PapierCraft* system, the user needs to press a ”gesture” button while specifying a command in order to switch from *annotation* to *editing* functionality. A command consists of a sequence of a *command scope* followed by an *intermediate delimiter* and finally a *command type*. Commands can be constructed in sequences, e.g., a copy command followed by a paste command.

Thereby, the external input (button press) can be expressed in W^5 using the EXT predicate as defined above. However, *parallel* execution of actions requires a novel relative predicate in the temporal dimension W_2, i.e., the $\|$ predicate (or the less restrictive temporal *interval* predicate $\Vdash_T$) defined below. Command types in *PapierCraft* can be specified either using gestures, i.e., with the $G_S(x)$ predicate as defined above, or by writing the command name, e.g., ”copy”. Expressing the latter in W^5 requires an additional predicate describing written command *words* in the content dimension W_3.

$\|(\mathbf{x_1}, \mathbf{x_2}, \ldots)$ The relative $\|$ predicate expresses *parallelism* of its contained variables in the temporal dimension (W_2), i.e., it describes (aspects of) user actions occurring simultaneously. It is defined as n-ary predicate with $n \geq 2$, that is, it can be used to express a temporal parallelism of an arbitrary number of constituents. As such, $\| (x_1, \ldots, x_n)$ evaluates to *true* iff $\forall x_i, x_j$ the digital ink x_i and the digital ink x_j were received *simultaneously*.

$\Vdash_{\mathbf{T}}(\mathbf{x_1}, \mathbf{x_2}, \ldots)$ Similar to the $\|$ predicate above, the $\Vdash_T$ predicate expresses its contained variables occurring in a temporal *interval* (W_2), i.e., it describes (aspects of) user actions occurring shortly after each other. The parameter T thereby describes the length of the maximum time interval allowed for all variables to occur. As the $\|$ predicate, the $\Vdash_T$ is defined as n-ary predicate with $n \geq 2$. Thereby, $\Vdash_T (x_1, \ldots, x_n)$ evaluates to *true* iff $max_t(x_1, \ldots, x_n) - min_t(x_1, \ldots, x_n) \leq T$, that is if the time between observing the first and the last digital ink is no larger then T, i.e., all occur within the *interval* T.

$\mathbf{W_S}(\mathbf{x})$ The absolute $W_S(x)$ predicate describes a written *word* in the content dimension W_3. Similar to $G_S(x)$, $W_S(x)$ is a parametric predicate relating to recognition of a particular word. Which particular word it refers to is defined by parameter S and depends on the command vocabulary of the underlying system. This predicate evaluates to *true* for all digital ink in x that is recognized by the system as word S.

Technique	Formalization
copy & paste	$\curvearrowright(\|(EXT, At_{R_1}(\curvearrowright(G_{CS}, G_{CE}, G_{PGT}, G_E, W_{CP}))),$ $\|(EXT, At_{R_2}(\curvearrowright(G_{CS}, G_{CE}, G_{PGT}, G_W))))$
hyperlink	$\curvearrowright(\|(EXT, At_{R_1}(\curvearrowright(G_{MB}, G_{PGT}, G_N))),$ $\|(EXT, At_{R_2}(\curvearrowright(G_{PGT}, G_S))))$
stitch	$\|(EXT, \curvearrowright(At_{R1}, At_{R2}, At_{R1})$ $\wedge \curvearrowright(G_{ST}, G_{PGT}, G_S))$

Table 4.2: Interaction Techniques in *PapierCraft*, [Liao et al., 2008]

With these additional predicates, W^5 can express the interaction techniques employed in *PapierCraft*, [Liao et al., 2008]. Again, a representative selection of three interaction techniques was chosen in order to facilitate analysis. Other interaction techniques in *PapierCraft* consist of (sub-)portions of these, or combine them sequentially. Chosen techniques are

- *copy and paste* technique using the crop mark selection gesture with an explicit written command
- *hyperlink* technique with a margin bar selection gesture
- *stitching* technique to combine two paper artifacts

Thereby *copy and paste* technique starts by selecting some content on paper through a *crop mark* selection gesture at a certain place in the document. Here content is marked through drawing corners (G_{CS}, G_{CE}) around it[7], similar to the hotspot association gesture described by Yeh et al. [Yeh et al., 2006a]. Then follows a pigtail gesture (G_{PGT}) oriented to the right, i.e., east (G_E), and the command word ”copy” (W_{CP}) marking selected functionality. Text is then pasted through following a similar sequence: marking the area to paste text to, followed by a pigtail gesture oriented to the left, i.e., west(G_W).

The two other techniques follow a similar pattern. However, the *hyperlink* technique employs a different selection technique: here, content is marked through drawing a vertical marking bar (G_{MB}) in the region containing the text. Pigtail gestures and different orientations of the tail mark the hyperlink start and endpoints (G_N, G_S). The *stitching* techniques also uses a gesture for the stitch mark (G_{ST}) followed by

[7] For instance, as depicted in Figure 4.1 on page 142.

a pigtail to issue its link command. However, this gesture needs to span to paper documents and end on the same document where it started.

Table 4.2 presents the expressions describing these three interaction techniques in W^5 (using hierarchic shorthand notation).

4.2.3 Cross-media Techniques: CoScribe

CoScribe, [Steimle, 2009a], provides support for collaborative knowledge work in a hybrid paper digital setting. It offers a comprehensive set of PPI techniques enabling knowledge workers to simultaneously work with paper and digital documents in a tabletop setup. Thereby, it particularly focuses on establishing cross-media links between paper and digital artifacts. *CoScribe* differs from other approaches as it incorporates the contextual domain into the design of interaction techniques through its *conceptual activities*, [Steimle, 2009a]. Regarding interaction, this implies that the current task plays a role in interaction: executing an action as part of activity A might result in a different outcome than doing the same action as part of activity B.

With respect to the analysis, again two representative interaction techniques were chosen; thereby omitting techniques that exist as sub-techniques or combinations of chosen techniques, or were already analyzed as part of other approaches above, e.g. stitching. The examined techniques include a technique for creating *hyperlinks* between paper and digital documents and a technique for *tagging* documents.

In the hyperlink technique, the user presses the digital pen at a certain location on paper for a duration exceeding $500ms$, followed by tipping the pen on the tabletop screen displaying a (digital) document. However, both actions need to be executed in the context of document annotation. The tagging technique relies on similar steps, here in the context of document classification: the user can write a label on a tagging card (optional), then presses the pen on the card for at least $500ms$ and finally tips the pen on the digital document. This classifies the document according to this tag. Expressing these interaction techniques in W^5 requires two novel interaction predicates:

$\vdash_{\mathbf{T}}(\mathbf{x})$ The absolute $\vdash_T$ interaction predicate expresses (an aspect of) a user action executed for a certain *duration* in the temporal dimension W_2. It is a parametric predicate relating the duration for receiving digital ink contained in its variable x to its parameter T. As such, the $\vdash_T(x)$ predicate evaluates to *true* for all digital ink in x that spans the duration T, i.e., where the first and the last ink data received lie at least T apart.

$\mathbf{T_S}(\mathbf{x})$ The absolute $T_S(x)$ predicate describes a *task* in the contextual dimension W_4. It is a parametric predicate relating to execution of a particular task S recognized by the system. The $T_S(x)$ predicate evaluates to *true* for all digital ink x received as part of executing task S.

Technique	Formalization
hyperlink	$T_L(\curvearrowright (For_t(At_{R_1}), At_{R2}))$
tag	$T_T(\curvearrowright (At_{R_1}(W_T), For_t(At_{R_1}), At_{R2}))$ $\vee \quad T_T(\curvearrowright (For_t(At_{R_1}), At_{R2}))$

Table 4.3: Interaction Techniques in *CoScribe*, [Steimle, 2009a]

The composition of these techniques out of novel and previously defined interaction predicates is shown in table 4.3 (using hierarchic shorthand notation). As can be seen, the only distinction between the two interaction techniques in this example is the contextual task T_S (if we omit the optional label writing action $At_{R_1}(W_T)$ yielding the first sub-expression used in *tag*). Thereby, contextual tasks used are *linking* (T_L) and *tagging* (T_T).

Note also, the At_R predicate expresses an arbitrary interactive region R. It does not conceptually distinguish whether this interactive region reflects a paper document or a digital screen (in this example, At_{R_2} actually refers to a screen). This is due to W^5 aiming to serve as application independent foundation of toolkits. Thereby, the semantics of interactive regions, i.e., what a particular interactive region means in a particular context, are exclusively defined by the application.

4.2.4 Coverage of Derived Interaction Predicates

As shown above, a relatively small set of nine *core interaction predicates* suffices to model a multitude of interaction techniques in the three representatives chosen. However, their applicability does not end there. Other systems proposed in the literature offer interaction techniques that can be described using these nine core interaction predicates.

For instance, *Knotty gestures*, [Tsandilas and Mackay, 2010], introduces interaction techniques consisting of tapping, holding, circling and marking. These are an example for using the gesture predicate G_X in combination with the absolute spatial At_R and temporal $\vdash_T$ predicates. An interesting observation here is, that the user ”creates” the regions for the spatial At_R predicate at run time, i.e., the *knot* which is used in other techniques, e.g. by tapping.

Furthermore, *NiceBook*, [Brandl et al., 2010], as representative of a sophisticated PPI based note taking application relies on interaction techniques entirely expressible using the core interaction predicates. This includes Pidgets, a tagging system comparable to the one described in *CoScribe*, [Steimle, 2009a], and a dog-ear mark corresponding to the stitching gesture discussed above.

predicate			PapierCraft	PaperPoint	PaperProof	CoScribe	Knotty Gest.	NiceBook	ButterflyNet
W₁	At_R	occurrence	X	X	X	X	X	X	X
W₂	$\Vdash_T$	interval	-	-	-	X	-	-	X
	$\|\|$	parallelism	X	-	-	-	-	-	-
	$\curvearrowright$	sequence	X	-	X	X	X	X	X
	$\vdash_T$	duration	-	-	-	X	X	X	-
W₃	G_S	gesture	X	-	X	X	X	-	X
	W_S	word	X	-	X	X	-	-	-
	C	content	X	X	X	X	-	X	X
W₄	T_S	task	-	-	-	X	-	-	-
W₅	-		-	-	-	-	-	-	-

Table 4.4: Use of core interaction predicates in PPI based systems

ButterflyNet, [Yeh et al., 2006a], represents another well-known PPI based note taking application. It supports multi-medial data capture for field biologists and introduces a set of interaction techniques used for associating media. Its interaction techniques are automatic time-based correlation, *hotspot* association and visual specimen tagging. These are expressible using gesture G_X, absolute location At_R and temporal sequence $\curvearrowright$ in combination with the temporal interval predicate $\Vdash_T$. Similarly, *PLink*, [Steimle et al., 2011], offers cross-media links and note-taking activities relying on temporal association between digital media and paper artifacts in the interaction techniques described. Thus its cross-media links can be expressed using the $\Vdash_T$ predicate.

Although independently designed and developed, all these techniques can be expressed in W^5 using its previously established core interaction predicates. Table 4.4 provides an overview of approaches discussed here. It shows which predicates are required in order to express their respective interaction techniques. As can be seen, the core interaction predicates allow expressing a broad variety of Pidget based interaction techniques, gesture systems and cross-media link based techniques thus spanning a significant portion of the design space.

Therefore, it can be concluded, that this empirically derived core predicates constitute a *relevant* portion of the design vocabulary. As such they provide a sound basis for the interaction vocabulary offered by W^5. Toolkits based on W^5 need to offer this core interaction predicates while remaining open to extension at the conceptual level through additional, user defined interaction predicates.

4.3 Evaluation

The W^5 conceptual framework of PPI was evaluated in order to assess its suitability as conceptual underpinning of toolkits for PPI and mPPI. The evaluation method employed consisted of a *theoretical analysis* and a *proof-of-concept* evaluation. Toward this end, the evaluation comprised three steps.

In a first step, an analytical evaluation of the W^5 conceptual framework of interaction was conducted. Thereby, the approach was analyzed with respect to providing a suitable theoretical underpinning for PPI and mPPI toolkits based on the four essential requirements toward conceptual frameworks as established in chapter 2, section 2.4.2. This constitutes the *theoretical analysis*.

Then, in a second step, theoretical concepts were applied in developing an mPPI toolkit based on W^5 in order to assess practical applicability. Based on this mPPI toolkit, a set of three interaction techniques were developed in a third step. These techniques are modeled after existing interaction techniques as described in the literature. Together, these steps constitute the *proof-of-concept* evaluation, i.e., demonstrate that the concepts described can actually serve as basis of an mPPI toolkit.

4.3.1 Analytic Evaluation

This section provides an analytic evaluation of the W^5 conceptual framework of PPI. It thereby demonstrates that W^5 provides an adequate conceptual underpinning for PPI and mPPI toolkits, as it offers the conceptual abstractions required by interaction designers in order to express interaction techniques. Analysis thereby bases on the requirements for conceptual frameworks as established in chapter 2, section 2.4.2: providing a *design vocabulary* (R3.1), allowing for *composition* of interaction techniques (R3.2), supporting *openness and extensibility* (R3.3) and being *machine understandable* (R3.4).

R3.1: Design Vocabulary In order to satisfy this requirement, conceptual frameworks need to specify a design vocabulary representing the basic abstractions and building blocks offered to interaction designers by toolkits. In particular, expressing interaction techniques must be possible and the framework needs to provide building blocks spanning existing classes of PPI techniques. W^5 addresses this by offering expressions describing user actions required in order to perform interaction techniques. Thereby, it bases the vocabulary describing these expressions on elementary aspects of user actions. Associated functionality and feedback are then triggered whenever corresponding aspects or complete expressions were detected. However, W^5 defines no fixed, self-contained domain vocabulary. Instead, it defines basic abstractions organizing the vocabulary and provides an open, extensible set of relevant core vocabulary constituents (c.f., discussion of openness and extensibility below). The basic abstractions with respect to vocabulary constituents are conceptual *dimensions* and *interaction predicates*.

Predicates describe observable, atomic aspects of user actions fully coinciding with a single dimension. Predicates consist of *absolute* and *relative* interaction predicates. Absolute predicates can be detected independently, or relate aspects of a user action to an absolute value (parametric absolute interaction predicates). Relative interaction predicates relate multiple aspects of user actions in a conceptual dimension. Based on these abstractions, the core vocabulary constituents are the five core dimensions (W_1, W_2, W_3, W_4 and W_5) described in section 4.1.2; and the nine empirically derived core interaction predicates described in section 4.2 spanning the three classes of PPI techniques as established in chapter 2, section 2.4.1: Pidgets and proxies, gesture systems and cross-media links.

R3.2: Composition Supporting composition requires conceptual support for composition of interaction techniques out of basic abstractions defined by the design vocabulary and composition of multiple interaction techniques into larger, composite techniques (c.f., chaining as introduced in chapter 2, section 2.4.1). W^5 supports this through two concepts. On the one hand, the structure of its expressions supports composition of interaction predicates. Thereby, expressions base on first-order predicate logic and are composed of predicates connected through logical operators ($\wedge$, $\vee$ and $\neg$). Together with variables and the concept of bindings, this allows formulating and connecting expressions out of arbitrary sub-expressions and as such supports syntactic composition of expressions. On the other hand, W^5 introduces the concept of *relative interaction predicates* as described above. Thereby, relative interaction predicates allow relating aspects of user actions, or complete sub-expressions (through bound variables). This concept enables semantic composition, as it enables designers to describe the relations between different (parts of) expressions.

R3.3: Openness and Extensibility This requires conceptual frameworks to provide an open, extensible design vocabulary in order to support extensibility with respect to novel interaction techniques at the conceptual level. This implies that the addition of novel design vocabulary constituents required by these techniques must be possible. Satisfying this requirement comes natural in W^5 as it bases on principle of *openness*. W^5 does not assume a fixed, self-contained interaction vocabulary, it rather offers a flexible and extensible framework for specifying the PPI design space. Thereby, it only provides the structural abstrac-

Requirement		RSL	Steimle's Framework	W^5
Suitability for Toolkit	R3.1 Design Vocabulary	-	(x)	**x**
	R3.2 Composition	x	x	**x**
	R3.3 Openness and Extensibility	-	-	**x**
	R3.4 Machine Understandable	x	-	**x**

Table 4.5: Comparison of Suitability for Toolkit Design

tions as discussed above, i.e., conceptual *dimensions* and *interaction predicates*, in combination with an open set of *core* vocabulary constituents. As such, it remains conceptually extensible: if required, additional conceptual dimensions can be added, as well as novel interaction predicates. However, an essential caveat is that toolkits based on W^5 need to reflect this extensibility in order to avoid compromising the openness of the approach.

R3.4: Machine Understandable This requires conceptual frameworks of PPI to offer machine understandable descriptions of interaction in order to serve as basis for PPI and mPPI toolkits. W^5 satisfies this requirement by basing its semantics on concepts adopted from logic programming. Here, the *input* of interaction techniques is described using expressions in first-order predicate logic, where interaction predicates constitute the domain of discourse. Detecting user interaction then corresponds to evaluating the different expressions (which describe supported interaction techniques) with respect to known facts, i.e., digital ink received by the system. Thereby, *feedback* can be triggered at the interaction predicate level whenever there is a non-empty binding for an interaction predicate available (considering its variables). *Invocation* is triggered whenever there is digital ink allowing a complete expression describing an interaction techniques to evaluate to *true*, i.e., if there is a non-empty binding for all variables used in the expression.

Comparison to Existing Approaches

W^5 is grounded on prior work. It forms around the general associative paradigm for PPI to model and describe interaction techniques similarly to the *RSL* approach employed in *iServer / iPaper*, [Signer and Norrie, 2007b]. However, W^5 combines this with the semantic and syntactic levels of PPI described by Steimle, [Steimle, 2009a]. Thereby, it generalizes the syntactic level, as it uses three dimensions (spatial, temporal, content) which are, in contrast to [Steimle, 2009a], independent from any application domain. As a result, the model of Steimle can be derived from W^5 by picking appropriate representatives from each dimension. However, the two models are not isomorphic: W^5 allows expressing interaction techniques that cannot be expressed in the conceptual framework introduced by Steimle, e.g., temporal sequences. Also, in contrast to [Steimle, 2009a], W^5 models PPI without adding aspects of tangible interaction. Only input created by touching the paper with a pen is considered, e.g., writing, drawing or pointing. Other input such as folding or rearranging paper requires the EXT predicate.

As can be seen in table 4.5, W^5 combines the advantages of *RSL*, i.e., its composition semantics (R3.2) and machine understandability (R3.4), with the advantages of Steimle's conceptual framework, i.e., its basic structuring of the design vocabulary (R3.1), and composition of interaction techniques (R3.2). However, it adds semantics derived from logic programming and provides an open and extensible basis (R3.3). It also offers a set of concrete interaction predicates in contrast to Steimle's framework, which offers interaction vocabulary only at the conceptual level, e.g., ”combining”. As such W^5 represents the first approach satisfying the requirements toward conceptual frameworks of PPI offering a theoretic toolkit basis.

4.3.2 Proof-of-Concept Evaluation: mPPI Toolkit based on W^5

This section reports on the *proof-of-concept* evaluation conducted in order to validate the practical applicability of the W^5 conceptual framework. It describes design considerations and implementation of a lightweight mPPI toolkit based on W^5. The toolkit thereby supports expressing interaction using the W^5 conceptual framework and offers an implementation of its core interaction predicates suitable for the mobile domain. As such, it lays the foundation for the three applications described in section 4.3.3 demonstrating its application in a mobile use case.

Basic Concepts

In the W^5 framework, interaction techniques are modeled as composite expressions of interaction predicates representing elementary, observable aspects of user actions. In particular the two classes of *absolute* and *relative* predicates exist. Thereby, absolute predicates express certain conditions of input data, or relate input data to absolute values; whereas relative predicates describe relations between other predicates (and sub-expressions). Evaluating these expressions requires that at each system state, i.e., for each observable combination of system inputs, all interaction predicates can be independently evaluated. If the composite expression describing an interaction technique evaluates to *true*, corresponding application functionality is triggered. Of course, parts of the composite expression (sub-expressions) could have been *true* before, i.e., at the point of a prior observation.

The mPPI toolkit based on W^5 reflects this as a system of interrelated rules. In this approach, rules correspond to the interaction predicates. Each rule *fires* whenever it observes data corresponding to this rule and marks digital ink it is applied to as *consumed*. Data can consist of digital ink (in the case of *absolute predicates*) or of data generated by the system when firing other rules (in the case of *relative predicates*), i.e., recognition events.

Following this scheme, the rule-based implementation of the mPPI toolkit based on W^5 splits the actual recognition and the structure imposed by the formal description of an interaction technique into separate concerns, as discussed below. It thereby offers

i. support for the core interaction predicates described in W^5 (as rules and recognizers)

ii. support for extensibility in terms of new interaction predicates (as interfaces for rules and recognizers)

Thereby, the *proof-of-concept* presented here omits implementation of two core interaction predicates: the *task* predicate T_S and the *word* predicate W_S, e.g., as described in *CoScribe*, [Steimle, 2009a]. Although these predicates must be part of a theoretic basis of the design space, implementing them requires sophisticated recognition technology, i.e., handwriting recognition and activity recognition. Adding such systems is beyond the scope of a *proof-of-concept* implementation. Therefore, the task rule is realized simply as a placeholder for a global system state which can, e.g., be activated as result of a preceding interaction technique such as pressing a button in order to switch to executing a certain task; word recognition can be prototyped by modeling the words as distinct gestures.

Integration into *Letras*

The mPPI toolkit was implemented based on the *Letras* infrastructure (c.f., chapter 3) and integrates into the distributed PPI processing pipeline forming the backbone of *Letras*. Basically, it provides both a component for the developer of interaction techniques allowing to model the techniques by formulating expressions as defined by W^5 and a hook into the *Letras* pipeline as part of the semantic stage. The W^5 toolkit implementation thereby adds support for the core interaction predicates and the W^5 dimensions in the form of a rule engine that evaluates the data traveling through the *Letras* pipeline.

As stated above, the toolkit design splits the actual recognition and the structure imposed by the formal description of interaction techniques into separate concerns:

Recognizers First a set of *recognizers* allows detecting events and data along the core dimensions. This directly provides support for the *absolute predicates*, e.g., At_R or G_S. All recognizers corresponding parametric predicates can be configured to take different parameters as input, e.g., several regions R or gestures S. When a predicate is recognized, the recognizer fires a corresponding event. Recognizers thereby can be regarded as empty rules, or a rule that fires when a single event occurs.

Rules Second, *relative predicates* and complete interaction techniques are modeled as *rules*. Rules relate events received and as such, allow expressing more complex, composite conditions. Thereby, rules receive the events emitted by the recognizers and fire events, iff all required events specified by a particular rule have been received. These events then triggers digital functionality and feedback in the application.

Within the *Letras* infrastructure, the second processing stage, i.e., the *region stage*, channels data to interactive regions defined by the PPI based applications. As described in section 2.1, interactive regions can be seen as the area of interest defined by a PPI based application, e.g., the paper documents it handles. However, the concept of interactive regions can also be used to express elementary aspects of user actions in the spatial domain; an application can define multiple interactive regions and any received input within these regions corresponds already to the elementary At_R predicate: the recognizer here has only to determine whether there is any digital ink present in order to fire its event.

The third processing stage, the *semantic stage*, is concerned with interpreting digital ink, e.g., recognizing gestures. Existing components of this stage, e.g., the gesture recognizer, partially correspond to required predicate recognizers. For instance, the gesture recognizer directly supports recognition of the absolute interaction predicate G_S in W^5's content dimension (W_3). Additionally, the semantic stage provides the entry point for deploying executable components of the toolkit itself, i.e., its rule engine (as it interprets digital ink). This engine basically consumes events generated in the pipeline and injects events back into the pipeline whenever rules fire. This can then be used by the application in order to trigger digital functionality and feedback.

Thereby, the distributed pipeline architecture considerably eased adding interaction predicates to *Letras*: on the one hand, as extensions to pipeline stages in order to provide recognizers for absolute predicates, e.g., the At_R recognizer at the *region* stage;

on the other hand, by offering a flexible micro service architecture for recognizers and the rule engine as dedicated services in the semantic pipeline stage, e.g., to realize gesture recognition in order to offer the G_S predicate.

Implementation The *proof-of-concept* implementation of an mPPI toolkit based on W^5 was implemented in Java, as were most parts of *Letras*. It was designed in order to support rapid development of PPUIs for mobile devices and as such bases on the *MobiLetras* version of the *Letras* platform. Thus, it was deployed on the Android[8] operating system for smartphones. It is thereby based on the *Android 2.1 API version* and was tested on the *Motorola Milestone* and the *HTC Desire* smartphones. The experimental setup described below in section 4.3.3, constituted the mPPI toolkit, the *MobiLetras* platform and its pen drivers in combination with *Nokia SU-1B* digital pens.

4.3.3 Proof-of-Concept Evaluation: Interaction Techniques

Three applications employing interaction techniques described in the literature were implemented in order to demonstrate coverage and suitability of the W^5 based mPPI toolkit with respect to supported interaction techniques. As described above, these applications offer completely functional prototypes and employ all three classes of interaction techniques, i.e., Pidgets and proxies, gesture systems and cross-media links. However, these applications form a *proof-of-concept* and as such do not compete with the original applications they resemble in terms of functional depth and usefulness.

Initially described in section 3.4.2, prototypes developed basing on the W^5 mPPI toolkit and the respective interaction techniques covered encompass

Hybrid Photo Scrapbook The hybrid photo scrapbook, as described in section 3.4.2, was modeled after several applications described in the literature, most notably *ButterflyNet*, [Yeh et al., 2006a]. This mobile application enables users to combine notes on paper with digital photos. Thereby, photos are interactively captured using a camera built into the mobile device and pasted into the facsimile of notes on paper. This application employs the *hotspot* association interaction technique described as part of *ButterflyNet*, [Yeh et al., 2006a] (see Figure 4.1); this technique also constitutes the main part of the gesture based selection interaction techniques described in *PapierCraft*, [Liao et al., 2008]. Corresponding recognizers for the At_R, $\|$, $\Vdash_T$, $\curvearrowright$, C, EXT and G_X predicates where developed in order to allow expressing these interaction techniques as well as the facsimile writing.

[8] http://www.android.com (accessed: July 2015)

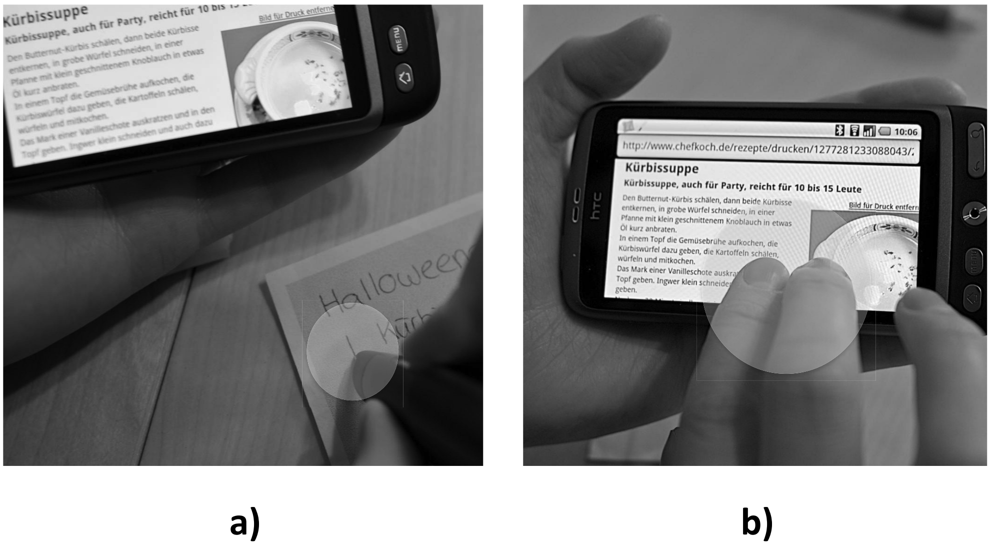


Figure 4.3: A cross-media linking technique of CoScribe in our prototype: a) drawing a marker on paper and b) on the smartphone in sequence to establish a link

Magic Drawing Application As initially described in section 3.4.2, this application enables the user to draw or write on a sheet of paper. At the same time, the digital ink is recorded and its facsimile is presented on a smartphone. This application uses a palette printed on paper documents to grant the user control over various stroke widths and colors. Thus it employs a Pidget based interaction technique to set the drawing mode of the digital pen, e.g., similar to *NiceBook*, [Brandl et al., 2010], or *PaperPoint*, [Signer and Norrie, 2007b]. This application employs recognizers for the At_R, $\curvearrowright$ and C predicates in order to allow expressing the Pidget based interaction techniques as well as facsimile writing.

Cross-media Bookmarks As described in section 3.4.2, the cross-media bookmark application was modeled after *CoScribe*, [Steimle, 2009a]. In this application, the user can establish cross-media hyperlinks between paper documents and websites. It thereby uses a similar interaction technique to the techniques offered by *CoScribe*: the user draws a vertical line beside the part of the page that should be linked and subsequently associates this page to a web page currently displayed in the smartphone's browser. In order to link the two pages, a similar gesture can be used, however, requiring touch input on the smartphone, i.e., using the finger instead of the pen (see Figure 4.3). This application also employs recognizers for the At_R, $\curvearrowright$ and $\vdash_T$ predicates, as well as the EXT predicate in order to allow expressing the cross-media interaction techniques.

As can be seen, a limited set of recognizers in combination with the rule engine suffices to conveniently model interaction techniques described in the literature. The implementation and adaption of recognizers thereby was limited to the absolute minimum, still it proved to allow an efficient implementation of differing interaction techniques while minimizing the boilerplate code (as well as redundancies that inevitably occur if every application builds on custom recognition systems). Therefore this implementation suggests that a rule based toolkit based on the W^5 conceptual framework provides a convenient way to express and develop PPI techniques for interactive systems.

Discussion

The three example applications demonstrate that the W^5 conceptual framework provides a suitable basis for an mPPI toolkit, enabling interaction designers to develop novel interaction techniques. Thereby, the mPPI toolkit implementation bases on a set of machine understandable rules and maps evaluation of expressions to an event processing approach (R3.4). It thereby provides W^5's core interaction predicates, with the exception of the word recognition predicate W_X and the user task predicate T_X, as argued above. As such, it offers a basic interaction vocabulary supporting design and development of interaction techniques stemming from all three classes of existing PPI techniques (R3.1). Thereby, the rule engine and the basic recognizers support composition of vocabulary and sub-expressions (R3.2).

It also offers extensibility in terms of predicates: new recognizers can be attached to the pipeline, injecting additional recognition events. These events can then be consumed either directly by applications (absolute predicates), or by generic rules. Additionally, novel rules can be integrated by implementing the rule interfaces offered by the toolkit (i.e., fire events upon recognition and consume events as defined by *Letras*) thus adding support for relative predicates. As such the toolkit remains open and extensible (R3.3).

However, The *proof-of-concept* implementation using a custom made rule-based system still has drawbacks in terms of developer support. For instance, the interaction designer currently has to maintain active knowledge how the recognizer affects the rules in the system and rules have to be specified programmatically. In some cases, this leads to rules having a low re- use factor: although they define relative predicates, some developers used them to provide application logic.

A possible solution to this problem would be a domain specific language based on logic programming, or an actual mapping to a logic programming system such as *SWI Prolog*[9]. This would allow to specify interaction techniques directly and offers a cleaner separation of concerns, preventing developers from accidentally mixing definitions of interaction techniques and application logic.

4.4 Summary and Conclusion

Conceptual frameworks as the basis of mPPI toolkits form the connecting element between infrastructure and interaction in the context of mobile PPI as they determine conceptual abstractions offered to interaction designers. Section 4.1 introduced W^5, a novel conceptual framework of PPI, specifically designed to provide a sound basis for PPI and mPPI toolkits. Section 4.2 then derived an interaction design vocabulary through analysis of representatives of the three main classes of PPI techniques: Pidgets and proxies, gesture system and cross-media links. Finally, section 4.3 provided an analytical, as well as a *proof-of-concept* evaluation of the approach.

W^5 describes interaction at the level of interaction techniques. It derives its semantics from logic programming and formalizes the input of interaction techniques through composite, first order predicate logic expressions. Expressions are thereby formed of interaction predicates: elementary aspects of user actions along conceptual dimensions. Conceptual dimensions provide analytical guidelines to interaction designers and structure the design space. Toward this end, W^5 defines an extensible set of five core dimensions. In addition, W^5 defines nine empirically derived core interaction predicates spanning a significant portion of the design space. Both, dimensions and predicates, remain extensible and open at the conceptual level.

W^5 currently does not express *feedback* as it is problematic to generically describe feedback: it depends on available hardware and software components. W^5 therefore just binds feedback to individual interaction predicates, i.e., the atomic constituents of expressions. Similarly, the evocation of *functionality* is bound to expressions: whenever the system observes input matching an expression, the functionality associated with that expression is triggered. Toolkits based on the W^5 conceptual framework therefore need to offer appropriate hooks in predicates and expressions in order to enable applications to couple input with feedback and function.

Essentially, W^5 presents a way how the designer can look at and talk about PPI. It allows precisely describing the interaction with a digital system by means of digital pen and paper and hence classifying and exploring different interaction techniques. It can also be used to structure the design space and support its exploration, i.e., the systematic discovery and development of new interaction techniques by searching for predicates or combinations thereof that have not been used so far.

[9] http://www.swi-prolog.org/ (accessed: July 2015)

Further predicates will most certainly be required for novel interaction techniques. For instance, one can observe that relative spatial predicates, such as *above*, *below*, *close* etc., have been neglected so far. Furthermore, the core interaction predicates only stem from four out of five core dimensions of W^5. Extending the interaction predicates to incorporate the contextual dimension W_4 (Why) and the originator dimension W_5 (Who) seems promising. For instance, a PPI based application could develop a user ID predicate in the originator dimension to allow users different actions based on their respective access rights.

W^5 can also serve as the basis of PPI and mPPI toolkits as it satisfies the requirements toward conceptual frameworks of interaction derived in chapter 2, section 2.4.2. Thereby, W^5 allows constructing operative, machine understandable descriptions of PPI techniques as demonstrated in the *proof-of-concept* evaluation in section 4.3. Thus, W^5 provides the connecting element of interaction design and supporting infrastructure for mobile PPI as conceptual basis of toolkits, employing an alternative and interaction-centric approach toward the design of infrastructure components. However, in its current form it focuses on the pen and paper aspects of mobile PPI alone: further research will be needed toward interaction predicates expressing the combination of pen and touch, e.g., as prevalent in hybrid mPPI ensembles, as well as other mobile interaction concepts, e.g., embodied interaction.

5 Theory of Mobile Pen-and-Paper Interaction

Synopsis: *Interaction design lies at the core of every usable system in general and every mobile PPI based system in particular. This chapter investigates the theoretical foundations of mobile PPI design. It develops an empirically substantiated theory of interaction in hybrid mPPI ensembles comprising digital pen, paper and mobile device. This theory allows generating guidelines aiding interaction designers in the task of designing mobile PPI. Several guidelines are outlined and their practical relevance is demonstrated.*

While infrastructures (chapter 3) support the development of mobile PPI based systems and conceptual frameworks as foundation of mPPI toolkits (chapter 4) allow expressing interaction between a user and such a system, an important question remains: How should this interaction be designed? What are the essential concepts an interaction designer needs to take into account? What are the key problems users face and how can interaction designers overcome them, i.e., what are the pain-points for developing usable interactive systems employing mPPI and how can they be coped with?

This chapter provides answers to these questions by introducing an empirically substantiated theory of interaction with respect to hybrid mPPI ensembles comprising digital pen, paper and smartphone (c.f., chapter 1, section 1.1.3). Thereby, this theory describes the central concepts in the domain of mPPI design and their interrelations. This allows deriving concrete interaction design guidelines. It also enables interaction designers to determine when and why some interaction techniques should be preferred over others, depending on circumstances. As such, this theory has huge practical impact and can guide the design of compelling and usable PPI based interactive systems in the mobile domain.

The presented theory of interaction was developed based on existing theoretical insights and a qualitative empirical research method substantially inspired by *grounded theory*, [Corbin and Strauss, 1990]. Toward this end, section 5.1 introduces and elaborates on the employed research method which is based on an exploratory, qualitative study to generate theoretical understanding. Section 5.2 covers the study design, apparatus and procedure. Section 5.3 presents the derived theory of interaction in hybrid mPPI ensembles and describes its central constituents, i.e., its *pillars* and *connectors*, as well as the relationships between them. Finally, section 5.4 lays out design guidelines derived from the theory and discusses interaction techniques described in the literature in the light of newly gained theoretical understanding.

5.1 Developing Theory

Currently, hybrid mPPI ensembles lack theoretical understanding explaining specific characteristics and essential concepts to be considered with respect to interaction design. The employed research method therefore bases theory development on empiric, qualitative data generated by an exploratory study using a set of stimulus applications in order to ensure theoretical coverage of essential characteristics and concepts.

The theory development process thereby starts by analyzing the domain and reviewing existing theoretical foundations as described in section 5.1.1 and section 5.1.2 respectively. This yields the initial setup for the exploratory, stimulus driven study to generate qualitative data; which is then subsequently developed into a theory of interaction in hybrid mPPI ensembles following the process outlined in section 5.1.3. Finally, iterative refinement allows investigating important aspects of the emerging theory in depth, as introduced in section 5.1.4.

5.1.1 Understanding the Domain

Studying theory of interaction for hybrid mPPI ensembles needs to take two key characteristics of interaction into account: *Media Transitions* and *Integrated Interaction*. These central characteristics drive the design of any study aimed at developing theory. Given their central role, the remainder of this section aims at introducing and explaining the two terms.

Media Transitions. In hybrid mPPI ensembles, *media transitions* between ensemble components form a distinctive characteristic. The user has to switch between the rather static medium of paper and the dynamic mobile device while interacting with the ensemble. So far, this has been investigated only for stationary settings, e.g., in [Steimle, 2009a]. Media transitions occur in any interaction involving both, digital media accessed within and through the mobile device, and physical media accessed through the digital pen, e.g., paper documents.

Therefore, theoretical understanding of interaction in hybrid mPPI ensembles needs to explain the role and impact of media transitions. Consequently, the theory development approach chosen bases on the sub-portion of the design space where media transitions occur frequently and unmitigated. Strategies to ease the problem of media transitions per se, such as *overlaid information* or *unified interaction devices* (c.f., chapter 2, section 2.5.1), are not included for now.

In this context, overlaid information describes an approach to interaction design that actual projects digital information onto physical information and as such eases or smooths the transition between media, e.g., projection based interfaces as in *PenLight*, [Liao et al., 2010b], where digital information is projected round the pen, *MouseLight*, [Song et al., 2010], where a secondary interaction device is introduced or *AR Lamp*, [Kim et al., 2014], where paper artifacts are overlaid at a certain location on the desk. In contrast to this, unified interaction devices aims to mitigate the transition between media by supporting the same interaction device both for digital and physical media, e.g., using the same pen to interact with paper and a digital device as in *CoScribe*, [Steimle et al., 2008a] or *NiceBook* , [Brandl et al., 2007].

Integrated Interaction. Existing applications for hybrid mPPI ensembles use the mobile device mostly as proxy for paper documents, e.g., to access multi-media content linked to paper documents, [Signer et al., 2006], or to publish sketches to a social network, [Weibel et al., 2010a] (c.f., section 2.5.1 and section 5.1.2 below). The mobile device in employed interaction techniques is restricted to minimal feedback. For example, output on the mobile phone is used to confirm that an action was performed in [Weibel et al., 2010a]. The mobile phone is not used in integration with pen and paper to provide input to an application. Likewise, the *hotspot association gesture* from *ButterflyNet*, [Yeh et al., 2006a], where the user draws two edge marks as a placeholder for a photo to be inserted later, is designed in a way that avoids simultaneous and tightly integrated interaction with pen, paper and digital device.

These approaches do not offer insights in how to design interaction techniques spanning pen, paper and the mobile device in one technique: *integrated interaction* techniques have not been analyzed in depth[1] However, hybrid mPPI ensembles offer the potential to support *integrated interaction*. An example of such an integrated interaction technique for a tabletop setting is described in *CoScribe*, [Steimle, 2009a], as part of a system supporting paper based access and management of learning material. Others have been reported in [Pietrzak et al., 2012] and [Tsandilas, 2012] aiming at hybrid mPPI ensembles.

Thereby, the technique reported in [Steimle, 2009a] enables users to create cross-media links, involving both paper documents and digital documents. The interaction technique was shown to be well accepted by users in a study, which under-

[1] Prior to initial publication of the theory presented in this chapter they had not been analyzed at all. However, Tsandilas and Pietrzak et al. subsequently reported on integrated interaction techniques [Tsandilas, 2012, Pietrzak et al., 2012].

lines the potential of integrated interaction techniques. However, the application in [Steimle, 2009a] is restricted to tabletop computers. The specifics of combining pen and paper with mobile phones are not considered. The integrated interaction technique reported in [Pietrzak et al., 2012] aims at the same scenario, i.e., cross-media links, specifically targeting a hybrid ensemble. The techniques presented in [Tsandilas, 2012] further extend this idea: here users can correct digital ink recognition using integrated interaction techniques.

As a result, any theory of interaction in hybrid mPPI ensembles, needs to explain the characteristics and concepts of integrated interaction techniques.

Other Factors. As argued by Steimle with respect to the related field of PPI based systems for tabletop computers, [Steimle, 2009a], taking an ecological perspective on interaction in hybrid mPPI ensembles becomes important. This is due to user actions occurring in the context of activities and not as isolated, self-sufficient entities. Thus, besides considering digital pen and paper as a technical input medium for mobile devices, theoretical understanding must take the setting in which the interaction takes place into account, the direction of information flow and the users goals. This also includes behavioral components, e.g., the relative placements of devices and other components of the ensemble on a table, the way the user holds the pen or other ensembles components, the user's center of attention, etc.

5.1.2 Extending Existing Theories

Existing work on interaction theory in hybrid mPPI ensembles had been reviewed in chapter 2, section 2.5.2. Basically, existing work studied several applications employing PPI in the mobile domain. Examples are a mobile lab book for field scientists, [Yeh et al., 2006a], access of multi media content via links on paper documents, [Signer et al., 2006], and publishing sketches and handwriting to a social network, [Cowan et al., 2011]. However, theoretical understanding of the domain so far had been neglected. As a consequence, existing theoretical work for PPI in non-mobile settings forms the starting point of theory development for the mobile use case.

Hybrid mPPI Ensembles vs. Tabletops. As reported in section 2.5.2, Steimle emphasized the lack of theory for PPI and developed a set of frequently occurring core interactions, [Steimle, 2009a]. However, the considered stationary setting with a tabletop does not take the specific characteristics of hybrid mPPI ensembles into account: relative placement of components, the interaction capabilities and physical form factors of mobile devices, as well as the mobile setting in terms of user activities (as opposed to a stationary one).

The exploratory study on interaction in device ensembles comprising tabletop computers and pens reported by Hinckley et. al. presented several design considerations for this setting, [Hinckley et al., 2010]. In particular it suggested to combine pen and touch input in novel interaction techniques. This also applies to hybrid mPPI ensembles. However, the mobile setting and the physical properties of mobile device and paper introduce additional aspects: the (relative) physical placement and the limited input / output capabilities of the encompassed components, as well as *media transitions*, i.e., switching between a physical medium (paper) and a digital medium (mobile device) during interaction.

The hybrid mPPI ensemble here differs in several aspects. Obviously, the multi-touch enabled area on a mobile phone is much smaller in size compared to the one of a tabletop computer, which makes the use of other interaction techniques necessary. At the same time, the small form factor of the mobile device coupled with sensor technology permits its use as a tangible device that can be manipulated in three dimensions; this cannot be done with a tabletop computer. Also the physical medium of paper which plays an important role in hybrid mPPI ensembles is not capable of displaying any instantaneous feedback beyond the ink traces of the pen. It can, however, be moved and arranged in a physical way, unlike the tabletop screen. Here, the question of appropriate feedback design in hybrid mPPI ensembles is raised leading to a particular focus on feedback in the setting studied in this chapter (c.f., section 5.1.4).

5.1.3 Research Method

As outlined in section 5.1.2, existing theories focus on PPI stationary settings. Developing a theory of interaction for hybrid mPPI ensembles, however, requires taking the particular aspects of mobile settings and hybrid mPPI ensembles into account. It was therefore decided to develop an interaction theory for hybrid mPPI ensembles ”from scratch”. Toward this end an iterative, qualitative approach to theory development was chosen. This approach derives theoretical insights from empirical data and was substantially inspired by *grounded theory*, [Corbin and Strauss, 1990], a widespread research method in human computer interaction [Lazar et al., 2010].

Thereby, the research method was adapted in the light of domain characteristics and existing theoretic insights as discussed in section 5.1.1 and section 5.1.2 respectively, as well as necessities of the experimental setup. Most crucially, the theory generation method deviates from grounded theory by focusing on several central concepts and their relations, as opposed to a single core category. This is due to existing work pointing to the interconnected role of media transitions and feedback. The experimental setup is also based on pre-existing theoretic concepts, i.e., media transitions, feedback and integrated interaction techniques, an ”informed guess” with respect to

Figure 5.1: Selective coding of domain concepts in order to derive theory.

stimulation of interesting phenomena if you will. This yields the following iterative research method:

(i) The research method used for developing a theory of interaction in hybrid mPPI ensembles starts by designing and implementing a stimulus to obtain empirical data based on existing theoretic concepts. This stimulus triggers data generation regarding the two distinguishing concepts of hybrid mPPI ensembles as outlined in section 5.1.1: *media transitions* and *integrated interaction* techniques, the latter being based on theoretical insights gathered from the stationary domain as discussed above in section 5.1.2.

(ii) After collecting data obtained through the stimulus, this data is analyzed in order to identify the main concepts of the domain and their interrelations. Analysis thereby employs an open, axial, selective coding approach by independent coders, similar to the analysis method suggested in grounded theory. In this approach, open coding attaches an open set of semantic labels to parts of the obtained data, e.g., observations, aiming to identify emerging patterns. Axial coding further classifies relations among these labels. Finally, selective coding identifies the most essential codes observed and derives theory based on these essential concepts. Fig. 5.1 depicts the clustering of codes during the selective coding process of obtained data, illustrating identification and refinement of the emerging theory.

(iii) In order to refine the emerging theory, analysis is performed iteratively, employing micro-iterations (short circles of coding of different parts of the obtained data). Thereby, the emerging theory is refined with respect to the main phenomena and their interrelations until theoretical saturation is reached, i.e., no new codes emerge from analyzing new data. This iterative qualitative approach allows deepening the investigation on relevant parts of the data and obtaining more meaningful results.

(iv) Following this, the role of *Feedback* is investigated in particular using a second macro-iteration (c.f., section 5.1.4 below). Here, additional data is generated in order to obtain a deeper understanding of this domain concept, as both, related work with respect to PPI theory and the experimental findings obtained, highlight its importance.

Generating Data

As outlined above, the two distinguishing concepts of hybrid mPPI ensembles and hence the starting point for generating data were *media transitions* and *integrated interaction*. Here the question is how these two concepts interrelate and how the three possible directions of media transitions affect tightly integrated interaction. Thus, the stimulus design compares a baseline interaction technique described in the literature to an integrated interaction technique designed to deliver the same functionality in order to generate data. Thereby, it varies over all three directions of media transitions, as outlined below.

Media Transitions. As outlined in section 5.1.1, media transitions between digital devices and paper are a central characteristic of hybrid mPPI ensembles. Here, the focus lies on media transitions in the course of interaction with an application. There are three possible directions of transitions

$P \rightarrow M$ (from paper to mobile device)

$P \leftarrow M$ (from mobile device to paper)

$P \leftrightarrow M$ (bi-directional)

Based on the central directions of transitions, the stimulus design consists of three small example applications stipulating these media transitions.

Each example application offers a baseline interaction technique to control its central functionality. Thereby, these baseline interaction techniques are designed to avoid media transitions within a single course of actions. The baseline interaction techniques employed in the exploratory study described in section 5.2 stem from the literature. They encompass the "hotspot association gesture" used in *ButterflyNet*, [Yeh et al., 2006a], Pidgets as used in *PaperPoint*, [Signer and Norrie, 2007b], and the cross media link technique used in *CoScribe*, [Steimle, 2009a].

Integrated Interaction. The study compares baseline interaction techniques to novel *integrated interaction* techniques described in section 5.2, in order to assess the impact of media transitions within a single course of actions. Additionally, novel interaction techniques make extensive use of the unique characteristics of the hybrid mPPI ensemble; for instance, that the small form-factor of the mobile device allows for one handed manipulation, while the other hand simultaneously handles the pen.

In these novel interaction techniques the mobile phone is used like a tangible device. This has, e.g., been studied in [Hudson et al., 2010], where low-attention interaction techniques are proposed, and in [Edge and Blackwell, 2009] with a focus on bi- manual interaction. However, none of the existing studies from that area take into account the pen as an additional interaction device.

In summary, the main aim of comparing existing techniques to newly designed ones is to stipulate the media transitions in all three directions within a single flow of interaction and investigate its impact on the user. Such a setting has not been studied in previous work.

5.1.4 Iterative Refinement: Spotlight on Feedback

As laid out above, the methodology chosen includes an iterative refinement step (iv) in order to put an additional emphasis on feedback. This is due to feedback design providing a central challenge in PPI systems as a result of media heterogeneity, [Liao and Guimbretièere, 2012]. In hybrid mPPI ensembles this is further complicated by having to design feedback in the face of distribution (between mobile device and paper): feedback presented on the mobile device can refer to actions carried out with the digital pen on paper, yet both components of the ensemble are mobile, i.e., can be moved around and rearranged arbitrarily.

In this context, the term *feedback* refers to *system responses as result of user actions*. Thereby, feedback forms an elementary part of interaction techniques as these consist of user actions combined with appropriate feedback [Hinckley, 2007] (c.f., section 2.4.1).

Challenges. Providing visual feedback for interaction carried out with a pen on a secondary screen has been suggested by several authors, e.g., [Cowan et al., 2011], as a solution to remedy the otherwise very limited output capabilities of the pen. However, no conclusive results and design guidelines exist so far. Furthermore, designing feedback in hybrid mPPI ensembles and mobile settings is not straightforward, [Witt et al., 2008]. In particular, it is incorrect to assume that *more* feedback equals better user experience. Unforeseen dependencies between user tasks and the provided feedback may exist.

A particular interesting aspect explored in this chapter is that all devices in a hybrid mPPI ensemble are tangible objects: their arrangement can be changed dynamically by the user while interacting with the system. It has been found for other settings that the physical position of interaction devices has a meaning to the user, and that users implicitly and explicitly arrange interaction devices and artifacts in their workspace. For instance, Scott et al. found that humans in collaborative group work distinguish three main locations for placing artifacts on a tabletop surface: personal area, storage area and group area, [Scott et al., 2004]. Previous work on feedback did not explicitly investigate the implications of this fact, but rather concentrated on one single type of visual feedback, in the center of attention or on a secondary screen, outside the center of attention.

Feedback can be provided in the center of attention, e.g., by overlaying information on paper via projection [Song et al., 2010, Liao et al., 2010b]. However, smartphones can also be used outside the focus of attention, i.e., as a peripheral display, [Matthews et al., 2004], by placing them away from the user, or can be used as an *information lens*, [Reilly et al., 2005]. Feedback can also be perceived peripherally. It is not always necessary to guide the users focus toward the feedback. For some interaction techniques it might be even preferable to give feedback outside the focus of attention. Thereby, the tangible nature of the devices in a hybrid mPPI ensemble also has an impact on the design of visual feedback, and how this can be exploited for designing feedback for interaction techniques.

Study Design. Given the important role of feedback, an iterative refinement step (iv) was conducted in order to extend the emerging theory of interaction in hybrid mPPI ensembles with respect to feedback. It consisted of further, independent analysis of data obtained in the first steps (i - iii), followed by a set of expert interviews using conceptual prototypes that based on the insights gathered through the preceding analysis. The research approach chosen with respect to investigating the role of feedback then consisted of two iterations, or ”rounds”:

Round 1 Analysis of the data of the main study with respect to the role and form of feedback.

Round 2 Design critique sessions of paper prototypes with domain experts (PPI designers). Prototypes consisted of different feedback strategies developed based on the insights of Round 1.

This two-fold approach was chosen in order to gather input from users as well as design experts on the suitability of the feedback design considerations. Aim of this iteration was to establish a set of guidelines for designing feedback, in particular covering design of peripheral feedback and design of feedback for changing positions of the mobile device. Furthermore, as will be seen in the results of the first part of the theory development study (steps i - iii), to gain understanding of feedback design in face of distribution, i.e., *media transitions*.

5.2 Exploratory Study

Following the research method outlined in section 5.1, an exploratory study was conducted in order to derive an empirically substantiated theory about how users interact with hybrid mPPI ensembles. In this study, empirical data was generated using a stimulus (c.f., step i in section 5.1.3): three small applications employing interaction techniques for hybrid mPPI ensembles. Thereby, stimulus applications were derived from applications described in the literature. This section describes the stimulus, study design, procedure of data collection and analysis of collected data.

5.2.1 Stimulus

As outlined above, the stimulus developed in step (i) consisted of three applications modeled after examples from literature. Individual applications, their functionality and references to related work describing such applications is shown in Tab. 5.1. Applications were selected, because they stimulate all possible directions of media transitions in hybrid mPPI ensembles as introduced in section 5.1.3. Thereby, the three stimuli revolve around the following transitions:

$P \rightarrow M$ From paper to mobile device: this transition means, that functionality on the mobile device itself is directly operated on paper

$P \leftarrow M$ From mobile device to paper: Here, functionality associated with the contents of the paper document is accessed via the mobile devices

$P \leftrightarrow M$ Bi-directional: here the flow of interaction goes both ways, i.e., functionality affects the mobile device's functionality as well as those attached to the paper documents.

Name	Description	Reference
Photo (A)	Designing a page of a photo scrap book, containing written text on paper and a digital photograph.	This application resembles ButterflyNet [Yeh et al., 2006a], where written text in lab-books was combined with digital photos. We chose scrap book framing to help participants grasp the idea more quickly as this was more natural to our participants
Draw (B)	Drawing a picture on paper and storing it digitally.	In this application, the interaction is comparable to the one in [Cowan et al., 2011]. Although the sketch in our setting was not posted to an online social networking site, this had no effect on the interaction.
Link (C)	Cross-media links between digital and physical documents.	This application draws from the cross media links between digital and physical documents as described in [Steimle, 2009a], or, with respect to the mobile domain in [Pietrzak et al., 2012].

Table 5.1: The three stimulus applications of the exploratory study.

Participants had to execute small tasks using each of these applications under varying conditions. Thereby, each application offered two interaction techniques in order to control its main functionality: a *baseline* and an *integrated* interaction technique. As laid out in section 5.1, this aimed to particularly stimulate phenomena related to media transitions and integrated interaction.

For each *baseline* interaction technique, techniques described in the literature were adapted to the applications. As laid out in section 5.1.3, baseline interaction techniques also stem from the literature. However, it must be pointed out that these interaction techniques are not necessarily the interaction techniques suggested by the authors of the original applications, i.e., baseline interaction techniques and application scenarios are not necessarily described in the same source (see below for a full list).

For each *integrated* interaction technique, novel interaction techniques were devised that rely on the particular characteristics of hybrid mPPI ensembles, e.g., the physical form factor of the mobile device. These techniques introduce the *media transition* within the flow of interaction and employ the concept of *integrated interaction* (c.f., section 5.1.3). Contrasting the novel techniques to existing techniques from the literature enabled the study to explore differences, respective merits and drawbacks.

Based on the three main directions of media transitions, baseline interaction techniques and novel, integrated interaction techniques, participants of the exploratory study were given a set of tasks to carry out. As described below, these tasks were carried out in a *latin-square* like design, [Grant, 1948], both with the *baseline* and the *integrated* interaction technique as conditions, for each application.

Additionally, data on the role of feedback was elicited as will be explained below in section 5.2.3. For the applications *Photo* and *Draw*, improved versions of the techniques from the literature were studied in separate settings that added instant feedback on the mobile device as opposed to the interaction technique in the original system. The rationale behind this, was to enable investigating whether any observed differences actually stem from integrated interaction, or whether the distinguishing factor would rather be feedback provided on the mobile device.

Based on these considerations, the settings employed in the exploratory study were as follows:

Photo (A) The direction of the media transition stimulated within this application is $P \rightarrow M$ ("from paper to phone").

In order to place a photo on paper, the user draws a gesture consisting of two corners of an imagined image frame, e.g., upper left and lower right corner. A temporal association then inserts a the most recent photo taken (using the mobile device) into the digital facsimile recorded on the mobile device. This constitutes the baseline interaction technique and has been described as the *hotspot association gesture* in *ButterflyNet*, [Yeh et al., 2006a]. Two variations of this baseline technique were examined, one with feedback of the written content including the photo on the mobile device (A2), one without such feedback (A1).

In the integrated interaction technique (A3) the mobile device's camera is directly controlled using the digital pen, instead of pressing the trigger on the mobile device itself. Upon pressing the pen tip to the paper surface the camera's auto focus is initiated, when the user lifts the pen tip, a picture is taken. In the integrated interaction technique (A3), the pen serves as a *remote control* device for the mobile phone. The same feedback as in (A2) was used, i.e., the user had a simultaneous display of the facsimile of her paper document including integrated photos.

Draw (B) The direction of the media transition stimulated by this application is $P \leftarrow M$ ("from phone to paper").

In the baseline interaction technique, stroke width and color are controlled with a printed palette while drawing on paper. Thus, the baseline interaction technique were standard Pidgets (c.f., section 2.4.1) as, e.g., in *PaperPoint*, [Signer and Norrie, 2007b], or *NiceBook*, [Brandl et al., 2010]. In setting (B1) no additional feedback was provided. In setting (B2) a facsimile of the drawing was shown on the phone including the markup the user had specified, i.e., showing the color and stroke width as they had been selected by the user.

In the integrated interaction technique (B3) the user draws with the pen on paper and selects stroke width and color using a palette on the phone via touch. A facsimile and the selected stroke-width and color was provided as feedback here (B3). Additionally, a combination of (B2) and (B3) was investigated. In this (B4) technique the user is free to select stroke width and color either using a palette on the mobile device, as in (B3), or using the Pidget palette printed on paper, as in (B2). This setting employs the same feedback as in (B3), i.e., a colorized facsimile of the digital ink.

Link (C) This application stimulates bi-directional media transitions, i.e., $P \leftrightarrow M$.

Here, the user creates cross media-links through marks on a paper document linking to Websites displayed on the mobile device. To create a link, the user first draws a vertical stroke on paper and subsequently draws a stroke on a Web page displayed on the mobile device. This constitutes the baseline interaction technique (C1). It was adapted from cross-media linking techniques described in *CoScribe*, [Steimle, 2009a]). However, the stroke on the mobile device had to be performed via touch.

In the integrated interaction technique (C2) the user bi-manually creates a link using a *zip* technique: holding the mobile device in one hand, lowering its side to the position where the link is to be created, tilting it by an angle exceeding $45°$, c.f., step (1) in Fig. 5.2, and finally drawing a line along the edge of the mobile device, c.f., step (2) in Fig. 5.2. This corresponds to a zipper stitching the mobile device's content to the location on paper, hence its name. A successfully established link is then visualized by a smooth slide of the Website "out of the mobile device".

Apparatus Applications were implemented in Java on the Android 2.1 platform for mobile devices[2]. Interaction techniques were implemented using Anoto digital pen technology[3] in combination with *Letras* and its *MobiLetras* extension for use on mobile platforms (c.f., chapter 3) to support digital pens smartphones. All applications were deployed and tested, as well as used in the exploratory study, on a Motorola

[2] http://developer.android.com/ (accessed: July 2015)

[3] http://www.anoto.com (accessed: July 2015)

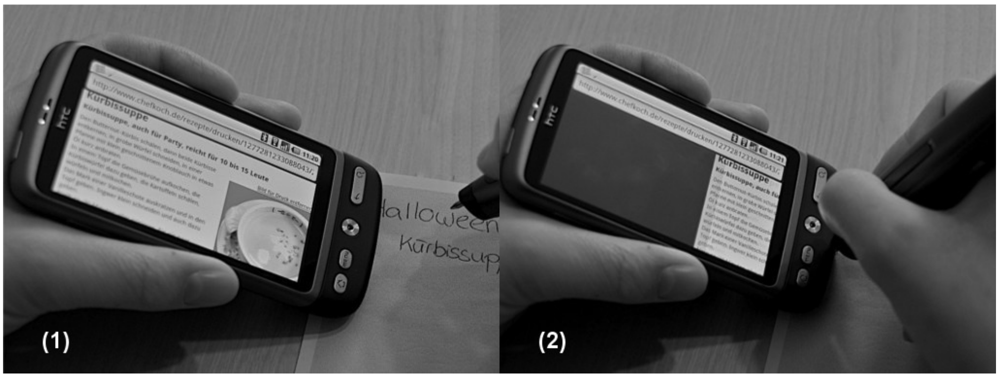


Figure 5.2: The Zip interaction technique for creating links: (1) the user bends the smartphone (> $45°$), (2) the user draws a line along the edge of the phone

Milestone Android smartphone[4]. The accelerometer and other sensors on the phone were used to detect how the phone was physically manipulated by the user, e.g., to detect the direction of the tilt in the link interaction technique. The interaction design toolkit based on W^5 (c.f., chapter 4) was used in order to develop the different baseline and integrated interaction techniques described above.

5.2.2 Design and Methodology

The study employed a within-subject design. Thereby, twelve test subjects (6m/6f) participated in the study. Participants were selected using a snowball sampling technique. Participant ages reached from 24 to 57 years ($M = 29.33$, median 26). Participants had widely varying levels of expertise using digital pens and hybrid mPPI ensembles. Three participants had worked with digital pen technology and hybrid mPPI ensembles before, even as developers. The other participants were completely novice and had never used hybrid mPPI ensembles before; one participant even described the technology as having a ”magic pen”.

The study was conducted in an experimental setting of nomadic interaction, i.e., users interacted with the system while sitting at a table. Participants interacted with all three stimulus applications. Thereby, they were application-wise exposed to all 9 settings and given small tasks to perform, e.g., to draw an image of a house in setting (B2). Their actions were recorded on camera. The order of applications and the order of the interaction techniques within the applications was randomized using

[4] Applications were also deployed and tested on a Samsung Nexus S smartphone, however, the Motorola milestone was used during the study

a latin-square like design [Grant, 1948]. After having accomplished all tasks for all settings related to each application, e.g., all (A) settings, participants were engaged in a semi-structured interview.

Experiment Walkthrough The experimenter explained the nature of the experiment and ensured that participants understood the experiment and agreed to participating in the experiment. For each setting, the experimenter explained the application and the interaction technique with which the respective next task could be performed, e.g., for (C1) the cross-media hyperlink from *CoScribe* [Steimle, 2009a] whereas for (C2) the *zip* technique explained above. The participant then got up to five minutes to try out the interaction technique on an empty sheet of paper. Then, the experimenter restarted the application and asked the participant to complete a task using the interaction technique associated with the condition at hand. The participants were filmed while solving the task. This was repeated for all other settings for a particular application.

Retrospective Thinking Aloud Sessions In order to enable better triangulation, additional retrospective thinking aloud sessions were recorded. In these sessions, the recorded footage of a participant was replayed to that participant and he or she was asked to comment on his or her actions, especially to name difficulties and thought processes. Comments and explanations given by the participant then were recorded in an additional video containing verbal commentary of the user and the experimenter, who asked questions. This aimed to discover thought processes which could not be captured by the camera when executing the tasks. A retrospective thinking aloud session was chosen over an ordinary thinking aloud session, i.e., where the user would comment directly while executing a task, in order to minimize the skew of timing measurements etc. and to avoid distracting the user while performing the task.

Each session lasted about 120 minutes. As a result of the study 9h (9h13m2sec) video footage were obtained. The data was coded by two independent coders using an iterative open, axial and selective coding approach substantially inspired by grounded theory (c.f., section 5.1.3).

5.2.3 Iterative Refinement: Re-Analysis and Expert Interviews

After completing the study (ii) and subsequent iterative analysis of collected data (iii), the theory of interaction was developed as introduced in section 5.3 below. Following this, a second macro-iteration (iv) was conducted in order to refine the understanding of the role of feedback in this theory, as laid out in section 5.1. This second iteration consisted of an independent re-evaluation of data obtained in the stimulus driven exploratory study (Round 1), this time focusing on the particular aspects of feedback in hybrid mPPI ensembles; followed by a series of expert interviews and design critiques of a set of newly developed paper prototypes applying newly obtained theoretic insights in a concrete user interface prototype (Round 2).

Round 1: Re-Analysis of Data. In this first round of step (iv), video recordings of the initial experiment were re-analyzed with respect to feedback related phenomena. Analysis thereby followed the same approach as laid out above and consisted of open, axial and selective coding steps executed by two independent coders. The data obtained during the exploratory study was described in detail throughout sections 5.2.1 and 5.2.2. Here, the employed feedback mechanisms and the different roles of feedback used in the stimulus applications are highlighted (as this was not part of the initial study description).

Essentially, the *role* of feedback in hybrid mPPI ensembles determines what information is conveyed to the user and its meaning with respect to interaction. Thereby, the stimulus applications covered three different roles feedback plays in hybrid mPPI ensembles:

content feedback Content feedback visualizes created content, e.g., a facsimile of the digital ink being drawn.

mode feedback Mode feedback informs the user about the currently selected mode with respect to the application, e.g., current color and stroke width of the pen tool in the digital representation.

action feedback Action feedback informs about the status of performed actions, e.g., recognition of partial completion of a chained interaction technique.

The *Photo (A)* application relied on visual *content feedback* combined with limited acoustic *action feedback*. In the first setting of the baseline technique (A1), the employed feedback mechanism during drawing and writing was limited to acoustic action feedback after inserting images, i.e., playing a ”click” sound after the system recognized an image insertion gesture. In the second setting of the baseline technique (A2), a continuous visual content feedback while drawing was added, i.e., users could see the facsimile of digital ink on the mobile device. In the integrated interaction technique setting (A3), the same feedback mechanism as in (A2) was used, i.e., acoustic action feedback and visual content feedback.

The *Draw (B)* application relied on visual *content feedback*, as well as visual *mode feedback*. In setting (B1), no feedback at all was provided to the user during drawing; visual content feedback, i.e., the facsimile of digital ink, was provided only after drawing. In setting (B2), continuous visual content feedback was provided showing

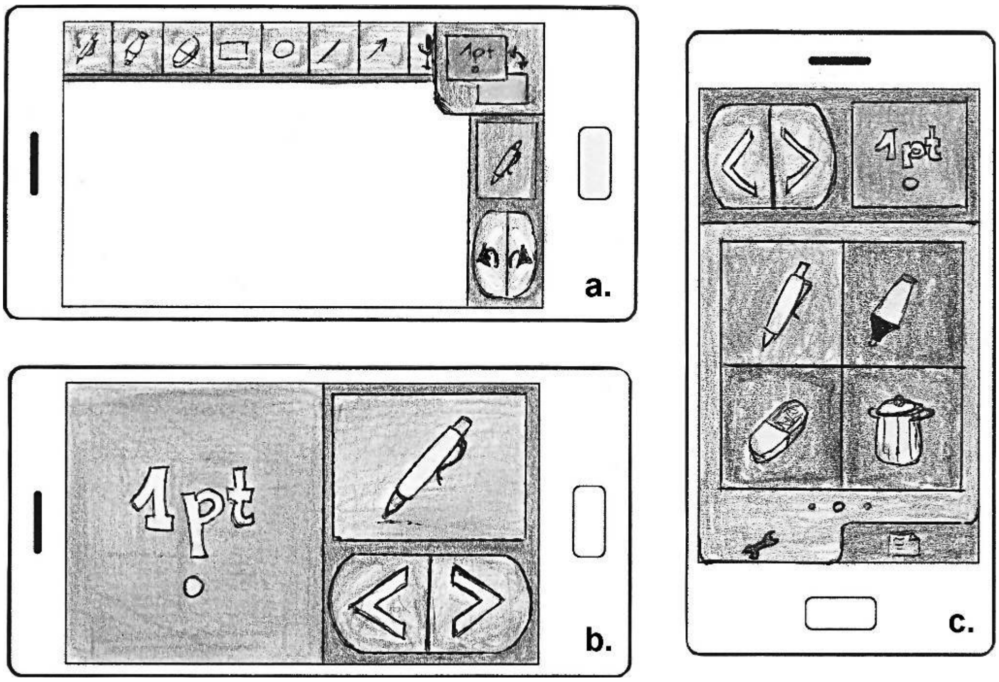


Figure 5.3: Paper prototypes used in the expert interviews: a. attached position, b. detached position, c. peripherally attached position

a facsimile on the phone while drawing. In setting (B3), this visual content feedback was extended by continuous visual mode feedback, i.e., a visual representation of the currently selected stroke width and color. In setting (B4), the same feedback mechanism as (B3) was used, i.e., visual content feedback and visual mode feedback.

The *Link (C)* application relied only on action feedback. In setting (C1), acoustic action feedback was provided after completing the link technique through a gesture on a Web page displayed on the phone. In setting (C2), the visual action feedback was provided as a result of the user drawing a stroke along one side of the smartphone and tilting the device. Here the ”sliding” of the Web page onto paper provided action feedback.

Round 2:Expert Interviews. In this second round of step (iv), the insights obtained as result of re-analysis were refined into design considerations with respect to feedback. Subsequently, these design considerations were applied to a paper prototype of an example note-taking application for hybrid mPPI ensembles. Here a user can draw or write notes on paper using different stroke widths and colors, similar to application (B) in the exploratory study. Based on this, a series of expert interviews was conducted during design critique sessions in order to assess whether the applied design considerations are indeed suitable guidelines for interaction designers

The expert interviews were structured as follows. Five design experts from associated research groups were recruited for participating in design critique sessions. All of them were familiar with the Anoto technology as a user and as a developer, and all of them had previously designed interactive systems for mobile applications. First the experimenter confronted them with the scenario of a hybrid mPPI ensemble and the task to design a drawing application. A sheet of Anoto paper was placed directly in front of the expert on top of the otherwise empty table. The experimenter placed the smartphone into various positions around this paper and asked whether they would design for different types of feedback on the mobile phone depending on its position. Afterwards she presented the paper prototypes for three distinct positions and asked the experts to comment on their designs with respect to feedback.

The paper prototypes presented to the experts are shown in Figure 5.3. They represent the UI shown to the user on the mobile phone. Prototypes are designed based on the observed relationship between how users perceive feedback, where they position their mobile device in relation to paper documents and how much feedback they are capable of digesting. These relations are part of the results of the exploratory study and were analytically derived as explained above. Please refer to section 5.3.4 for details. During the design critique session these paper prototypes were cut out and placed on top of the smartphone depending on its position on the table.

Thereby, design experts gave feedback on the design as such, but also were engaged in discussion regarding the rationale they used in judging the design. These statements were subsequently applied in refining the derived design guidelines. All expert interview sessions were video taped and analyzed as in the initial round of study yielding an additional 2.5h of video footage.

5.3 Theory of Interaction in Hybrid mPPI Ensembles

Following the research method introduced in section 5.1.3, an empirically substantiated theory of interaction for hybrid mPPI ensembles was derived through analysis of collected data, i.e., the results of the exploratory study described throughout section 5.2. The analysis thereby aimed to identify and extract phenomena related to interaction in hybrid mPPI ensembles and to derive a theory explaining the interrelations of the central aspects of interaction in this setting.

Toward this end, the employed research method differed from grounded theory in aiming to identify a set of interrelated *core concepts* and *core relations* as opposed to a

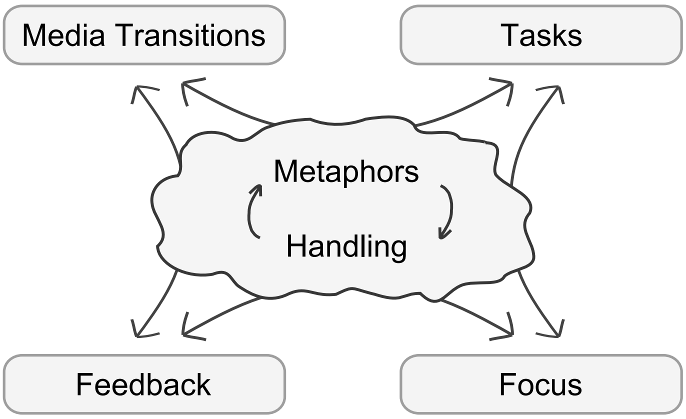


Figure 5.4: Theory of Interaction in Hybrid mPPI Ensembles

single core category (c.f., section 5.1.3). Here, the deductive axial and selective coding steps were applied to codes obtained during the open coding step after classifying these codes into *concepts* and *relations*. This was repeated until a set of core concepts and core relations emerged through clustering and aggregation based on cohesion, i.e., grouping functionally related codes, and theoretical saturation was obtained, i.e., no new codes or aggregation steps emerged from new data. At each time, the explanatory power of the emerging theory with respect to observed data was ensured through re-evaluation and independent coding cycles, i.e., two independent coders executed the steps and results were only included upon consensus.

The derived theory explains interrelations between four core concepts of interaction in hybrid mPPI ensembles. These core concepts derived through analysis are referred to as the *pillars of interaction* in the theory. Thereby, pillars comprise *Media Transitions*, *Tasks*, *Feedback* and *Focus*. Out of these, media transitions and feedback had already been highlighted as central concepts through analysis of related work (c.f., section 5.1 above and chapter 2, section 2.5.2). Here, these concepts were confirmed as core concepts in the derived theory. However, the theory adds tasks and focus as novel core concepts further characterizing and influencing interaction in hybrid mPPI ensembles.

In addition, the theory explains how core concepts interrelate, i.e., what influences concepts exert upon each other and particular phenomena associated with concepts. Toward this end, pillars of interaction are interrelated in the derived theory through two connecting core relations, referred to as the *connectors*. Connectors comprise *Metaphors* and *Handling*. These determine the relations and mutual influence of pillars in the theory. An overview of the observed interrelations is depicted in Fig. 5.4.

In the following, pillars and connectors will be introduced and explained. Additionally, the interrelations between pillars and connectors will be described and discussed in detail.

5.3.1 Pillars of Interaction

As introduced above, pillars describe the core concepts of the domain, i.e., the most important characteristics of hybrid mPPI ensembles directly affecting interaction. Pillars were identified by analyzing both the comments of users regarding their actions, as well as certain problems that occurred during the flow of interaction. The latter can be observed in the video data, e.g., a user trying to change the mode on the phone with the digital pen instead of a touch gesture (a phenomenon referred to during analysis as *interaction device slip*). Thereby, pillars are by no means independent: their interdependencies are affected by the connectors as introduced below in section 5.3.2. The theory encompasses four pillars of interaction: *Media Transitions*, *Tasks*, *Feedback* and *Focus*.

Media Transitions. As explained in section 5.1.1, *Media Transitions* are a distinguishing concept of hybrid mPPI ensembles. This can also be observed in the data: The user switches back and forth between interacting with the mobile device and paper, and between using the pen and her finger as an input device. As such, the media transition occurs both with respect to accessing information, i.e., inspecting paper contents and the digital contents presented on the mobile device, as well as with respect to user actions. This yields further classification of media transitions into

active transitions This media transition occurs when the user starts acting upon another medium and in particular manipulating its contents, e.g., when the users switches from paper to smartphone and actually interacts with the smartphone's contents via touch.

passive transitions This media transition occurs when the user switches to another medium merely consuming and not editing its contents, e.g., when the user switches from paper to smartphone for inspection or comparison of its contents[5].

Naturally, these transitions come at a considerable cost with respect to flow of interaction. However, despite our original idea of media transitions as always being disruptive, some did not disrupt participants and even went unnoticed. This was even

[5] Note that acting upon the other medium is not strictly excluded here, e.g., task as zooming into contents on the smartphone via the ”pinch” touch gesture could still be considered *passive transitions*.

reported for the same task (depending on the circumstances): one participant reported that while inspecting the digital contents on the mobile device did not pose any problem to her, a selection task on the mobile device did [F3:B3/B4]. However, that cannot be explained with the associated costs of changing between pen and multi-touch interaction alone: the same participant interacted with the mobile device via touch while holding the pen in the same hand, a phenomenon we refer to as *multi-use* (c.f., section 5.3.2 *handling*).

The transition performed within the *zip* technique in (C2) (*link*) was done very casually and reported to be *"exactly as if the devices belonged together"* by one participant [F1:C2]. On the other hand, several participants tried to minimize the number of media transitions in the *photo* application. They explained this strategy with the perceived costs of media transitions. Hence, media transitions need to be designed carefully, otherwise problems such as an *interaction device slip* can occur: users will accidentally use the pen on the phone or touch with their fingers on paper, which was observed for 10 out of 12 participants during (B3) (*draw*).

Tasks. This concept describes repeated groups of actions in hybrid mPPI ensembles, which are perceived as a meaningful unit by users. During interview and retrospective thinking aloud, participants mostly referred to their actions in the hybrid mPPI ensemble using tasks as descriptive units, e.g., one participant said about application (A): *"first I took a photo, then I placed it on the paper, then I described its contents [on paper]"* [F2:A]. Whereby taking a photo, placing it on paper and describing the contents are the tasks, which require multiple operations with pen, paper and mobile phone for their execution.

In addition to that, there are several generic activities participants repeatedly executed with respect to interaction with the hybrid mPPI ensemble itself (in contrast to actions with respect to content, e.g., copy, tag, ink etc. as described by Steimle, [Steimle, 2009a]). These encompass:

inspect Users inspect mobile device contents and whether these are in the desired state; users also inspect the paper document contents to review their actions and compare paper documents and mobile device contents.

select Users select functionality (on the mobile device, on paper); if multi step interaction was required this could be interpreted as a control task encompassing multiple elementary functionality selections.

pose Users position the mobile device, paper or pen to a specific, dedicated position or layout, rearranging ensemble components to suit their current needs.

These recurring activities were also referred to, but rather a a sub-unit of the tasks users executed, suggesting that users do not perceive them as unrelated to a particular task (although they occur in every scenario independently).

Feedback. As originally assumed, this is one of the most important components in interaction. In hybrid mPPI ensembles it specifically has to deal with distribution, e.g., feedback on the mobile device refers to actions on paper. Relevant types of feedback discussed by participants were content feedback (current document) and modal feedback (current mode of interaction). Action feedback was not discussed by users, however, its absence had a negative impact on the perceived cost of interaction in some cases.

Most importantly, feedback must match the user's needs. For instance, in the drawing application, participants explicitly requested feedback on the *mode* of interaction (modal feedback) during drawing (*"I liked it [M1:B4], because I had a feedback on the screen about the current mode I'm in"*). In contrast to this, participants requested *content* feedback during inspection of the digital document contents. However, too much or redundant feedback severely confused participants and thus hinders the flow of interaction. Thereby, two essential aspects of feedback in hybrid mPPI ensembles can be distinguished

type The type of feedback in hybrid mPPI ensembles refers to the relation between information conveyed as feedback and the information conveyed through generated physical ink, e.g., whether conveyed information is redundant or complimentary[6]

role As defined on page 192, the role of feedback determines the information conveyed to the user and its meaning with respect to interaction; the three fundamental roles thereby are *content feedback*, *mode feedback* and *action feedback*.

Problems with respect to feedback occurred particularly around media transitions and focus switches. For example, in application (A) users adopted economization strategies to actively avoid media transitions and thus focus switches: in (A1) users would alternate between first taking a picture and then starting to write, and the using opposite order, i.e., first starting to write and then taking a picture [e.g., F1,F4, M1,M6]. Here the insertion gesture naturally occurred within the writing stage (focus on paper). The approach of writing first and then inserting the picture, however, resulted in inserting the wrong picture into the document (as the picture to be inserted was not yet taken).

[6] Section 5.3.4 below further elaborates on different types of feedback in hybrid mPPI ensembles and their relationship to role and other aspects.

Similarly, in application (B) users checked selected color and stroke on the mobile device. Again the media transition occurred from paper to mobile device, with short temporary focus switches (c.f., *check* perception level in section 5.3.4). Here, two types of errors related to presented feedback could be observed: *modus recall errors* and *modus selection errors*. Thereby, a *modus recall error* occurs if the user, e.g., chooses a stroke width or color that was already activated. Likewise, a *modus selection error* occurs, when the user issued a selection command more than once. Interestingly, additional feedback did not reduce these errors. In fact, those errors were observed more often in the combined feedback setting. In the retrospective thinking aloud analysis several users reported that modal feedback alone would have been sufficient and less confusing.

Furthermore, problems such as *feedback oblivion* can occur in hybrid mPPI ensembles: the user does not notice a given type of feedback although it is actually present, *"oh this [modal feedback], I would have needed this at the time!"* [M3: B3/B4 also M2]. Also, feedback must be presented in a way the user can actually perceive and digest it. For instance, visual feedback was sometimes presented to small in the exploratory study in order for the participants to actually perceive it, e.g., as reported by [M3:B2]. Also, acoustic feedback was occasionally to short and volatile for the user to notice. Thus, the role of this pillar and associated phenomena were were further investigated in the second iteration directly aimed at obtaining insights on feedback (c.f., section 5.3.4 below).

Focus. Focus plays a distinctive role in hybrid mPPI ensembles. In this context, *focus* refers to the area of visual attention [Proctor and Wu, 2007]. Participants explicitly dealt with focus by re-arranging the mobile device close to paper or even hovering it on top of paper, thereby creating a *joint focus zone*. We observed pen, paper and mobile device to be constantly rearranged during interaction (*"I [positioned this] always in a way, that I do not have to switch around with my eyes"* [F1: referring to A/B]). Devices can unintentionally get out of focus due to accidental repositioning, e.g., pushing the mobile device aside while writing on paper [M6:A2]. Thus, focus must be carefully guided taking into account the main focus and the peripheral area.

main focus This refers to the area users currently focus and look at, i.e., the central area of visual attention, [Proctor and Wu, 2007]. Besides visual attention alone, this also indicates cognitive attention and perceptual motor attention, i.e., it characterizes the place where interaction currently occurs.

peripheral area This refers to an area outside of the main focus, where (visual) perception is still possible although the user attention lies elsewhere (i.e., on the main focus). Here, perceptional capabilities are limited albeit present (see also section 5.3.4 for a discussion of implications with respect to feedback).

Interestingly, switching focus back and forth between media (c.f., *media transitions* as described above) within a single (micro-)task leads to the *split focus* phenomenon: users loose time on the way while switching the focus, as a consequence loose trail of thought and subsequently make mistakes. Generally this is perceived as very inconvenient as indicated by the interviewed users, e.g., as reported by [F1, M1:A]. The split focus problem becomes eminent especially in cases where very fine grained feedback is relevant to the task, e.g., drawing in (B2) while the stroke color has been set, but no modal feedback is available.

5.3.2 Connectors between Pillars

Connectors describe the influencing factors with respect to the interrelations between the four pillars. As such, these concepts determine *how* the pillars affect each other. Therefore they become particularly relevant with respect to predictive statements derived from the theory, as explained below in section 5.3.3. The two connectors are *Metaphors* and *Handling*.

Metaphors. Metaphors play a strong inter-connecting role in hybrid mPPI ensembles. Since users build mental models expressed by metaphors, it becomes important that they build the right models, [Norman, 2002]. Metaphors typically involve the components of the hybrid mPPI ensemble, i.e., the digital pen, the mobile device and the paper documents. Common metaphors mentioned by participants referred either to *connection* between components, or their *usage*. Metaphors consistently mentioned by participants within their comments on their own actions in the retrospective thinking aloud sessions were

Gluing The metaphor of *gluing* things together in the physical world provides a central connection metaphor. It occurs in particular in cases where place holders are used to indicate digital physical connections, e.g., in [F1:A:Interview].

Anchoring The metaphor of *anchoring* digital resources on physical locations represents another connecting metaphor, e.g., mentioned in [M1:C:Interview]. This expresses the mental model explaining enforced physical proximity of the mobile device and the paper document as part of integrated interaction techniques.

Ink Pot / Palette The metaphor of an *ink pot*, or a *palette*, forms a usage metaphor. It is employed in situations where the pen is used for selecting a certain mode either on paper, or on the mobile devices itself (comparable to dipping it into an ink pot), e.g., as mentioned in [M1:B4].

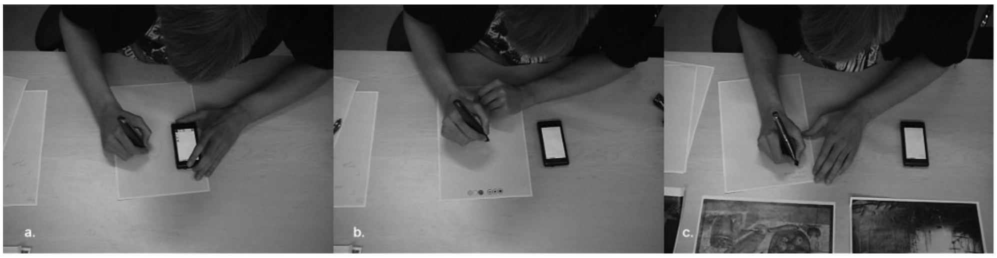


Figure 5.5: Three different placement schemes of components in hybrid mPPI ensembles: a. attached, b. peripherally-attached, c. detached

Ruler The *ruler* metaphor is a usage metaphor for the integrated interaction technique in (C) (*zip*), where a cross-media link is established by drawing a line alongside the edge of the mobile device, [M1:C2]. This shows, that although a metaphor (zipper) was provided as part of the interaction technique, the user forms an independent mental-model and might introduce differing metaphors.

Frame The *frame* metaphor forms a usage metaphor used to describe the hotspot association gesture executed as part of the (A) tasks. This technique is referred to as drawing a frame by several participants, some extending it to describing it as a photo frame [F1, F4: A].

As can be seen, expressed metaphors resembled tools or practices in the physical world and sometimes differ from the intended design metaphors. Thereby, participants used metaphors to describe employed interaction techniques through physical actions in the real world, hinting on their mental models. For instance, connecting the mobile device with paper using a stroke was described by several participants as using the *"mobile device as a ruler"* [M1:C2]; one participant described pen mode selection on the mobile device as *"dipping [the pen] into an ink pot"* [M1:B4].

Handling. Handling refers to how the user manipulates the ensemble: this encompasses relative spatial positioning of ensemble components as well as the grip in which components are held. The physical distribution of components, e.g., paper documents and mobile device, forms a central factor affecting user interaction in hybrid mPPI ensembles, as confirmed by Hong et al. for the domain of stationary knowledge work [Hong et al., 2012]. This yields two central aspects of handling

grip The grip determines the way in which ensemble components are held by the user (if manually supported), in particular with respect to the mobile device and the digital pen, as interactive ensemble components.

placement scheme The placement scheme determines the relative spatial layout of ensemble components on a surface, e.g., where the mobile device is placed with respect to a paper document.

Data gathered in the exploratory study showed that participants, often subconsciously, distinguished between three relative spatial positions with respect to the physical distribution of ensemble components. These *placement schemes* describe layouts where the mobile device and paper, were positioned *detached* from each other, *peripherally attached* or *attached* as shown in Fig. 5.5 on page 201 (see section 5.3.4 for a detailed analysis of the role of placement schemes regarding feedback).

Regarding the employed grip, mobile device and pen are either held in an *active* grip, allowing for direct interaction, or in a *passive* grip. If holding components in a passive grip, participants would sometimes execute another action with the same hand, a situation we refer to as *multi-use*. These observations confirm recent findings on prospective motor control, i.e., the subconscious planing and optimization of movements by users [Cohen and Rosenbaum, 2011].

Surprisingly, users were able to hold the pen in hand while manipulating the phone with the same hand, and did not even notice this behavior. For instance, one participant executed a pinch gesture to manipulate the view on the mobile device while holding the pen in the same hand without noticing [F1:B]. If both hands are simultaneously used to execute actions, *bi-manual* use occurs. Interestingly, multi-use was combined with bi-manual interaction by several participants.

5.3.3 Relationships between Pillars

Pillars and connectors form the basic constituents of the theory. Thereby, as shown in Fig. 5.4 on page 195, connectors determine the mutual effect of the pillars of interaction on each other, i.e., their interrelations. Those interrelations are depicted by the connecting lines between pillars in Fig. 5.4. This section provides an overview of existing relationships and explains their impact on interaction.

Media Transitions in Tasks If media transitions occur as part of a task, e.g., taking a photo or drawing a diagram, the impact of the transition on the flow of interaction – its disruptiveness – is partly determined by the handling required as part of the transition, partly by the usage metaphor. Metaphors, such as sliding the Web page from the phone to paper in combination with appropriate handling, as applied in the *zip* interaction technique (C2), can explain that media boundaries are transcended. Consequently, explained transitions were perceived as less disruptive by participants in the exploratory study as revealed by their commentary in the retrospective thinking aloud sessions.

Feedback follows Focus The perception of feedback follows its spatial position relative to the current main focus area. Metaphors therefore play a role in order to convey where to expect which type of feedback and thus to guide the focus. However, the relative spatial position (handling) determines whether high resolution feedback can be shown (main focus) or lower resolution feedback is required (peripheral area), or whether feedback is perceived at all. In particular, the perception of feedback depends on the employed placement scheme as it indicates where the user focus currently lies (c.f., section 5.3.4).

Feedback with Media Transitions If feedback is presented on another medium, its perception is influenced by metaphors and the spatial placement of components in the hybrid mPPI ensemble, i.e., by the currently employed *placement scheme*. The same holds for feedback for media transitions. If the metaphor provided by the feedback corresponds to the *direction* of transition, e.g., mobile device to paper, the feedback will reduce the cost of the media transition. Otherwise it will simply confuse users.

Focus for Tasks The focus during a task is induced by the spatial position (handling) and the salient metaphors. Users will naturally place relevant components into the focus if they understand that these are relevant. On the other hand, focus could be enforced by introducing interactions that will place relevant components into focus: if the user has to draw a stroke on the side of the mobile device (*zip* technique) in order to get access to required functionality, than this will remain inside the focus automatically and thus increase the convenience of interaction.

5.3.4 Iterative Refinement: Feedback in Hybrid mPPI Ensembles

In accordance with the research methodology outlined in section 5.1.4, a subsequent, second analysis of the data obtained within the exploratory study was conducted with respect to feedback related phenomena as part of step (iv) (c.f., section 5.1.3). This was based on the fact that the initial theory development round highlighted the particular importance of feedback.

Following this re-analysis of experiment data (Round 1), a series of expert interviews was conducted and their results were similarly analyzed (Round 2), in order to gradually refine the theory of interaction in hybrid mPPI ensembles with respect to feedback. This refinement further explains the role of feedback and the most important concepts associated with it. It also provides additional guidance to interaction designers when it comes to explaining how to adapt and provide feedback in hybrid mPPI ensembles.

With respect to feedback for actions carried out with the digital pen in hybrid mPPI ensembles, the main relationship can be stated as:

> The main *placement scheme* and *perception level* directly affect the *bandwith* of perceived feedback. Feedback *type* and *role* indicate how feedback can be adapted to provide required information given the current placement scheme and perception level.

Bandwith. In this context, *bandwith* refers to the total amount of information that can be perceived and digested by the user. If the provided amount of information given in the feedback exceeds the bandwith, phenomena such as *feedback oblivion* will occur: users will simply not be aware of (part of) the provided information. Sometimes, users even consciously ignore feedback in these cases, as one participant explained: *"here I [ignored] the [facsimile], as I considered it confusing"* [M5:B4]. This indicates strategies of users to avoid cognitive overload.

Designers need to carefully use the available bandwith in order to provide the "right amount" feedback. Two main strategies were discussed during the expert interviews

Abstraction Feedback abstracts from detailed information by assigning higher-level semantic to provided information (c.f. [Matthews, 2006]). Here designers must ensure that users understand the provided abstractions. The most common example would be tinting a digital display in order to indicate inactivity etc.

Reduction The amount of provided information in the feedback is reduced to match the available bandwith. Here the designer must carefully decide which information is currently needed, in order to avoid missing information. This again is induced by the task at hand.

Placement schemes. As laid out above, *placement schemes* refer to the current relative spatial position between mobile device and the central point of visual attention, i.e., the focus as defined in section 5.3.1. Here three consistently re-occurring placement schemes were observed as shown in the conceptual diagram in Fig. 5.6[7]. Users typically positioned the devices without conscious consideration of the optimal position for the task at hand. One participant described this as *"[placing devices] just as [he] needed them"* [M6:A:Interview].

The placement scheme directly affects the available bandwith: the further away from the focus, i.e., the center of attention, the less bandwith is available. While the

[7] For a video still showing these in actual interaction please refer to Fig. 5.5 on page 201, section 5.3.2 above.

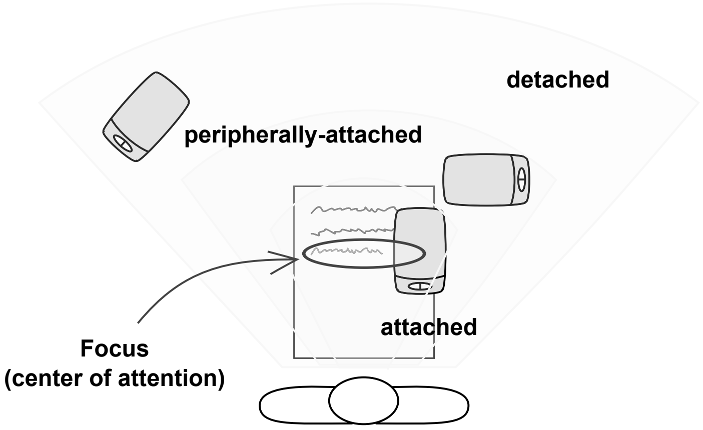


Figure 5.6: Placement schemes and focus (center of attention)

user can perceive even fine-grained feedback in the *attached* position where paper and mobile device form a *joint focus zone*, the *detached* position allows only for minimal feedback. Interestingly, this also holds for non-visual feedback, e.g., the action feedback provided as part of (C1): Users who accidentally placed the mobile device into a detached position, were consistently unaware of provided acoustic action feedback.

Perception level. The *perception level* describes the amount and time of attention users consciously devote in order to perceive feedback. This varies, according to task or ensemble related activity and is also determined by the metaphors / mental model, i.e., were and when users expect feedback to occur. Observation and interviews in our study revealed four distinct perception levels

Sense Users do not concentrate on feedback, nevertheless perceive some low level information, e.g., they sense whether the system records digital ink at all (without being aware of specific contents).

Check Users issue a quick check for required information in the feedback without really switching their focus, e.g., they throw a glance whether the color was changed correctly.

Inspect Users temporarily switch their focus exclusively to the feedback source in order assess whether the system responded correctly, e.g., they look at the digital facsimile and judge its appearance. In this perception level users typically do not interact with provided content feedback, or limit interaction to serve their goal of inspection, e.g, by issuing pinch gesture to zoom into particular areas of interest.

Switch Here users completely switch their focus to the feedback source and remain there for longer periods compared to the *inspect* level (or even remain there and proceed with their actions); here users also start interacting with provided content feedback, e.g., by exporting facsimile to an image and sending it per email.

Perception levels are related to placement schemes. For a detached placement we only observed the *sense* and *switch* levels. In a peripherally attached placement, users would perceive feedback in the *sense*, *check* and sometimes *inspect* level. In an attached placement, users would perceive the feedback continuously, so we observed only the *switch* level directly.

Thereby users tend to adapt the placement scheme in order to enable the required perception level. For instance one participant placed the smartphone during (B4) into a peripherally attached position, *checked* the mode feedback frequently during drawing and finally placed the phone into an attached placement where he *switched* to the digital representation [M2:B4].

However, perception levels also influence the available bandwith directly: shorter focus switches lead to less information that can be digested. For instance, one participant reported in (B4) that *"[he] could not see the complete [feedback], as [he] only checked quickly"* [M4]. Although these perception levels mostly refer to visual perception, auditory perception of feedback seems to follow similar patterns. At least for the *sense* and *check* perception levels this could be observed in our data.

Type and Role. *Role* defines the meaning of feedback with respect to the current user task. As laid out in section 5.2.3, page 192, the role of feedback in hybrid mPPI ensembles is to represent created *content*, indicate the current *mode* of interaction, or to provide a direct response to a particular *action*, e.g., to indicate correct (partial) recognition in chained interactions. The *type* of (digital) feedback refers to its relation with the physical ink trace left on paper during interaction. Three fundamental types can be distinguished

Redundant Redundant feedback provides the same information as the physical ink traces on paper, e.g., a digital facsimile. This typically refers to content feedback.

Extending Extending feedback adds additional information to ink traces on paper, e.g., colored strokes or embedded images.

Complementary Complementary feedback provides information exclusive to the digital representation, e.g., a small pictogram of the currently selected color and stroke width. This typically refers to mode feedback.

Role and type are analytical constructs in order to help designers when reducing or abstracting feedback appropriately. As described in section 5.3.1, too much or the wrong type (and role) of feedback can severely confuse users.

5.4 Interaction Design Guidelines for Hybrid mPPI Ensembles

Hybrid mPPI ensembles allow interaction designers exploring a broad variety of potentially engaging and compelling interactive systems supporting users in mobile or nomadic settings. Aim of this chapter was to establish a deeper understanding on how to actually design interaction in these settings. In this context, the empirically substantiated theory of interaction introduced throughout section 5.3 allows a deeper understanding of important concepts and their interrelations with respect to interaction in hybrid mPPI ensembles. However, the question remains how to actually leverage the theoretical understanding gained, in order to design better, more engaging and fluent interactive systems for hybrid mPPI ensembles.

Toward this end, the present section introduces a small set of concrete design guidelines derived from the theory of interaction. These guidelines demonstrate how the theory contributes to guiding interaction designers. Additionally, they deepen the understanding, explaining why certain interaction techniques work where others fail. Furthermore, this section explains how to design interaction using the particular combination of touch and pen that is unique to hybrid mPPI ensembles. Finally, it presents interaction design guidelines with respect to feedback design and its role in hybrid mPPI ensembles.

5.4.1 From Theory to Practice

Design guidelines derived from the theory of interaction for hybrid mPPI ensembles revolve around the interrelations between pillars as outlined in section 5.3.3. Thereby, the guidelines base on how the interconnecting concepts, the *connectors* (c.f., section 5.3.2), should be used in order to optimize the interaction and which particular aspects of the *pillars* (c.f., section 5.3.1) have to be taken into account.

In the following, this section demonstrates how each major interrelation can be applied to the design of interaction in hybrid mPPI ensembles and elaborates on derived interaction design guidelines.

Design Guideline 1: Guide Media Transitions in Tasks

> *Carefully guide media transitions occurring within tasks through appropriate metaphors and handling. While the former explains the transition to the user, the latter facilitates the actual transition, e.g., through spatial proximity required as part of the interaction.*

Discussion. A good metaphor, e.g. the ”sliding of a web page into the paper” as part of the *zip* technique, combined with adequate handling will result in the media transition within a task to be intuitive. However, if the user does not understand why to transcend media, or worse, the usage metaphor does not account for changing media at all, the media transition can lead to phenomena like the *interaction device slip* and *split perception*. The former thereby describes a situation, where the user accidentally tries to access a medium by using an interaction device associated with another medium. The latter refers to a situation, where the task feels more complicated to the user, e.g., as having more steps than another task with a better metaphor although that is actually not the case. This severely limits the notion of feeling that ”the devices work properly together”. Therefore designers should make sure to explain why to switch media within a task or reduce these switches.

Using this principle, the theory can explain why established interaction techniques work, e.g., the cross-media linking technique reported by Steimle [Steimle, 2009a]. Here the user positions a physical document overlapping a digital document (*Handling*) and issues a stitching gesture (*Metaphor*) in order to link the documents (*Media Transition in Tasks*).

Furthermore, in the setting studied by Tsandilas, i.e., correction and improvement of digital ink recognition through user interaction in hybrid mPPI ensembles, users preferred the *pen + pen* and the *pen + touch* interaction techniques over the *touch + pen* technique, [Tsandilas, 2012]. This is interesting, as the *touch + pen* technique showed *”significant performance benefits”*, [Tsandilas, 2012], with respect to the recognition system. The theory of interaction in hybrid mPPI ensembles allows explaining this paradox: the mental model of users foresees notes being written on paper and subsequently being played on the mobile device, however, the media transition in the *touch + pen* technique (constraining the recognition results prior to the recognition step via touch on the mobile device) remains unexplained. A better metaphor and enforced handling (mobile device position) might have allowed to better align actual system performance and user satisfaction in the given scenario.

Interjection: Media Transitions between Tasks. Note that the concepts introduced above mainly apply to media transitions within tasks, e.g., if the user has to take a picture on the mobile device by pressing the digital pen on paper as in (A3). In between tasks, although reported as less problematic in general, media transitions are harder to guide through metaphors. Also, users vary the sequence of tasks according to their needs which makes it harder for interaction designers to anticipate where and when they might occur. Here the analysis showed that users tend to optimize the sequence of their tasks in order to minimize the (perceived) costs of media transitions. Perception thereby relates again to the employed metaphor or the affordances of a particular medium as defined by [Norman, 2002].

Consider for instance application (A): if two tasks need to be performed, one starting on the mobile device and ending with pen and paper, the other one vice versa, users would think about how these two tasks can be chained so that the media transitions could be avoided. Users did this even though they stated during the retrospective thinking aloud session that they would have preferred to perform the tasks in a different order. This indicates high associated costs regarding the media transition.

Thus, designers should consider for designing the tasks supported by their applications in a way that enables users to freely optimize the sequence of execution as needed. Often, this means that one can first select the instrument and then chose the content to apply it to, and vice versa, an approach commonly described as *phrasing*, [Buxton, 1986]. Alternatively, where this is not possible, designers should aim to enforce the media transition within a given task and offer a proper guiding metaphor, as described above.

Design Guideline 2: Adapt Feedback to Focus

> *Take the current focus into account when designing feedback. Adapt feedback perception to the current placement scheme or use metaphors and handling in order to ensure that feedback is perceived.*

Discussion. Bringing the right feedback to the right place is key in hybrid mPPI ensembles. Being able to provide feedback on the mobile device is a huge benefit of hybrid mPPI ensembles compared to pen and paper alone. However, in contrast to stationary settings, hybrid mPPI ensembles allow the user rearranging the components as needed; which bears the risk of the mobile device being in a position where the user does not easily perceive its contents. At the same time, feedback on the mobile device is useless, if it is not perceived by the user because she focuses on paper.

An example of feedback on the mobile device is the modus and the facsimile in the *draw* application in our study. The metaphor applied by most users was that of a *”palette”* - the mobile device was placed next to the paper. Here, the metaphor suggests that the mobile device will be in the peripheral view, but not in the center of attention. Hence, the feedback needs to be low- bandwidth, so it can be digested without focusing on the mobile device. Accordingly, participants preferred the low-bandwidth feedback over the combined feedback of mode and current drawing.

How feedback can be adapted properly and what designers need to consider when designing feedback will be described below in section 5.4.3.

Design Guideline 3: Feedback for Media Transitions

> *Carefully design feedback in order to guide users through media transitions using appropriate metaphors and handling. Make sure to take the directionality of media transitions into account.*

Discussion. As laid out above, media Transitions are usually disruptive to the user. To minimize this effect, the user should be guided through transitions. A behavior we observed was that users ”got stuck” if they had to perform a media transition to the mobile phone, e.g., to create a link, but the phone was not in the main focus. The *zip* technique applied the *metaphor* of a ruler, which in turn required handling of the mobile device in a way that the screen can be seen while performing the link creation. At the same time such a technique provides an adequate metaphor: the linked artifact ”flows into” the paper.

Here the feedback actually emphasizes the direction of media transition to the user. This makes the subsequent transition more natural, as the user now understands where to proceed. Designers should employ this strategy wherever they have to guide users through media transitions, e.g., between tasks where there are not metaphors available. At the same time, the feedback can also be used to actually describe or reinforce the employed metaphor, e.g., use the visual representation of an ink pot in order to make this concept salient and accessible for users.

Design Guideline 4: Guide Focus in Tasks

> *Guide the focus to match the current task. Use handling to ensure all required ensemble components are in the focus, employ metaphors to explain their function.*

Discussion. The focus with respect to the current task is typically quite limited in hybrid mPPI ensembles. Thus, the actual placement of components should be enforced to match the desired focal area. With respect to this, it should be ensured that all the ensemble components required for executing a certain task are well within the focus area where and when they are needed. This enables the user to perceive information where and when it is needed. In order to achieve this, the designer can introduce management interaction techniques for the hybrid mPPI ensemble, such as the *connecting ruler* metaphor as part of the interaction. As stated above: if the metaphor for "what is to be seen where" can be conveyed to the user, the arrangement and thus the used focus comes naturally.

Interjection: Utilizing Component Layout. Users arrange the ensemble components according to their needs, thereby employing different placement schemes. Furthermore, users tend to optimize the spacial layout during the course of interaction, e.g., to economize on paper real estate by turning paper documents into landscape mode. Therefore, designers should consider relative placement of components and their orientation with respect to interaction design. Both can be used as an active design element, e.g., making portions of a paper document interactive by placing the mobile device on the paper. However, designers also need to consider that the different components may overlap each other, leading to occlusion. Here it must be noted, that although contemporary digital pen technology does not directly support detection of paper orientation designers can easily achieve this by introducing a small calibration step, e.g., by using a not rotation- invariant gesture (such as a *"V"*).

5.4.2 Combining Pen and Touch in Hybrid mPPI Ensembles

Hybrid mPPI ensembles typically encompass at least two modalities: pen and touch. Research has shown that their combinations allow to introduce novel interaction techniques combining the ensemble to a cohesive whole and thus forming novel, integrated interaction techniques, c.f., [Hinckley et al., 2010], [Frisch et al., 2010] and [Matulic and Norrie, 2013]. In this context, the designer might wonder whether it is possible to employ *bi-manual* interaction techniques when combining pen and touch in hybrid mPPI ensembles. Thereby, bi-manual interaction techniques require using both hands in order to trigger functionality. In essence this is possible, however, the design of the interaction technique should closely follow the guidelines laid out in this section.

Design Guideline 5: Bi-manual Interaction

> *Ensure bi-manual interaction occurs only in the center of visual attention and is guided by a convenient metaphor.*

Discussion. Although users interacted with the digital pen using their dominant hand, the non-dominant hand is by no means free for issuing multi-touch gestures: in our study, users typically stabilized the paper document on the surface with their non-dominant hand as, e.g., also reported by [Hong et al., 2012], or held the mobile device in such a way as to improve perception of the digital contents. Carrying out actions while writing or drawing, e.g., the mode switch in (B3) or holding the mobile device as a camera in (A3) was highly disliked, whereas bi-manual interaction in (C2) posed no problem to the users. Therefore, simultaneously combining touch gestures with pen interaction in hybrid mPPI ensembles should be avoided unless both occurs in the center of visual attention and guided by a convenient metaphor, e.g., as in (C2).

Interestingly, although simultaneous actions with pen and touch proved to be problematic (media transition within task, requiring non-dominant hand), an actual media transition for the dominant hand with respect to interaction device in short succession proved to pose no problem at all (media transition between tasks, executable with dominant hand). The experimental data also disproved the initial assumption that users would have to lay down the pen in order to issue touch gestures on the mobile device and that this would be highly disruptive with respect to the flow of interaction.

Design Guideline 6: Multi-use

> *Employ metaphors and handling during tasks in cases where no full-blown media transition is required in order to trigger multi-use. However, make sure to constrain the interaction to simple gestures, i.e., gestures that can be executed on the mobile device while still holding the pen, or design for a complete media transition.*

Discussion. As a matter of fact, users did not lay down the pen at all during tasks, they would rather execute touch gestures with their dominant hand while simultaneously holding the pen in the same hand. This behavior is here referred to as *multi-use* with respect to employed modalities. Whenever users exhibit multi-use behavior, the pen changes to a *passive grip*, i.e., users hold the pen in a way that does not allow writing without changing the grip first. Surprisingly this behavior was very common among participants of the exploratory study (10 out of 12 participants). One participant even issued pinch gestures with a her dominant hand while holding the pen in a passive grip [F1:B4].

However, we never observed multi-use where the mobile phone was passively held in the hand, while shortly switching to the pen. This is most probably due to its larger form factor and weight: the pen is very slender and light-weight, e.g., the approximate weight of the Logitech IO2 Bluetooth pen used in the exploratory study is about $35g$. In contrast to this, the mobile device is typically much heavier and has a bigger form factor, e.g., the Motorola Milestone used in the study weighs approximately $169g$. Furthermore, most people (at least all our participants) are accustomed to handling pens since their childhood, and thus are very skilled at handling the pen with their preferred hand. These factors seem to have prevented occurrences of multi-use with the mobile phone in the exploratory study.

Thus, designers should be aware that users may hold a pen in the hand while interacting via multi-touch with the mobile device, which, e.g., hinders performing pinch gestures (although, as described above, does not prevent them completely). On the other hand, this may be leveraged in the design, as the second hand can still be used for other tasks, e.g., handling paper documents.

5.4.3 Iterative Refinement: Designing Feedback

The theory introduced in section 5.3 states that *feedback follows focus*, i.e., that the perception of feedback is determined by the location where it is given relative to the current main focus area of the user. Thereby metaphors employed by the designer guide the user's focus by explaining where to expect which type of feedback. In addition the relative spatial position of the feedback medium determines the resolution of feedback, e.g., high resolution feedback can be shown in the center of focus whereas in the peripheral area lower resolution feedback should be chosen.

Section 5.3.4 further refined this relationship by establishing that the current placement scheme, e.g., when the mobile device is *peripherally-attached*, and the perception level, e.g., when the user *checks* the mobile devices display, determine the available bandwith, i.e., the amount of feedback that can be digested by the user without risking a cognitive overload. Here it becomes clear how handling (influencing placement scheme) and metaphors (influencing perception level) play a role in the relationship between feedback and focus.

As explained above in section 5.4.1 the designer has to carefully design feedback in hybrid mPPI ensembles in order to avoid insufficient or excessive feedback adversely affecting user experience. The following design guidelines aim to support interaction designers in the quest of bringing the right feedback to the right place to the user.

Design Guideline 7: Use Metaphors and Handling to Adjust Bandwith

> *Use metaphors and handling to control available bandwith and trigger a placement scheme matching the desired perception level of feedback. Metaphors thereby explain the perception level, handling actively controls the placement scheme.*

Discussion. Available bandwith depends on where the mobile device is positioned in relation to paper and how the user accesses feedback. This on the other hand can be controlled by appropriate metaphors and handling. Interaction designers could enforce positioning the mobile device in certain positions. For instance, an interaction technique could require the user to connect the phone to the paper document in order to enforce an attached position thereby employing a *ruler* metaphor (and sensing whether the device has been moved subsequently using built-in sensors). This would increase the bandwith of feedback the user could perceive. In such a setting, the placement scheme would be fixed. However, the designer would also have to make sure that the metaphor conveys the proper perception level to the user. Also, it must be added, that the perception level has to adequately match the user's demands: enforcing a full *switch* where the user prefers to continue working on paper would be highly adverse to a satisfactory and smooth flow of interaction. As such, all of this depends on the metaphor itself, i.e., the user needs to understand what to expect where.

Design Guideline 8: Reduce Feedback to match Bandwith

> *In cases where there is only limited bandwith available, e.g., a writing task on paper, carefully reduce the feedback to match the bandwith. However, ensure that remaining feedback carries sufficient information about system state.*

Discussion. Feedback reduction strategies are an important tool for the designer. The process starts by establishing which role the feedback has with respect to interaction, i.e., whether it is content feedback regarding digital ink recorded, mode feedback indicating the current tool used, e.g., color or stroke width of the pen, or response feedback confirming the recognition of user actions. The next step is to analyze the type of feedback with respect to interaction. Thereby, redundant feedback is the most likely target for reduction, but also extending feedback might be reduced in order to match the users needs. It is, however, imperative to assert that the reduction does not simplify the feedback to a degree where it becomes unintelligible.

Example. As a combined example, consider the interaction technique in (B3). The usage metaphor would be a palette, hence it is most likely to be positioned in a peripherally attached scheme where the user writes and draws on paper while changing the mode of the pen tool on her mobile device. However, as the main task is drawing on paper, the metaphor does not suggest a full switch to the mobile device during the task. This results in perception levels to alternate between check and sense; in consequence the available bandwith is limited.

In order to avoid it being limited any further, the designer would have to ensure that the mobile device at least remains in the peripherally attached position. A good option to do so, would be to introduce a metaphor for the interaction technique, where the *palette* has to be connected to the paper by a *stitching* operation, e.g., by drawing a line around the edge of the mobile device and thereby connecting it to paper.

Next, the feedback should be reduced by only providing complimentary feedback with respect to mode and response. This would enable the user to have all necessary information while drawing, writing and changing tools, e.g., the pen width and color. Extending or complimentary feedback, e.g., the colorized facsimile and related documents, would only become relevant after drawing, e.g., when the user inspects the digital ink facsimile or completely switches to the mobile device. This could be detected by the mobile devices internal sensors and trigger an alternate representation of the application, as here the full bandwith has become available due to the mobile device residing in the main focus.

5.5 Summary and Conclusion

This chapter presented an empirically substantiated theory of interaction in hybrid mPPI ensembles. Following the research approach introduced throughout section 5.1, the theory was developed based on the results of an exploratory study on interaction in hybrid mPPI ensembles. Section 5.2 described the approach, data and study setup in detail. The derived theory, as introduced in section 5.3, states that the four pillars of interaction *Tasks*, *Media Transitions*, *Focus* and *Feedback* are related via the connectors *Metaphors* and *Handling*. In addition, section 5.3.4 explained the role of feedback in particular and provided an in depth discussion of this important concept based on a second iteration of research.

The presented, novel theory of interaction in hybrid mPPI ensembles forms the first theoretical explanation of interaction in this setting. It can improve a designer's understanding of interaction when developing solutions combining the most common mobile tools: pen, paper and mobile devices. It explains how interaction should be designed and which important domain concepts the designer needs to take into account.

Toward this end, section 5.4 provided a set of 8 exemplary design guidelines derived from the theory laid out in section 5.3. These design guidelines can inform concrete solutions for re-occurring interaction design challenges in hybrid mPPI ensembles. Furthermore, these guidelines facilitate understanding and assessing existing interaction techniques with respect to hybrid mPPI ensembles and the support of PPI in the mobile domain. Ultimately, the presented design guidelines are able to guide interaction design for mobile PPI and, as such, complement the infrastructure presented in chapter 3 as well as the conceptual framework of interaction presented in chapter 4.

6 Summary and Conclusion

Synopsis: *This chapter concludes the thesis. It summarizes the important contributions and findings, as well as discusses their practical impact with respect to the domain of Mobile Pen-and-Paper Interaction toward answering the research questions raised in chapter 1. Additionally, it points out promising trails for future research both with respect to infrastructural support and interaction design.*

In todays fast moving society, supporting mobile usage practices has become a key challenge. Despite continuously evolving mobile technology, the highly mobile and intuitive combination of pen and paper continues to serve users in mobile situations. At the same time, users need interactive functionality, access to digital information and communication facilities as provided by mobile devices.

Here, the present thesis made significant steps forward toward integrating the best and by far most prevalent information processing technologies from the ”old” analog world – pen and paper – and the ”new” digital world – mobile devices, e.g., smartphones.

Although both technologies, by themselves, cater extremely well in supporting the modern, mobile user, the integration of Pen-and-Paper and computing technologies is still in its infancy today. In particular, it was not very well tuned to the important mobile use case. This motivated the focus of the present thesis: interaction and infrastructure support for hybrid mobile Pen-and-Paper Interaction (mPPI) ensembles, i.e., combinations of digital pens, paper and mobile devices.

6.1 Contributions

This thesis advances the field of mobile PPI by presenting contributions with respect to mPPI infrastructures, conceptual frameworks and interaction theories. Specifically, it presents a novel, distributed processing pipeline based infrastructure for mobile PPI, enabling support for hybrid mPPI ensembles at a large scale, while preserving important mobile usage characteristics of real pen and paper. In addition, it presents a flexible and extensible conceptual framework of PPI, specifically designed to serve as the foundation of toolkits. Finally, it presents the first comprehensive theoretical understanding of interaction design in the domain through an empirically substantiated theory, including a set of concrete design guidelines.

6.1.1 Infrastructure: System Support for Mobile PPI

Chapter 3 presented a novel infrastructure concept based on the distributed PPI processing pipeline. This infrastructure supports both PPI and mobile PPI, however, in particular suits mobile application as it preserves important mobile characteristics of pen and paper use: *user mobility*, i.e., interaction in mobile and nomadic settings, and *document mobility*, i.e., encountered documents and documents used in differing contexts (c.f., chapter 2, section 2.3.1).

Contribution. In a first step, the common conceptual underpinning employed in PPI infrastructures was described and analyzed. This generic PPI processing pipeline consists of several successive *processing stages* sequentially transforming digital ink from raw sensory information to higher level data constructs. First, the *driver stage* connects digital pen hardware and provides low level hardware access channeling digital ink into the system. Then, the *Region Stage* maps this digital ink to interactive regions, followed by the *Semantic Stage* interpreting the digital ink and adding semantic information. Finally, the *Application Stage* provides functionality commonly required at the application level.

Existing PPI infrastructures implicitly base on this generic PPI processing pipeline. As such they use a setup where all stages are all deployed in a monolithic scheme. This setup severely hinders supporting user mobility and document mobility, as demonstrated in chapter 3, section 3.4.1. In order to overcome this limitation, the *distributed PPI processing pipeline* architecture was introduced: it decouples the processing stages through *processing stage interfaces* consisting of data construct definitions and service specifications in combination with communication channels and a micro service architecture.

In contrast to the generic processing pipeline, the distributed processing pipeline supports mobile usage practices such as user mobility and document mobility by design. For instance, the distributed interaction processing pipeline allows for physical distribution of interaction processing in hybrid mPPI ensembles and thus allows adapting the infrastructure to changing environments, a common technique in mobile settings. It also enables the sharing of interaction resources between applications and supports discovery of encountered paper documents.

Based on the distributed PPI processing pipeline, an infrastructure for (mobile) PPI was presented and discussed in detail; including its reference implementation, *Letras*, demonstrating practical relevance and feasability of the approach. A theoretical analysis and *proof-of-concept* evaluation encompassing several prototypical applications and the in-depth case study of the *digital grocery list* underlined the validity of the approach.

Impact. The distributed interaction processing pipeline presents a novel conceptual underpinning for mPPI infrastructure. It enables infrastructures to support mobile PPI without compromising intrinsic mobile characteristics of pen and paper. In addition to this, the novel infrastructure concept introduced in this thesis allows leveraging hybrid mPPI ensembles to support applications in the mobile domain, based on the distributed PPI processing pipeline. This infrastructure offers a base platform for developing mobile PPI based applications and as such aids further bridging the gap between digital and physical tools in mobile settings. Most importantly, it does not force the developer of interactive systems spanning pen, paper and mobile devices into design decisions constraining the mobile experience at the conceptual level, or hinder mobile application of PPI altogether.

Thus, the novel mPPI infrastructure presented in this thesis allows application developers for the first time unhamperedly targeting mobile applications employing the modality of pen and paper. As such it provides a significant step forward with respect to an infrastructure able to leverage the concept of hybrid mPPI ensembles, from purely academic, to actual application scenarios.

6.1.2 Conceptual Framework of (mobile) PPI

Chapter 4 introduced a conceptual framework for PPI that allows formally expressing and describing Pen-and-Paper interaction, specifically designed to provide a suitable basis of PPI and mPPI toolkits. Thereby, the conceptual framework consists of a set of structural constituents describing interaction at the level of interaction techniques and offers semantics adopted from logic programming, as well as an empirically derived initial interaction vocabulary. As such, it provides the conceptual basis of toolkits for mPPI and hence forms the connecting element between infrastructural support and interaction design in hybrid mPPI ensembles.

Contribution. First, the basic structure and semantics of W^5 were introduced, presenting a novel conceptual framework for PPI. W^5 thereby extends concepts found in existing conceptual frameworks, e.g., the invocation of functionality and the semantic perspective on interaction. It describes interaction at the level of interaction techniques, using expressions specifying system input. Thereby, it derives its semantics from logic programming. Expressions are first order predicate logic expressions formed of interaction predicates describing elementary, observable aspects of user actions along an open set of five conceptual dimensions.

The five core dimensions lend W^5 its name and structure the PPI design space. Core dimensions comprise the *spatial dimension* W_1 (Where), the *temporal dimension* W_2 (When), the *content dimension* W_3 (What), the *contextual dimension* W_4 (Why) and the *originator dimension* W_5 (Who). Interaction predicates fully coincide with exactly one of these dimensions and describe either absolute, or relative aspects of user actions. Whenever these aspects are observed by the system, associated feedback can be triggered. Similarly, associated functionality is triggered whenever the system observes digital ink letting a complete expression formed of interaction predicates and logical connectors evaluate to *true*.

Second, an initial set of core interaction predicates was empirically derived from representatives of the three fundamental classes of PPI techniques spanning the currently explored design space (c.f., chapter 2, section 2.4.1). This yielded an open set of nine core interaction predicates. These core interaction predicates constitute the initial design vocabulary defined by W^5 and as such provide concrete abstractions for interaction designers. Thereby, core predicates enable designers to express interaction techniques from the existing classes and lay the foundation of toolkit development based on W^5.

Finally, the approach was evaluated using a combination of analytical evaluation and a *proof-of-concept* exemplifying a rule-based mPPI toolkit based on the W^5 conceptual framework of PPI. The evaluation also included prototypical implementation of interaction techniques in all three classes.

Impact. W^5 forms the first conceptual framework for PPI specifically designed to serve as toolkit basis. It combines insights from existing conceptual frameworks and puts them into a structural frame adding semantics adopted from logic programming. As such, it not only allows structuring the design space, it also adds specific conceptual abstractions which toolkits can adopt and offer to interaction designers in order to enable rapid development of novel interaction techniques and design space exploration. Thereby, its machine understandable representations of interaction techniques enable the infrastructure to support a broad variety of interaction techniques through a limited set of recognizers.

Although W^5 aims at expressing PPI in general, the proof-of-concept demonstrates its applicability to toolkit design for mobile PPI as well. Thus, W^5 as the basis of mPPI toolkits forms the connecting element between infrastructure and interaction in the context of mobile PPI.

6.1.3 Interaction: Theory of Mobile PPI

Chapter 5 presented an empirically substantiated theory of interaction in hybrid mPPI ensembles. Based on an extensive exploratory study, this theory describes the main aspects of mobile PPI and their respective interrelations. Theory development thereby employed a qualitative, iterative research approach. In particular, the role of feedback in hybrid mPPI ensembles was investigated extending initial insights with respect to the design of feedback for user actions. The resulting theory allows deriving a concrete set of design guidelines that can inform interaction design for hybrid mPPI ensembles.

Contribution. Initially, a domain analysis and review of existing theoretical insights was conducted with respect to interaction in hybrid mPPI ensembles. This yielded a basic setup for an exploratory, stimulus driven study. Thereby, media transitions were identified as the central entry point for investigation as they bear the potential to disrupt interaction, [Steimle, 2009b]. As a consequence, the design of the exploratory study revolved around the three main directions of media transitions and compared existing interaction techniques from the literature to a set of novel, integrated interaction techniques designed in a way to stimulate the media-transition within the course of interaction.

In order to generate empiric data, a set of three stimulus applications was developed. Users participating in the explorative study had to execute different tasks in several conditions, varying over media transitions, presence of feedback (acoustic and visual) and integrated interaction concepts. Actions were recorded on camera and subsequently discussed with the participants during a set of interviews and retrospective thinking-aloud sessions. Data elicited in the exploratory study was subjected to a qualitative analysis using an open, axial, selective coding approach by independent coders. Thereby, results were iteratively compiled into a theory of interaction in hybrid mPPI ensembles.

This theory states, that four central concepts, the *pillars* of interaction in hybrid mPPI ensembles, are connected via two interrelating factors, or *connectors*. Connectors thereby determine how the concepts exert influence on each other. Pillars of the theory thereby are *Media Transitions*, *Tasks*, *Feedback* and *Focus*, while *Metaphors* and *Handling* constitute the connectors.

Furthermore, the role of feedback was further investigated in a second macro-iteration. Here, a set of paper prototypes employing different feedback strategies was conceived based on re-evaluation of data gathered during the study with a focus on feedback. Subsequently, these prototypes were subjected to design-critique sessions with domain experts in order to elicit further coping strategies for challenging design questions.

Obtained theoretical insights then allowed for developing a set of basic interaction design guidelines for hybrid mPPI ensembles. For instance, derived interaction design guidelines inform designers to carefully *guide media transitions in tasks* through metaphors and handling, e.g., the sliding of a webpage ”into” a paper document in order to explain the direction and form of transition while the user has to hold the mobile device over the paper document (c.f., section 5.3.3). Other guidelines show how feedback should be adapted to user focus, i.e., the current area of visual attention, how to guide this focus during user tasks and how to appropriately design feedback for media transitions.

Impact. The presented theory forms the first theoretical foundation of interaction design in hybrid mPPI ensembles. It allows deriving concrete design guidelines for interaction spanning pen, paper and mobile device. Thereby, it informs interaction design in the domain of hybrid mPPI ensembles and answers the research questions raised in section 1.2 with respect to interaction design. Furthermore, it allows discussing existing interaction techniques in the light of newly gained theoretical understanding, ultimately enabling interaction designers to make informed choices based on the key characteristics of hybrid mPPI ensembles. As such, the mPPI theory breaks new ground for human computer interaction research addressing the combination of the most common mobile tools pen, paper and mobile device.

6.2 Directions for Further Research

Extending domain understanding through research is never complete. Despite the advances this thesis contributes to the fields of infrastructure and interaction research for hybrid mPPI ensembles, a plethora of further interesting research trails exists in the domain. Several of these research trails are particularly promising with respect to both, *infrastructure* and *interaction* design. In the following, a brief outline of particularly promising research trails extending the concepts contributed by this thesis is given and the reader is pointed to further resources where appropriate.

6.2.1 Improving the Infrastructure

Infrastructural research for hybrid mPPI ensembles should mainly focus on extending and improving the infrastructure in the light of special use cases. Additionally, the integration of infrastructure based on the distributed processing pipeline into large-scale publication and authoring systems should be investigated. Four promising research trails exist toward that end: research toward improving security aspects of the infrastructure, research toward comprehensive authoring for interactive regions, research toward realizing an interactive region naming system capable of scaling toward the demands of global PPI and research toward integrating user interaction with respect to infrastructural tasks, i.e., management tasks.

Infrastructural Security. An important consideration for the distributed PPI processing pipeline that has been neglected so far is to provide adequate security mechanisms for the distribution of digital ink processing, especially when targeting security critical use cases, e.g., cheque payment systems as in [Vines et al., 2012]. Consider for example the region processing stage. In the scope of the described processing pipeline, the distribution enables all applications to access all data on all interactive regions. This enables attackers, e.g., to obtain the facsimile of a signature. Currently, the only mechanism to overcome this situation in the presented infrastructure is to limit the visibility of services used. This approach exclusively relies on security mechanisms in the underlying middleware, e.g., *MundoCore*, [Aitenbichler et al., 2007], in *Letras*. However, finer grained access control to digital ink and published interactive regions is required in order to support real world applications demanding higher standards in security. This provides a promising road of future research.

Authoring of Interactive Regions. Advanced authoring concepts for PPI based applications are required [Signer et al., 2014]. How the static, document based approach for defining interactive regions in contemporary approaches can be mapped to the flexible concept of interactive region design remains a challenging issue. An approach for a de-centralized management and an authoring environment is necessary that enables more flexible forms of use, e.g., the use of a single document in multiple application contexts as form of *document mobility*. Such an authoring environment should encompass a way to define interactive regions not only on paper documents, as most existing approaches envision, but also on arbitrary surfaces, e.g., whiteboards, [Brandl et al., 2008], small paper artifacts, [Hurter et al., 2012], and table-top surfaces, [Doeweling et al., 2013]. As timely publishing of these interactive regions remains a challenging issue, this trail of research is closely related to the design of a flexible and extensible peer-to-peer based interactive region naming system.

Interactive Region Naming System. As described in section 3.1.3, chapter 3, the 2-stage approach to interactive region discovery addresses the problem of time critical interaction processing on the one hand and global lookup for interactive regions on the other hand. Here, a peer to peer based approach for the global naming system of interactive regions is suggested. Such a naming system essentially forms a multi-dimensional *distributed hash table* allowing to look up for the application responsible for a given interactive region. Thus, a two dimensional *content addressable network* (CAN), [Ratnasamy et al., 2001], offers a promising solution capable of scaling to global dimensions. However, the applicability of the concept in the domain of PPI and its performance in relation to the *P-Grid* based approach suggested by Weibel, [Weibel, 2009, p. 199], requires further research.

User Interaction for Infrastructural Tasks. Toward deploying hybrid mPPI ensembles in real-world applications, it becomes important to investigate how to integrate *management tasks* with respect to the infrastructure into the interaction design of applications. Thereby, associated challenges range from designing and integrating interaction with respect to connecting digital pen hardware to mobile devices (which, as described in section 3.2.2, requires manual setup), to including user interaction in infrastructural tasks, e.g., to improve online recognition schemes of handwriting or other handwritten input as in [Tsandilas, 2012]. Research here would have to determine how to integrate these schemes into applications without binding them exclusively to these applications. At the same time, the effect of propagating infrastructural peculiarities to the actual application design needs to be minimized, [Edwards et al., 2010]. Hence, this research trail is situated at the cross-over between infrastructure and interaction research with respect to hybrid mPPI ensembles.

6.2.2 Exploring Interaction

Research on interaction in hybrid mPPI ensembles should broaden the scope of available interaction techniques. Thereby, novel interaction techniques should be specifically designed for the hybrid nature of ensembles comprising pen, paper and digital devices and integrated interaction should be leveraged following the principles established in the theory (c.f., chapter 5). At the same time, novel technological developments should be taken into account. Thereby, the theory should be investigated in the light of additional mobile devices and device classes and, if necessary, extended. Future research is also needed, in order to examine the explanatory power of the theory in the light of mitigation strategies for media transitions, e.g., overlaid information and unified interaction devices.

Novel Interaction Techniques. Novel integrated interaction techniques for hybrid mPPI ensembles are required to develop interactive systems supporting users in the mobile domain. Research should in particular investigate the concept of interactively creating proxies on paper documents: data with respect to user satisfaction in the study presented in chapter 5 shows that users favor Pidget based interaction due to the reduced amount of media transitions when writing or drawing on paper. At the same time, users acknowledge that generalizing this concept is not possible, as it requires specially prepared paper, i.e., only applies to the dedicated class of paper documents which constitutes only a sub-portion of the paper documents used in mobile settings (c.f., results of the ethnographic study in section 1.1.1). A solution to this would be interactively created proxies on paper documents, where the user ”creates” the Pidgets needed.

Smart Spaces. Results of the explorative study showed that the relative physical placement of ensemble components plays an important role in hybrid mPPI ensembles (c.f., *placement schemes* and their effect on available *bandwith*, see section 5.3.2 and section 5.3.4 respectively). Besides explicitly designing interaction techniques toward manipulating placement schemes, the placement scheme itself could become means of system input. Additionally, the contextual dimension (W_4) and the originator dimension (W_5) remain severely underrepresented in terms of defined interaction predicates in W^5. User activity recognition and user recognition could therefore add further system input when designing novel interaction techniques. This leads to the concept of *smart spaces*, integrated environments with a broad variety of sensing capabilities, infrastructure services and recognition techniques, [Aitenbichler et al., 2007]. Investigating novel interaction concepts for mPPI based on ensemble component sensing, user activity recognition and user recognition in the context of such smart spaces provides another promising research trail.

Additional Mobile Devices. Future research needs to shed light on how the theory applies to other mobile devices types and multiple mobile devices. Here, additional concepts might become crucial, e.g., transitions between mobile devices. This is also induced by recent technological development, e.g., personal glasses enabling ubiquitous augmented reality as the Google Glasses project[1]. Additionally, the role of the form factor of the mobile device itself needs to be further investigated, e.g., it would be interesting to investigate differences between devices of the tablet and smartphone class and how or if they affect media transitions in hybrid mPPI ensembles.

Mitigation Strategies for Media Transitions. The current form of the theory of interaction for hybrid mPPI ensembles focuses on the portion of the design space where media transitions between pen, paper and the mobile device occur unmitigated. However, several researchers have introduced mitigation strategies mostly aimed at stationary settings, but recently also for hybrid mPPI ensembles. As such, the explanatory power and completeness of the theory needs to examined in the light of these strategies. Thereby, one strategy to mitigate media transitions is overlaid information on paper as in *PenLight*, [Song et al., 2009a], *MouseLight*, [Song et al., 2010], or *PenBook*, [Winkler et al., 2013]. This allows for feedback directly on a paper document. Another strategy are unified interaction devices where the same interaction device is used to enable interaction with the mobile device and paper, e.g., by leveraging the digital pen to interact with the mobile device, [Winkler et al., 2013], or by supporting touch input on paper, [Zhou et al., 2014].

[1] https://developers.google.com/glass/ (accessed: July 2015)

List of Figures

List of Tables

Bibliography

[Pat, 2001] (2001). Determination of a position code.

[Pat, 2010] (2010). A digital pen.

[Aberer et al., 2002] Aberer, K., Punceva, M., Hauswirth, M., and Schmidt, R. (2002). Improving data access in p2p systems. *Internet Computing, IEEE*, 6(1):58–67.

[Aitenbichler, 2006] Aitenbichler, E. (2006). *System Support for Ubiquitous Computing*. PhD thesis, Darmstadt University of Technology.

[Aitenbichler et al., 2007] Aitenbichler, E., Kangasharju, J., and Mühlhäuser, M. (2007). Mundocore: A light-weight infrastructure for pervasive computing. *Pervasive and Mobile Computing*, 3(4):332 – 361. Middleware for Pervasive Computing.

[Andrew et al., 2009] Andrew, A., Karlson, A., and Brush, A. (2009). Investigating the use of voice and ink for mobile micronote capture. In Gross, T., Gulliksen, J., Kotz, P., Oestreicher, L., Palanque, P., Prates, R., and Winckler, M., editors, *Human-Computer Interaction INTERACT 2009*, volume 5726 of *Lecture Notes in Computer Science*, pages 782–795. Springer Berlin / Heidelberg. 10.1007/978-3-642-03655-2_85.

[Anthony and Wobbrock, 2010] Anthony, L. and Wobbrock, J. O. (2010). A lightweight multistroke recognizer for user interface prototypes. In *Proceedings of Graphics Interface 2010*, GI '10, pages 245–252, Toronto, Ont., Canada, Canada. Canadian Information Processing Society.

[Ao et al., 2006] Ao, X., Li, J., Wang, X., and Dai, G. (2006). Structuralizing digital ink for efficient selection. In *Proceedings of the 11th international conference on Intelligent user interfaces*, IUI '06, pages 148–154, New York, NY, USA. ACM.

[Arzt-Mergemeier et al., 2007] Arzt-Mergemeier, J., Beiss, W., and Steffens, T. (2007). The digital voting pen at the hamburg elections 2008: electronic voting closest to conventional voting. In *Proceedings of the 1st international conference on E-voting and identity*, VOTE-ID'07, pages 88–98, Berlin, Heidelberg. Springer-Verlag.

[Baader and Snyder, 2001] Baader, F. and Snyder, W. (2001). *Handbook of Automated Reasoning*, volume I, chapter Unification Theory, pages 447 – 533. Elsevier Science.

[Baddeley, 1992] Baddeley, A. (1992). Working memory. *Science*, 255(5044):556–559.

[Balakrishnan et al., 2003] Balakrishnan, H., Kaashoek, M. F., Karger, D., Morris, R., and Stoica, I. (2003). Looking up data in p2p systems. *Communications of the ACM*, 46(2):43–48.

[Bassett et al., 2008] Bassett, R., Beagan, B., and Chapman, G. E. (2008). Grocery lists: connecting family, household and grocery store. *British Food Journal*, 110(2):206 – 217.

[Beaudouin-Lafon, 2000] Beaudouin-Lafon, M. (2000). Instrumental interaction: an interaction model for designing post-wimp user interfaces. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 446–453, New York, NY, USA. ACM.

[Beaudouin-Lafon, 2004] Beaudouin-Lafon, M. (2004). Designing interaction, not interfaces. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 15–22, New York, NY, USA. ACM.

[Bellotti et al., 2004] Bellotti, V., Dalal, B., Good, N., Flynn, P., Bobrow, D. G., and Ducheneaut, N. (2004). What a to-do: studies of task management towards the design of a personal task list manager. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 735–742, New York, NY, USA. ACM.

[Berglund et al., 2006] Berglund, A., Berglund, E., Larsson, A., and Bang, M. (2006). Paper remote: an augmented television guide and remote control. *Universal Access in the Information Society*, 4:300–327. 10.1007/s10209-004-0108-8.

[Blagojevic et al., 2010] Blagojevic, R., Chang, S. H.-H., and Plimmer, B. (2010). The power of automatic feature selection: Rubine on steroids. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, SBIM '10, pages 79–86, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

[Block et al., 2008] Block, F., Haller, M., Gellersen, H., Gutwin, C., and Billinghurst, M. (2008). Voodoosketch: extending interactive surfaces with adaptable interface palettes. In *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages 55–58, New York, NY, USA. ACM.

[Block and Morwitz, 1999] Block, L. G. and Morwitz, V. G. (1999). Shopping lists as an external memory aid for grocery shopping: Influences on list writing and list fulfillment. *Journal of Consumer Psychology*, 8(4):343 – 375.

[Borggräfe et al., 2008] Borggräfe, B., Dörner, C., Heß, J., and Pipek, V. (2008). Drawing services: towards a paper-based interface for end-user service orchestration. In *Proceedings of the 4th international workshop on End-user software engineering*, WEUSE '08, pages 66–70, New York, NY, USA. ACM.

[Brandl et al., 2007] Brandl, P., Haller, M., Hurnaus, M., Lugmayr, V., Oberngruber, J., Oster, C., Schafleitner, C., and Billinghurst, M. (2007). An adaptable rear-projection screen using digital pens and hand gestures. In *Proceedings of the 17th International Conference on Artificial Reality and Telexistence*, ICAT '07, pages 49–54, Washington, DC, USA. IEEE Computer Society.

[Brandl et al., 2008] Brandl, P., Haller, M., Oberngruber, J., and Schafleitner, C. (2008). Bridging the gap between real printouts and digital whiteboard. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '08, pages 31–38, New York, NY, USA. ACM.

[Brandl et al., 2010] Brandl, P., Richter, C., and Haller, M. (2010). Nicebook: supporting natural note taking. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, pages 599–608, New York, NY, USA. ACM.

[Buxton, 1986] Buxton, W. (1986). Chunking and phrasing and the design of human-computer dialogues. In *Proceedings of the IFIP World Computer Congress*, pages 475–480. North Holland Publishers.

[Card et al., 2000] Card, S. K., Newell, A., and Moran, T. P. (2000). *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA.

[Catmull and Rom, 1974] Catmull, E. and Rom, R. (1974). A class of local interpolating splines. In Barnhill, R. E. and Reisenfeld, R. F., editors, *Computer Aided Geometric Design*, pages 317–326, New York. Academic Press.

[Chapman et al., 2009] Chapman, C., Lahav, M., and Burgess, S. (2009). Digital pen: Four rounds of ethnographic and field research. In *Proceedings of the 42nd Hawaii International Conference on System Sciences, HICSS '09*, pages 1–10.

[Choe et al., 2010] Choe, B. W., Min, J.-K., and Cho, S.-B. (2010). Online gesture recognition for user interface on accelerometer built-in mobile phones. In *Proceedings of the 17th international conference on Neural information processing:*

models and applications - Volume Part II, ICONIP'10, pages 650–657, Berlin, Heidelberg. Springer-Verlag.

[Cohen and McGee, 2004] Cohen, P. R. and McGee, D. R. (2004). Tangible multimodal interfaces for safety-critical applications. *Commun. ACM*, 47(1):41–46.

[Cohen and Rosenbaum, 2011] Cohen, R. G. and Rosenbaum, D. A. (2011). Prospective and retrospective effects in human motor control: planning grasps for object rotation and translation. *Psychological Research*, 75(4):341–349.

[Corbin and Strauss, 1990] Corbin, J. M. and Strauss, A. (1990). Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology*, 13:3–21. 10.1007/BF00988593.

[Costa-Cunha and Mackay, 2003] Costa-Cunha, P. and Mackay, W. (2003). Augmented paper and anoto stylus. In *Proceedings of the 15th French-speaking conference on human-computer interaction on 15eme Conference Francophone sur l'Interaction Homme-Machine*, IHM 2003, pages 232–235, New York, NY, USA. ACM.

[Cowan et al., 2011] Cowan, L. G., Weibel, N., Pina, L. R., Hollan, J. D., and Griswold, W. G. (2011). Ubiquitous sketching for social media. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 395–404, New York, NY, USA. ACM.

[Dachselt et al., 2008] Dachselt, R., Frisch, M., and Decker, E. (2008). Enhancing uml sketch tools with digital pens and paper. In *SoftVis '08: Proceedings of the 4th ACM symposium on Software visualization*, pages 203–204, New York, NY, USA. ACM.

[Dai et al., 2009] Dai, L., Sears, A., and Goldman, R. (2009). Shifting the focus from accuracy to recallability: A study of informal note-taking on mobile information technologies. *ACM Trans. Comput.-Hum. Interact.*, 16(1):4:1–4:46.

[Doeweling et al., 2013] Doeweling, S., Tahiri, T., Sowinski, P., Schmidt, B., and Khalilbeigi, M. (2013). Support for collaborative situation analysis and planning in crisis management teams using interactive tabletops. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 273–282, New York, NY, USA. ACM.

[Edge and Blackwell, 2009] Edge, D. and Blackwell, A. F. (2009). Bimanual tangible interaction with mobile phones. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, TEI '09, pages 131–136, New York, NY, USA. ACM.

[Edwards et al., 2010] Edwards, W. K., Newman, M. W., and Poole, E. S. (2010). The infrastructure problem in hci. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 423–432, New York, NY, USA. ACM.

[Erol et al., 2008] Erol, B., Antúnez, E., and Hull, J. J. (2008). Hotpaper: multimedia interaction with paper using mobile phones. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 399–408, New York, NY, USA. ACM.

[Fitts, 1954] Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381 – 391. (Reprinted in Journal of Experimental Psychology: General, 121(3):262269, 1992).

[Fitzmaurice, 1993] Fitzmaurice, G. W. (1993). Situated information spaces and spatially aware palmtop computers. *Commun. ACM*, 36(7):39–49.

[Fouse et al., 2013] Fouse, A., Weibel, N., Johnson, C., and Hollan, J. D. (2013). Reifying social movement trajectories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2945–2948, New York, NY, USA. ACM.

[Frisch et al., 2010] Frisch, M., Heydekorn, J., and Dachselt, R. (2010). Diagram editing on interactive displays using multi-touch and pen gestures. In *Proceedings of the 6th International Conference on Diagrammatic Representation and Inference*, Diagrams'10, pages 182–196, Berlin, Heidelberg. Springer-Verlag.

[Fujisawa, 2007] Fujisawa, H. (2007). Information just-in-time: an approach to the paperless office. In *Proceedings of the 2nd Pacific Rim conference on Advances in image and video technology*, PSIVT'07, pages 2–2, Berlin, Heidelberg. Springer-Verlag.

[Garcia et al., 2013] Garcia, J., Bigo, L., Spicher, A., and Mackay, W. E. (2013). Papertonnetz: Supporting music composition with interactive paper. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 3051–3054, New York, NY, USA. ACM.

[Garcia et al., 2014a] Garcia, J., Tsandilas, T., Agon, C., and Mackay, W. (2014a). Papercomposer: Creating interactive paper interfaces for music composition. In *Proceedings of the 26th Conference on L'Interaction Homme-Machine*, IHM '14, pages 1–8, New York, NY, USA. ACM.

[Garcia et al., 2014b] Garcia, J., Tsandilas, T., Agon, C., and Mackay, W. E. (2014b). Structured observation with polyphony: A multifaceted tool for studying music composition. In *Proceedings of the 2014 Conference on Designing Interactive Systems*, DIS '14, pages 199–208, New York, NY, USA. ACM.

[Gauthier et al., 2014] Gauthier, Y., Marcy, J., Duprat, D., Paoleschi, A., Letondal, C., Lesbordes, R., Vinot, J.-L., and Hurter, C. (2014). Gesture-based interaction for strip'tic, a tangible space for air traffic controllers. In *Proceedings of the 26th Conference on L'Interaction Homme-Machine*, IHM '14, pages 18–27, New York, NY, USA. ACM.

[Gebhardt et al., 2014] Gebhardt, C., Rädle, R., and Reiterer, H. (2014). Integrative workplace: Studying the effect of digital desks on users' working practices. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pages 2155–2160, New York, NY, USA. ACM.

[Geyer et al., 2012] Geyer, F., Budzinski, J., and Reiterer, H. (2012). Ideavis: A hybrid workspace and interactive visualization for paper-based collaborative sketching sessions. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, NordiCHI '12, pages 331–340, New York, NY, USA. ACM.

[Geyer and Reiterer, 2010] Geyer, F. and Reiterer, H. (2010). A cross-device spatial workspace supporting artifact-mediated collaboration in interaction design. In *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, CHI EA '10, pages 3787–3792, New York, NY, USA. ACM.

[Gottschalk et al., 1996] Gottschalk, S., Lin, M. C., and Manocha, D. (1996). Obbtree: A hierarchical structure for rapid interference detection. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 171–180, New York, NY, USA. ACM.

[Grant, 1948] Grant, D. A. (1948). The latin square principle in the design and analysis of psychological experiments. *Psychological bulletin*, 45(5):427.

[Guimbretière, 2003] Guimbretière, F. (2003). Paper augmented digital documents. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 51–60, New York, NY, USA. ACM.

[Ha et al., 2014] Ha, S., Park, J., and Lee, J. (2014). Increasing interactivity of paper prototyping with smart pen. In *Proceedings of HCI Korea*, HCIK '15, pages 76–82, South Korea. Hanbit Media, Inc.

[Haller et al., 2010] Haller, M., Leitner, J., Seifried, T., Wallace, J. R., Scott, S. D., Richter, C., Brandl, P., Gokcezade, A., and Hunter, S. (2010). The nice discussion room: Integrating paper and digital media to support co-located group meetings. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 609–618, New York, NY, USA. ACM.

[Harboe and Huang, 2015] Harboe, G. and Huang, E. M. (2015). Real-world affinity diagramming practices: Bridging the paper-digital gap. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 95–104, New York, NY, USA. ACM.

[Hartl et al., 2002] Hartl, A., Aitenbichler, E., Austaller, G., Heinemann, A., Limberger, T., Braun, E., and Mühlhäuser, M. (2002). Engineering Multimedia-Aware Personalized Ubiquitous Services. In *IEEE Fourth International Symposium on Multimedia Software Engineering (MSE'02)*, pages 344–351.

[Heinrichs, 2009] Heinrichs, F. (2009). Unstructured interaction: Integrating informal handwritten knowledge into business processes. In *LWA '09: Proceedings of Lernen Wissen Adaption, ABIS 2009*, pages 17 – 22.

[Heinrichs et al., 2011a] Heinrichs, F., Schreiber, D., Huber, J., and Mühlhäuser, M. (2011a). W5: A meta-model for pen-and-paper interaction. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, EICS '11, pages 47–52, New York, NY, USA. ACM.

[Heinrichs et al., 2012] Heinrichs, F., Schreiber, D., Huber, J., and Mühlhäuser, M. (2012). Toward a theory of interaction in mobile paper-digital ensembles. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pages 1897–1900, New York, NY, USA. ACM.

[Heinrichs et al., 2010a] Heinrichs, F., Schreiber, D., and Mühlhäuser, M. (2010a). Pen-and-paper interaction for everyday tasks. In *MuC'10: Proceedings of the 1st Workshop on Be-Greifbare Interaktion in Gemischten Wirklichkeiten, Mensch&Computer 2010*.

[Heinrichs et al., 2011b] Heinrichs, F., Schreiber, D., and Schöning, J. (2011b). The hybrid shopping list: Bridging the gap between physical and digital shopping lists. In *Proceedings of the 13th international conference on Human computer interaction with mobile devices and services*, MobileHCI '11, pages 251–254, New York, NY, USA. ACM.

[Heinrichs et al., 2010b] Heinrichs, F., Steimle, J., Schreiber, D., and Mühlhäuser, M. (2010b). Letras: An architecture and framework for ubiquitous pen-and-paper interaction. In *EICS '10: Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems*, pages 193–198, New York, NY, USA. ACM.

[Herold and Stahovich, 2012] Herold, J. and Stahovich, T. F. (2012). The 1c recognizer: a fast, accurate, and easy-to-implement handwritten gesture recognition technique. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, SBIM '12, pages 39–46, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

[Hess et al., 2008] Hess, J., Küstermann, G., and Pipek, V. (2008). Premote: a user customizable remote control. In *CHI '08 extended abstracts on Human factors in computing systems*, CHI '08, pages 3279–3284, New York, NY, USA. ACM.

[Hess et al., 2011] Hess, J., Wan, L., Pipek, V., and Kuestermann, G. (2011). Using paper and pen to control home-it: Lessons learned by hands-on experience. In *Proceddings of the 9th International Interactive Conference on Interactive Television*, EuroITV '11, pages 203–212, New York, NY, USA. ACM.

[Hinckley, 2007] Hinckley, K. (2007). *The Human Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, chapter Input Technologies and Techniques, pages 161 – 176. Lawrence Erlbaum Associates, Mahwah, NJ, USA, 2 edition.

[Hinckley et al., 2005] Hinckley, K., Baudisch, P., Ramos, G., and Guimbretiere, F. (2005). Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '05, pages 451–460, New York, NY, USA. ACM.

[Hinckley et al., 2006] Hinckley, K., Guimbretiere, F., Agrawala, M., Apitz, G., and Chen, N. (2006). Phrasing techniques for multi-stroke selection gestures. In *Proceedings of Graphics Interface 2006*, GI '06, pages 147–154, Toronto, Ont., Canada, Canada. Canadian Information Processing Society.

[Hinckley et al., 1997] Hinckley, K., Pausch, R., and Proffitt, D. (1997). Attention and visual feedback: the bimanual frame of reference. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, I3D '97, pages 121–ff., New York, NY, USA. ACM.

[Hinckley et al., 2004] Hinckley, K., Ramos, G., Guimbretiere, F., Baudisch, P., and Smith, M. (2004). Stitching: pen gestures that span multiple displays. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 23–31, New York, NY, USA. ACM.

[Hinckley et al., 2010] Hinckley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H., and Buxton, B. (2010). Pen + touch = new tools. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pages 27–36, New York, NY, USA. ACM.

[Hofer and Kunz, 2010] Hofer, R. L. and Kunz, A. (2010). Digisketch: taming anoto technology on lcds. In *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems*, EICS '10, pages 103–108, New York, NY, USA. ACM.

[Hollan et al., 2000] Hollan, J., Hutchins, E., and Kirsh, D. (2000). Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Trans. Comput.-Hum. Interact.*, 7(2):174–196.

[Holman et al., 2005] Holman, D., Vertegaal, R., Altosaar, M., Troje, N., and Johns, D. (2005). Paper windows: interaction techniques for digital paper. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 591–599, New York, NY, USA. ACM.

[Holzmann and Vogler, 2012] Holzmann, C. and Vogler, M. (2012). Building interactive prototypes of mobile user interfaces with a digital pen. In *Proceedings of the 10th Asia Pacific Conference on Computer Human Interaction*, APCHI '12, pages 159–168, New York, NY, USA. ACM.

[Hong et al., 2012] Hong, M., Piper, A. M., Weibel, N., Olberding, S., and Hollan, J. (2012). Microanalysis of active reading behavior to inform design of interactive desktop workspaces. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, pages 215–224, New York, NY, USA. ACM.

[Hudson et al., 2010] Hudson, S. E., Harrison, C., Harrison, B. L., and LaMarca, A. (2010). Whack gestures: inexact and inattentive interaction with mobile devices. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, TEI '10, pages 109–112, New York, NY, USA. ACM.

[Hurlbutt and Klemmer, 2006] Hurlbutt, T. and Klemmer, S. R. (2006). Bridging the gap: fluidly connecting paper notecards with digital representations for story/task-based planning. In *CHI '06 extended abstracts on Human factors in computing systems*, CHI '06, pages 887–892, New York, NY, USA. ACM.

[Hurter et al., 2012] Hurter, C., Lesbordes, R., Letondal, C., Vinot, J.-L., and Conversy, S. (2012). Strip'tic: Exploring augmented paper strips for air traffic controllers. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 225–232, New York, NY, USA. ACM.

[Ikeda et al., 2006a] Ikeda, H., Furakawa, N., and Konishi, K. (2006a). iJITinLab: Information Handling Environment Enabling Integration of Paper and Electronic Documents. In *Proceedings of CoPADD 2006, 1st International Workshop on Collaborating over Paper and Digital Documents*, pages 25–28, Banff, Canada.

[Ikeda et al., 2006b] Ikeda, H., Konishi, K., and Furakawa, N. (2006b). iJITinOffice: Desktop Environment Enabling Integration of Paper and Electronic Documents. In *Extended Abstracts (Posters) of UIST '06, the 19th annual ACM Symposium on User Interface Software and Technology*, Montreux, Switzerland.

[Ishii and Ullmer, 1997] Ishii, H. and Ullmer, B. (1997). Tangible bits: towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 234–241, New York, NY, USA. ACM.

[Ispas et al., 2010a] Ispas, A., Li, N., Norrie, M. C., and Signer, B. (2010a). Paper-digital meeting support and review. In *CollaborateCom '10: Proceedings of the 6th International Conference on Collaborative Computing*, Chicago, USA.

[Ispas et al., 2012] Ispas, A., Schöni, S., and Norrie, M. C. (2012). Areno: Ambient reminder notes. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*, OzCHI '12, pages 253–262, New York, NY, USA. ACM.

[Ispas et al., 2010b] Ispas, A., Signer, B., and Norrie, M. C. (2010b). A study of incidental notetaking to inform digital pen and paper solutions. In *HCI '10: Proceedings of the 24th BCS Conference on Human Computer Interaction*, Dundee, United Kingdom.

[Ispas et al., 2011] Ispas, A., Signer, B., and Norrie, M. C. (2011). An extensible digital ink segmentation and classification framework for natural notetaking. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, EICS '11, pages 231–240, New York, NY, USA. ACM.

[Iwata et al., 2009] Iwata, K., Kise, K., Nakai, T., Iwamura, M., Uchida, S., and Omachi, S. (2009). Capturing digital ink as retrieving fragments of document images. In *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, ICDAR '09, pages 1236–1240, Washington, DC, USA. IEEE Computer Society.

[Jacob et al., 2008] Jacob, R. J., Girouard, A., Hirshfield, L. M., Horn, M. S., Shaer, O., Solovey, E. T., and Zigelbaum, J. (2008). Reality-based interaction: a framework for post-wimp interfaces. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 201–210, New York, NY, USA. ACM.

[Jain et al., 2008] Jain, J., Ghosh, R., and Dekhil, M. (2008). Multimodal capture of consumer intent in retail. In *CHI '08 extended abstracts on Human factors in computing systems*, CHI EA '08, pages 3207–3212, New York, NY, USA. ACM.

[Kim et al., 2014] Kim, J., Seo, J., and Han, T.-D. (2014). Ar lamp: Interactions on projection-based augmented reality for interactive learning. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, IUI '14, pages 353–358, New York, NY, USA. ACM.

[Kim et al., 2009] Kim, K., Turner, S. A., and Pérez-Quiñones, M. A. (2009). Requirements for electronic note taking systems: A field study of note taking in university classrooms. *Education and Information Technologies*, 14(3):255–283.

[Konishi et al., 2007] Konishi, K., Furukawa, N., and Ikeda, H. (2007). Data model and architecture of a paper-digital document management system. In *Proceedings of the 2007 ACM symposium on Document engineering*, DocEng '07, pages 29–31, New York, NY, USA. ACM.

[Kusunoki et al., 2013] Kusunoki, D., Sarcevic, A., Weibel, N., and Burd, R. S. (2013). Opportunities for blended interaction in emergency rooms: Coupling displays with digital pen input. In *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare*, PervasiveHealth '13, pages 288–289, ICST, Brussels, Belgium, Belgium. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[Lawson et al., 2009] Lawson, J.-Y. L., Al-Akkad, A.-A., Vanderdonckt, J., and Macq, B. (2009). An open source workbench for prototyping multimodal interactions based on off-the-shelf heterogeneous components. In *Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '09, pages 245–254, New York, NY, USA. ACM.

[Lazar et al., 2010] Lazar, J., Feng, J. H., and Hochheiser, H. (2010). *Research methods in human-computer interaction*. John Wiley & Sons.

[Lee and Stahovich, 2014] Lee, W. and Stahovich, T. F. (2014). Papercad: A system for interrogating cad drawings using small mobile computing devices combined with interactive paper. *Adv. in Hum.-Comp. Int.*, 2014:14:14–14:14.

[Leitner et al., 2009] Leitner, J., Powell, J., Brandl, P., Seifried, T., Haller, M., Dorray, B., and To, P. (2009). Flux: a tilting multi-touch and pen based surface. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, CHI '09, pages 3211–3216, New York, NY, USA. ACM.

[Leitner et al., 2010] Leitner, J., Rendl, C., Perteneder, F., Gokcezade, A., Seifried, T., Haller, M., Zeleznik, R., and Bragdon, A. (2010). Nice formula editor. In *ACM SIGGRAPH 2010 Talks*, SIGGRAPH '10, pages 55:1–55:1, New York, NY, USA. ACM.

[Letondal et al., 2013] Letondal, C., Hurter, C., Lesbordes, R., Vinot, J.-L., and Conversy, S. (2013). Flights in my hands: Coherence concerns in designing strip'tic, a tangible space for air traffic controllers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2175–2184, New York, NY, USA. ACM.

[Liao and Guimbretièere, 2012] Liao, C. and Guimbretièere, F. (2012). Evaluating and understanding the usability of a pen-based command system for interactive paper. *ACM Trans. Comput.-Hum. Interact.*, 19(1):3:1–3:24.

[Liao et al., 2005] Liao, C., Guimbretière, F., and Hinckley, K. (2005). Papiercraft: a command system for interactive paper. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 241–244, New York, NY, USA. ACM.

[Liao et al., 2008] Liao, C., Guimbretière, F., Hinckley, K., and Hollan, J. (2008). Papiercraft: A gesture-based command system for interactive paper. *ACM Trans. Comput.-Hum. Interact.*, 14(4):1–27.

[Liao et al., 2006] Liao, C., Guimbretière, F., and Loeckenhoff, C. E. (2006). Pen-top feedback for paper-based interfaces. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 201–210, New York, NY, USA. ACM.

[Liao et al., 2009] Liao, C., Guimbretire, F., Anderson, R., Linnell, N., Prince, C., and Razmov, V. (2009). Papercp: Exploring the integration of physical and digital affordances for active learning. In *Human-Computer Interaction INTERACT 2007*, volume 4663/2009 of *Lecture Notes in Computer Science*, pages 15–28. Springer Berlin / Heidelberg.

[Liao and Liu, 2011] Liao, C. and Liu, Q. (2011). Mixpad: augmenting interactive paper with mice &#38; keyboards for fine-grained cross-media interaction with documents. In *Proceedings of the 13th international conference on Ubiquitous computing*, UbiComp '11, pages 539–540, New York, NY, USA. ACM.

[Liao et al., 2010a] Liao, C., Liu, Q., Liew, B., and Wilcox, L. (2010a). Pacer: fine-grained interactive paper via camera-touch hybrid gestures on a cell phone. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, pages 2441–2450, New York, NY, USA. ACM.

[Liao et al., 2010b] Liao, C., Tang, H., Liu, Q., Chiu, P., and Chen, F. (2010b). Fact: fine-grained cross-media interaction with documents via a portable hybrid paper-laptop interface. In *Proceedings of the international conference on Multimedia*, MM '10, pages 361–370, New York, NY, USA. ACM.

[Lin et al., 2004] Lin, M., Lutters, W. G., and Kim, T. S. (2004). Understanding the micronote lifecycle: improving mobile support for informal note taking. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pages 687–694, New York, NY, USA. ACM.

[Lissermann et al., 2012] Lissermann, R., Olberding, S., Petry, B., Mühlhäuser, M., and Steimle, J. (2012). Papervideo: Interacting with videos on multiple paper-like displays. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 129–138, New York, NY, USA. ACM.

[Liu and Liao, 2012] Liu, Q. and Liao, C. (2012). Paperui. In *Proceedings of the 4th International Conference on Camera-Based Document Analysis and Recognition*, CBDAR'11, pages 83–100, Berlin, Heidelberg. Springer-Verlag.

[Liwicki et al., 2010] Liwicki, M., Rostanin, O., El-Neklawy, S. M., and Dengel, A. (2010). Touch & write: a multi-touch table with pen-input. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, pages 479–484, New York, NY, USA. ACM.

[Liwicki et al., 2011] Liwicki, M., Thieme, S., Kahl, G., and Dengel, A. (2011). An intelligent shopping list - combining digital paper with product ontologies. In *Proceedings of the 15th International Conference on Knowledge-based and Intelligent Information and Engineering Systems - Volume Part IV*, KES'11, pages 187–194, Berlin, Heidelberg. Springer-Verlag.

[Liwicki et al., 2009] Liwicki, M., Weber, M., and Dengel, A. (2009). Automatic recognition and interpretation of pen- and paper-based document annotations. In *Proceedings of the 32nd annual German conference on Advances in artificial intelligence*, KI'09, pages 581–588, Berlin, Heidelberg. Springer-Verlag.

[Lloyd, 1984] Lloyd, J. W. (1984). *Foundations of Logic Programming*. Springer-Verlag New York, Inc., New York, NY, USA.

[Mackay et al., 2002] Mackay, W. E., Pothier, G., Letondal, C., Bøegh, K., and Sørensen, H. E. (2002). The missing link: augmenting biology laboratory notebooks. In *UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 41–50, New York, NY, USA. ACM.

[Malacria et al., 2011] Malacria, S., Pietrzak, T., Tabard, A., and Lecolinet, é. (2011). U-note: Capture the class and access it everywhere. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part I*, INTERACT'11, pages 643–660, Berlin, Heidelberg. Springer-Verlag.

[Malik et al., 2012] Malik, M. I., Ahmed, S., Dengel, A., and Liwicki, M. (2012). A signature verification framework for digital pen applications. In *Proceedings of the 2012 10th IAPR International Workshop on Document Analysis Systems*, DAS '12, pages 419–423, Washington, DC, USA. IEEE Computer Society.

[Matthews, 2006] Matthews, T. (2006). Designing and evaluating glanceable peripheral displays. In *Proceedings of the 6th conference on Designing Interactive systems*, DIS '06, pages 343–345, New York, NY, USA. ACM.

[Matthews et al., 2004] Matthews, T., Dey, A. K., Mankoff, J., Carter, S., and Rattenbury, T. (2004). A toolkit for managing user attention in peripheral displays. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, UIST '04, pages 247–256, New York, NY, USA. ACM.

[Matulic and Norrie, 2012] Matulic, F. and Norrie, M. C. (2012). Supporting active reading on pen and touch-operated tabletops. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 612–619, New York, NY, USA. ACM.

[Matulic and Norrie, 2013] Matulic, F. and Norrie, M. C. (2013). Pen and touch gestural environment for document editing on interactive tabletops. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 41–50, New York, NY, USA. ACM.

[McGee et al., 2002] McGee, D. R., Cohen, P. R., Wesson, R. M., and Horman, S. (2002). Comparing paper and tangible, multimodal tools. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 407–414, New York, NY, USA. ACM.

[Miura et al., 2007] Miura, M., Kunifuji, S., and Sakamoto, Y. (2007). Practical environment for realizing augmented classroom with wireless digital pens. In *Proceedings of the 11th international conference, KES 2007 and XVII Italian workshop on neural networks conference on Knowledge-based intelligent information and engineering systems: Part III*, KES'07/WIRN'07, pages 777–785, Berlin, Heidelberg. Springer-Verlag.

[Miura et al., 2010] Miura, M., Sugihara, T., and Kunifuji, S. (2010). Instant seat mapping for student note sharing process. In *Proceedings of the 2010 10th IEEE International Conference on Advanced Learning Technologies*, ICALT '10, pages 382–383, Washington, DC, USA. IEEE Computer Society.

[Newcomb et al., 2003] Newcomb, E., Pashley, T., and Stasko, J. (2003). Mobile computing in the retail arena. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 337–344, New York, NY, USA. ACM.

[Norman, 2002] Norman, D. A. (2002). *The Design of Everydayday Things*. Basic Books.

[Norman and Nielsen, 2010] Norman, D. A. and Nielsen, J. (2010). Gestural interfaces: a step backward in usability. *interactions*, 17(5):46–49.

[Norrie et al., 2005] Norrie, M. C., Palinginis, A., and Signer, B. (2005). Content publishing framework for interactive paper documents. In *Proceedings of the 2005 ACM symposium on Document engineering*, DocEng '05, pages 187–196, New York, NY, USA. ACM.

[Norrie and Signer, 2003] Norrie, M. C. and Signer, B. (2003). Issues of information semantics and granularity in cross-media publishing. In *Proceedings of the 15th international conference on Advanced information systems engineering*, CAiSE'03, pages 421–436, Berlin, Heidelberg. Springer-Verlag.

[Norrie and Signer, 2005] Norrie, M. C. and Signer, B. (2005). Information server for highly-connected cross-media publishing. *Inf. Syst.*, 30:526–542.

[Norrie et al., 2007] Norrie, M. C., Signer, B., Grossniklaus, M., Belotti, R., Decurtins, C., and Weibel, N. (2007). Context-aware platform for mobile data management. *Wirel. Netw.*, 13:855–870.

[Norrie et al., 2006a] Norrie, M. C., Signer, B., and Weibel, N. (2006a). General framework for the rapid development of interactive paper applications. In *In Proceedings of CoPADD 2006, 1st International Workshop on Collaborating over Paper and Digital Documents*, pages 9–12.

[Norrie et al., 2006b] Norrie, M. C., Signer, B., and Weibel, N. (2006b). Print-n-link: weaving the paper web. In *Proceedings of the 2006 ACM symposium on Document engineering*, DocEng '06, pages 34–43, New York, NY, USA. ACM.

[Norrie et al., 2008] Norrie, M. C., Signer, B., and Weibel, N. (2008). Interactive paper as a reading medium in digital libraries. In *Proceedings of the 12th European conference on Research and Advanced Technology for Digital Libraries*, ECDL '08, pages 232–243, Berlin, Heidelberg. Springer-Verlag.

[Nurmi et al., 2009] Nurmi, P., Forsblom, A., Floréen, P., Peltonen, P., and Saarikko, P. (2009). Predictive text input in a mobile shopping assistant: methods and interface design. In *IUI '09: Proceedings of the 13th international conference on Intelligent user interfaces*, pages 435–438, New York, NY, USA. ACM.

[Nurmi et al., 2008] Nurmi, P., Lagerspetz, E., Buntine, W., Floréen, P., and Kukkonen, J. (2008). Product retrieval for grocery stores. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 781–782, New York, NY, USA. ACM.

[Nyu and Miura, 2011] Nyu, T. and Miura, M. (2011). Applying gesture command input method for pen-based group kj system. In *Proceedings of the 15th International Conference on Knowledge-based and Intelligent Information and Engineering Systems - Volume Part III*, KES'11, pages 96–105, Berlin, Heidelberg. Springer-Verlag.

[O'Hara and Sellen, 1997] O'Hara, K. and Sellen, A. (1997). A comparison of reading paper and on-line documents. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '97, pages 335–342, New York, NY, USA. ACM.

[Olsen, 2007] Olsen, Jr., D. R. (2007). Evaluating user interface systems research. In *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 251–258, New York, NY, USA. ACM.

[Oviatt et al., 2007] Oviatt, S., Arthur, A., Brock, Y., and Cohen, J. (2007). Expressive pen-based interfaces for math education. In *Proceedings of the 8th iternational conference on Computer supported collaborative learning*, CSCL'07, pages 573–582. International Society of the Learning Sciences.

[Oviatt et al., 2006] Oviatt, S., Arthur, A., and Cohen, J. (2006). Quiet interfaces that help students think. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, UIST '06, pages 191–200, New York, NY, USA. ACM.

[Oviatt et al., 2013] Oviatt, S., Cohen, A., and Weibel, N. (2013). Multimodal learning analytics: Description of math data corpus for icmi grand challenge workshop. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 563–568, New York, NY, USA. ACM.

[Perteneder et al., 2015] Perteneder, F., Grossauer, E.-M., Xu, Y., and Haller, M. (2015). Catch-up 360: Digital benefits for physical artifacts. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '15, pages 105–108, New York, NY, USA. ACM.

[Pietrzak et al., 2012] Pietrzak, T., Malacria, S., and Lecolinet, E. (2012). S-notebook: Augmenting mobile devices with interactive paper for data management. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 733–736, New York, NY, USA. ACM.

[Piper et al., 2012a] Piper, A. M., Weibel, N., and Hollan, J. (2012a). Tap & play: an end-user toolkit for authoring interactive pen and paper language activities. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pages 149–158, New York, NY, USA. ACM.

[Piper et al., 2013] Piper, A. M., Weibel, N., and Hollan, J. (2013). Audio-enhanced paper photos: Encouraging social interaction at age 105. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, pages 215–224, New York, NY, USA. ACM.

[Piper et al., 2010] Piper, A. M., Weibel, N., and Hollan, J. D. (2010). Introducing multimodal paper-digital interfaces for speech-language therapy. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '10, pages 203–210, New York, NY, USA. ACM.

[Piper et al., 2011] Piper, A. M., Weibel, N., and Hollan, J. D. (2011). Write-n-speak: Authoring multimodal digital-paper materials for speech-language therapy. *ACM Trans. Access. Comput.*, 4(1):2:1–2:20.

[Piper et al., 2012b] Piper, A. M., Weibel, N., and Hollan, J. D. (2012b). A pen-based toolkit for authoring collaborative language activities. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work Companion*, CSCW '12, pages 269–270, New York, NY, USA. ACM.

[Piper et al., 2014] Piper, A. M., Weibel, N., and Hollan, J. D. (2014). Designing audio-enhanced paper photos for older adult emotional wellbeing in communication therapy. *Int. J. Hum.-Comput. Stud.*, 72(8):629–639.

[Prange et al., 2015] Prange, A., Sandrala, I. P., Weber, M., and Sonntag, D. (2015). Robot companions and smartpens for improved social communication of dementia patients. In *Proceedings of the 20th International Conference on Intelligent User Interfaces Companion*, IUI Companion '15, pages 65–68, New York, NY, USA. ACM.

[Proctor and Wu, 2007] Proctor, R. W. and Wu, K.-P. L. (2007). *The Human Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, chapter Human Information Processing: An Overview for HCI, pages 54 – 56. Lawrence Erlbaum Associates, 2 edition.

[Puccinelli et al., 2009] Puccinelli, N. M., Goodstein, R. C., Grewal, D., Price, R., Raghubir, P., and Stewart, D. (2009). Customer experience management in retailing: Understanding the buying process. *Journal of Retailing*, 85(1):15 – 30. Enhancing the Retail Customer Experience.

[Ratnasamy et al., 2001] Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S. (2001). A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172, New York, NY, USA. ACM.

[Raychoudhury et al., 2013] Raychoudhury, V., Cao, J., Kumar, M., and Zhang, D. (2013). Middleware for pervasive computing: A survey. *Pervasive Mob. Comput.*, 9(2):177–200.

[Reilly et al., 2005] Reilly, D., Welsman-Dinelle, M., Bate, C., and Inkpen, K. (2005). Just point and click?: using handhelds to interact with paper maps. In *Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, MobileHCI '05, pages 239–242, New York, NY, USA. ACM.

[Rogers et al., 2011] Rogers, Y., Sharp, H., and Preece, J. (2011). *Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, Inc., New York, NY, USA, 3rd edition.

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

[Rowstron and Druschel, 2001] Rowstron, A. I. T. and Druschel, P. (2001). Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware '01: Proceedings of the IFIP/ACM International Conference* *on Distributed Systems Platforms Heidelberg*, volume 2218/2001 of *Lecture Notes in Computer Science*, pages 329–350, London, UK. Springer Berlin / Heidelberg.

[Rubine, 1991] Rubine, D. (1991). Specifying gestures by example. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '91, pages 329–337, New York, NY, USA. ACM.

[Schilit and Sengupta, 2004] Schilit, B. N. and Sengupta, U. (2004). Device ensembles. *Computer*, 37(12):56–64.

[Schreiner, 2008] Schreiner, K. (2008). Uniting the paper and digital worlds. *Computer Graphics and Applications, IEEE*, 28(6):6 –10.

[Scott et al., 2004] Scott, S. D., Carpendale, M. S. T., and Inkpen, K. M. (2004). Territoriality in collaborative tabletop workspaces. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, CSCW '04, pages 294–303, New York, NY, USA. ACM.

[Sellen and Harper, 2003] Sellen, A. J. and Harper, R. H. R. (2003). *The Myth of the Paperless Office*. MIT Press, Cambridge, MA, USA.

[Sezgin and Davis, 2005] Sezgin, T. M. and Davis, R. (2005). Hmm-based efficient sketch recognition. In *Proceedings of the 10th international conference on Intelligent user interfaces*, IUI '05, pages 281–283, New York, NY, USA. ACM.

[Shaer et al., 2004] Shaer, O., Leland, N., Calvillo-Gamez, E. H., and Jacob, R. J. K. (2004). The tac paradigm: specifying tangible user interfaces. *Personal Ubiquitous Comput.*, 8:359–369.

[Shekar et al., 2003] Shekar, S., Nair, P., and Helal, A. S. (2003). igrocer: a ubiquitous and pervasive smart grocery shopping system. In *SAC '03: Proceedings of the 2003 ACM symposium on Applied computing*, pages 645–652, New York, NY, USA. ACM.

[Shneiderman, 1983] Shneiderman, B. (1983). Direct manipulation: A step beyond programming languages. *Computer*, 16(8):57–69.

[Signer, 2005] Signer, B. (2005). *Fundamental concepts for interactive paper and cross-media information spaces*. Diss., technische wissenschaften, Eidgenössische Technische Hochschule ETH Zürich.

[Signer et al., 2009] Signer, B., De Spindler, A., and Norrie, M. C. (2009). A framework for link sharing in cooperative cross-media information spaces. In *Proceedings of the 6th international conference on Cooperative design, visualization, and engineering*, CDVE'09, pages 185–192, Berlin, Heidelberg. Springer-Verlag.

[Signer et al., 2007a] Signer, B., Grossniklaus, M., and Norrie, M. C. (2007a). Interactive paper as a mobile client for a multi-channel web information system. *World Wide Web*, 10:529–556.

[Signer et al., 2007b] Signer, B., Kurmann, U., and Norrie, M. (2007b). igesture: A general gesture recognition framework. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 02*, ICDAR '07, pages 954–958, Washington, DC, USA. IEEE Computer Society.

[Signer and Norrie, 2007a] Signer, B. and Norrie, M. C. (2007a). As we may link: a general metamodel for hypermedia systems. In *Proceedings of the 26th international conference on Conceptual modeling*, ER'07, pages 359–374, Berlin, Heidelberg. Springer-Verlag.

[Signer and Norrie, 2007b] Signer, B. and Norrie, M. C. (2007b). Paperpoint: a paper-based presentation and interactive paper prototyping tool. In *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 57–64, New York, NY, USA. ACM.

[Signer and Norrie, 2010] Signer, B. and Norrie, M. C. (2010). Interactive paper: Past, present and future. In *Proceedings of PaperComp 2010, 1st International Workshop on Paper Computing*, Copenhagen Denmark.

[Signer and Norrie, 2011] Signer, B. and Norrie, M. C. (2011). A model and architecture for open cross-media annotation and link services. *Information Systems*, 36(3):538–550.

[Signer et al., 2006] Signer, B., Norrie, M. C., Grossniklaus, M., Belotti, R., Decurtins, C., and Weibel, N. (2006). Paper-based mobile access to databases. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 763–765, New York, NY, USA. ACM.

[Signer et al., 2014] Signer, B., Norrie, M. C., Weibel, N., and Ispas, A. (2014). Advanced authoring of paper-digital systems. *Multimedia Tools Appl.*, 70(2):1309–1332.

[Song et al., 2011] Song, H., Benko, H., Guimbretiere, F., Izadi, S., Cao, X., and Hinckley, K. (2011). Grips and gestures on a multi-touch pen. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 1323–1332, New York, NY, USA. ACM.

[Song et al., 2009a] Song, H., Grossman, T., Fitzmaurice, G., Guimbretiere, F., Khan, A., Attar, R., and Kurtenbach, G. (2009a). Penlight: combining a mobile projector and a digital pen for dynamic visual overlay. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pages 143–152, New York, NY, USA. ACM.

[Song et al., 2010] Song, H., Guimbretiere, F., Grossman, T., and Fitzmaurice, G. (2010). Mouselight: bimanual interactions on digital paper using a pen and a spatially-aware mobile projector. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 2451–2460, New York, NY, USA. ACM.

[Song et al., 2006] Song, H., Guimbretière, F., Hu, C., and Lipson, H. (2006). Modelcraft: capturing freehand annotations and edits on physical 3d models. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, UIST '06, pages 13–22, New York, NY, USA. ACM.

[Song et al., 2009b] Song, H., Guimbretière, F., and Lipson, H. (2009b). The modelcraft framework: Capturing freehand annotations and edits to facilitate the 3d model design process using a digital pen. *ACM Trans. Comput.-Hum. Interact.*, 16:14:1–14:33.

[Sonntag et al., 2011a] Sonntag, D., Liwicki, M., and Weber, M. (2011a). Digital pen in mammography patient forms. In *Proceedings of the 13th international conference on multimodal interfaces*, ICMI '11, pages 303–306, New York, NY, USA. ACM.

[Sonntag et al., 2011b] Sonntag, D., Liwicki, M., and Weber, M. (2011b). Interactive paper for radiology findings. In *Proceedings of the 16th international conference on Intelligent user interfaces*, IUI '11, pages 459–460, New York, NY, USA. ACM.

[Steimle, 2009a] Steimle, J. (2009a). Designing pen-and-paper user interfaces for interaction with documents. In *TEI '09: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, pages 197–204, New York, NY, USA. ACM.

[Steimle, 2009b] Steimle, J. (2009b). *Integrating Printed and Digital Documents - Interaction Models and Techniques for Collaborative Knowledge Work*. PhD thesis, Technische Universität Darmstadt.

[Steimle, 2012] Steimle, J. (2012). Survey of pen-and-paper computing. In *Pen-and-Paper User Interfaces*, HumanComputer Interaction Series, pages 19–65. Springer Berlin Heidelberg.

[Steimle et al., 2008a] Steimle, J., Brdiczka, O., and Mühlhäuser, M. (2008a). Cross-media linking and tagging support for learning groups. In *Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on*, pages 714–719.

[Steimle et al., 2008b] Steimle, J., Brdiczka, O., and Mühlhäuser, M. (2008b). Digital paper bookmarks: collaborative structuring, indexing and tagging of paper documents. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 2895–2900, New York, NY, USA. ACM.

[Steimle et al., 2008c] Steimle, J., Brdiczka, O., and Mühlhäuser, M. (2008c). Paper-centric structuring in learning processes. In *Proceedings of the 2008 Eighth IEEE International Conference on Advanced Learning Technologies*, pages 685–687, Washington, DC, USA. IEEE Computer Society.

[Steimle et al., 2011] Steimle, J., Weibel, N., Olberding, S., Mühlhäuser, M., and Hollan, J. D. (2011). Plink: Paper-based links for cross-media information spaces. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, pages 1969–1974, New York, NY, USA. ACM.

[Stoica et al., 2001] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., and Balakrishnan, H. (2001). Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 149–160, New York, NY, USA. ACM.

[Sugihara et al., 2010a] Sugihara, T., Miura, M., and Kunifuji, S. (2010a). Practicing on stage: increasing transparency and interaction in class activity with digital pen system. In *Proceedings of the 14th international conference on Knowledge-based and intelligent information and engineering systems: Part IV*, KES'10, pages 457–464, Berlin, Heidelberg. Springer-Verlag.

[Sugihara et al., 2010b] Sugihara, T., Miura, T., Miura, M., and Kunifuji, S. (2010b). Examining the effects of the simultaneous display of students' responses using a digital pen system on class activity - a case study of an early elementary school in japan. In *Proceedings of the 2010 10th IEEE International Conference on Advanced Learning Technologies*, ICALT '10, pages 294–296, Washington, DC, USA. IEEE Computer Society.

[Tabard et al., 2008] Tabard, A., Mackay, W. E., and Eastmond, E. (2008). From individual to collaborative: the evolution of prism, a hybrid laboratory notebook. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, CSCW '08, pages 569–578, New York, NY, USA. ACM.

[Thomas and Garland, 2004] Thomas, A. and Garland, R. (2004). Grocery shopping: list and non-list usage. *Marketing Intelligence & Planning*, 22(6):623 – 635.

[Tsandilas, 2012] Tsandilas, T. (2012). Interpreting strokes on paper with a mobile assistant. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 299–308, New York, NY, USA. ACM.

[Tsandilas et al., 2009] Tsandilas, T., Letondal, C., and Mackay, W. E. (2009). Musink: composing music through augmented drawing. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 819–828, New York, NY, USA. ACM.

[Tsandilas and Mackay, 2010] Tsandilas, T. and Mackay, W. E. (2010). Knotty gestures: subtle traces to support interactive use of paper. In *AVI '10: Proceedings of the International Conference on Advanced Visual Interfaces*, pages 147–154, New York, NY, USA. ACM.

[Underwood et al., 2013a] Underwood, H., Sterling, S. R., and Bennett, J. K. (2013a). The design and implementation of the partopen maternal health monitoring system. In *Proceedings of the 3rd ACM Symposium on Computing for Development*, ACM DEV '13, pages 8:1–8:10, New York, NY, USA. ACM.

[Underwood et al., 2013b] Underwood, H., Sterling, S. R., and Bennett, J. K. (2013b). The partopen in practice: Evaluating the impact of digital pen technology on maternal health in kenya. In *Proceedings of the Sixth International Conference on Information and Communication Technologies and Development: Full Papers - Volume 1*, ICTD '13, pages 274–283, New York, NY, USA. ACM.

[Vines et al., 2012] Vines, J., Blythe, M., Dunphy, P., Vlachokyriakos, V., Teece, I., Monk, A., and Olivier, P. (2012). Cheque mates: Participatory design of digital payments with eighty somethings. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1189–1198, New York, NY, USA. ACM.

[Vinot et al., 2014] Vinot, J.-L., Letondal, C., Lesbordes, R., Chatty, S., Conversy, S., and Hurter, C. (2014). Tangible augmented reality for air traffic control. *interactions*, 21(4):54–57.

[Weibel, 2009] Weibel, N. (2009). *A Publishing Infrastructure for Interactive Paper Documents: Supporting Interactions across the Paper-Digital Divide*. Diss., technische wissenschaften, Eidgenössische Technische Hochschule ETH Zürich.

[Weibel et al., 2010a] Weibel, N., Cowan, L. G., Pina, L. R., Griswold, W. G., and Hollan, J. D. (2010a). Enabling social interactions through real-time sketch-based communication. In *Adjunct proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pages 405–406, New York, NY, USA. ACM.

[Weibel et al., 2012] Weibel, N., Fouse, A., Emmenegger, C., Friedman, W., Hutchins, E., and Hollan, J. (2012). Digital pen and paper practices in observational research. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pages 1331–1340, New York, NY, USA. ACM.

[Weibel et al., 2011a] Weibel, N., Fouse, A., Hutchins, E., and Hollan, J. D. (2011a). Supporting an integrated paper-digital workflow for observational research. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*, IUI '11, pages 257–266, New York, NY, USA. ACM.

[Weibel et al., 2008] Weibel, N., Ispas, A., Signer, B., and Norrie, M. C. (2008). Paperproof: a paper-digital proof-editing system. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 2349–2354, New York, NY, USA. ACM.

[Weibel et al., 2007] Weibel, N., Norrie, M. C., and Signer, B. (2007). A model for mapping between printed and digital document instances. In *Proceedings of the 2007 ACM symposium on Document engineering*, DocEng '07, pages 19–28, New York, NY, USA. ACM.

[Weibel et al., 2010b] Weibel, N., Piper, A. M., and Hollan, J. D. (2010b). Exploring pen and paper interaction with high-resolution wall displays. In *Adjunct proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pages 455–456, New York, NY, USA. ACM.

[Weibel et al., 2010c] Weibel, N., Piper, A. M., and Hollan, J. D. (2010c). Hiperpaper: introducing pen and paper interfaces for ultra-scale wall displays. In *Adjunct proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pages 407–408, New York, NY, USA. ACM.

[Weibel et al., 2011b] Weibel, N., Signer, B., Norrie, M. C., Hofstetter, H., Jetter, H.-C., and Reiterer, H. (2011b). Papersketch: a paper-digital collaborative remote sketching tool. In *Proceedings of the 16th international conference on Intelligent user interfaces*, IUI '11, pages 155–164, New York, NY, USA. ACM.

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

[Wellner, 1993] Wellner, P. (1993). Interacting with paper on the digitaldesk. *Commun. ACM*, 36(7):87–96.

[West et al., 2007] West, D., Quigley, A., and Kay, J. (2007). Memento: a digital-physical scrapbook for memory sharing. *Personal Ubiquitous Comput.*, 11:313–328.

[Winkler et al., 2013] Winkler, C., Seifert, J., Reinartz, C., Krahmer, P., and Rukzio, E. (2013). 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*, ITS '13, pages 283–286, New York, NY, USA. ACM.

[Witt et al., 2008] Witt, H., Lawo, M., and Drugge, M. (2008). Visual feedback and different frames of reference: The impact on gesture interaction techniques for wearable computing. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '08, pages 293–300, New York, NY, USA. ACM.

[Wobbrock et al., 2007] Wobbrock, J. O., Wilson, A. D., and Li, Y. (2007). Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 159–168, New York, NY, USA. ACM.

[Wu and Natchetoi, 2007] Wu, H. and Natchetoi, Y. (2007). Mobile shopping assistant: integration of mobile applications and web services. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 1259–1260, New York, NY, USA. ACM.

[Yang et al., 2011] Yang, X., Liao, C., Liu, Q., and Cheng, K.-T. (2011). Minimum correspondence sets for improving large-scale augmented paper. In *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, VRCAI '11, pages 59–68, New York, NY, USA. ACM.

[Yeh et al., 2007] Yeh, R., Klemmer, S., and Paepcke, A. (2007). Design and evaluation of an event architecture for paper uis: Developers create by copying and combining. Technical Report 2007-29, Stanford InfoLab.

[Yeh et al., 2006a] Yeh, R., Liao, C., Klemmer, S., Guimbretière, F., Lee, B., Kakaradov, B., Stamberger, J., and Paepcke, A. (2006a). Butterflynet: a mobile capture and access system for field biology research. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 571–580, New York, NY, USA. ACM.

[Yeh et al., 2006b] Yeh, R. B., Brandt, J., Klemmer, S. R., Boli, J., and Su, E. (2006b). Interactive gigapixel prints: Large paper interfaces for visual context, mobility, and collaboration. Technical report, Stanford University Computer Science Department.

[Yeh et al., 2008] Yeh, R. B., Paepcke, A., and Klemmer, S. R. (2008). Iterative design and evaluation of an event architecture for pen-and-paper interfaces. In *UIST '08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 111–120, New York, NY, USA. ACM.

[Zamarripa et al., 2007] Zamarripa, M. S., Gonzalez, V. M., and Favela, J. (2007). The augmented patient chart: seamless integration of physical and digital artifacts for hospital work. In *Proceedings of the 4th international conference on Universal access in human-computer interaction: applications and services*, UAHCI'07, pages 1006–1015, Berlin, Heidelberg. Springer-Verlag.

[Zhao et al., 2001] Zhao, B. Y., Kubiatowicz, J. D., and Joseph, A. D. (2001). Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical report, University of California at Berkeley, Berkeley, CA, USA.

[Zhou et al., 2014] Zhou, Y., Xu, T., David, B., and Chalon, R. (2014). Innovative wearable interfaces: An exploratory analysis of paper-based interfaces with camera-glasses device unit. *Personal Ubiquitous Comput.*, 18(4):835–849.

CV and Publications

Farid Hamza Felix Heinrichs, born August 29, 1979 in Frankfurt am Main, Germany

Academic CV

2009 – 2015	PhD Studies of Computer Science under supervision of Prof. Dr. Max Mühlhäuser at the Telecooperation lab (since 2012 external affiliation), Technische Universität Darmstadt, Germany
March 2009	German Diploma in Computer Science (Grade *"sehr gut"*), at Technische Universität Darmstadt, Germany
2002 – 2009	Studies of Computer Science at Technische Universität Darmstadt, Germany
2000 – 2002	Studies of Computer Science at Johann-Wolfgang-Goethe Universität, Frankfurt am Main, Germany
1999	Abitur, Lessing Gymnasium Frankfurt am Main

List of Publications

Heinrichs, F., Schreiber, D., Huber, J., and Mühlhäuser, M. (2012). Toward a theory of interaction in mobile paper-digital ensembles. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pages 1897–1900, New York, NY, USA. ACM

Heinrichs, F., Schreiber, D., Huber, J., and Mühlhäuser, M. (2011a). W5: A meta-model for pen-and-paper interaction. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, EICS '11, pages 47–52, New York, NY, USA. ACM

Heinrichs, F., Schreiber, D., and Schöning, J. (2011b). The hybrid shopping list: Bridging the gap between physical and digital shopping lists. In *Proceedings of the 13th international conference on Human computer interaction with mobile devices and services*, MobileHCI '11, pages 251–254, New York, NY, USA. ACM

Heinrichs, F., Steimle, J., Schreiber, D., and Mühlhäuser, M. (2010b). Letras: An architecture and framework for ubiquitous pen-and-paper interaction. In *EICS '10:*

Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems, pages 193–198, New York, NY, USA. ACM

Heinrichs, F., Schreiber, D., and Mühlhäuser, M. (2010a). Pen-and-paper interaction for everyday tasks. In *MuC'10: Proceedings of the 1st Workshop on Be-Greifbare Interaktion in Gemischten Wirklichkeiten, Mensch&Computer 2010*

Heinrichs, F. (2009). Unstructured interaction: Integrating informal handwritten knowledge into business processes. In *LWA '09: Proceedings of Lernen Wissen Adaption, ABIS 2009*, pages 17 – 22