# $W^5$: A Meta-Model for Pen-and-Paper Interaction

**Felix Heinrichs, Daniel Schreiber, Jochen Huber, Max Mühlhäuser**
FG Telecooperation, Department of Computer Science
Technische Universität Darmstadt
Hochschulstr. 10, 64289 Darmstadt, Germany
felix_h,schreiber,jhuber,max@tk.informatik.tu-darmstadt.de

**ABSTRACT**

Pen-and-Paper Interaction (PPI) is used in an increasing number of applications to bridge the digital-physical gap between paper and interactive computer systems. We present $W^5$, a meta-model for describing PPI, and demonstrate its expressiveness by applying it to several interaction techniques from the literature. In doing so, we derive a set of basic interaction primitives, which can be used to inform the design of development toolkits for PPI and guide interaction designers in a structured exploration of the design space. We present a proof-of-concept implementation for a PPI toolkit based on $W^5$ in order to demonstrate the practical relevance of our findings.

**ACM Classification Keywords**

H.5.2 Information Interfaces and Presentation: [Theory and methods]

**General Terms**

Human Factors, Design, Theory

**Author Keywords**

Model-based Interactive System Development, Development Tools / Toolkits / Programming Environments, Pen and Tactile Input, Handheld Devices and Mobile Computing, Digital Pen, Anoto

## INTRODUCTION

Pen and paper has recently gained some popularity as input modality for digital applications, ranging from note-taking [2] to computer based learning support [9]. These systems use a novel form of human-computer interaction, where the user interacts with a system using a digital pen and normal paper, here referred to as *Pen-and-Paper Interaction* (PPI). To support this form of interaction, applications provide a *Pen-and-Paper User Interface* (PPUI) in addition to, or instead of a graphical user interface (GUI).

As any other interactive application, these applications employ one or more interaction techniques, to trigger application functionality. Thereby, an *interaction technique* consists of *input* combined with appropriate *feedback* [4]. However, interaction techniques in PPUIs differ from those employed in traditional GUIs. For example, unlike a mouse, the pen 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 are needed for PPI based applications.

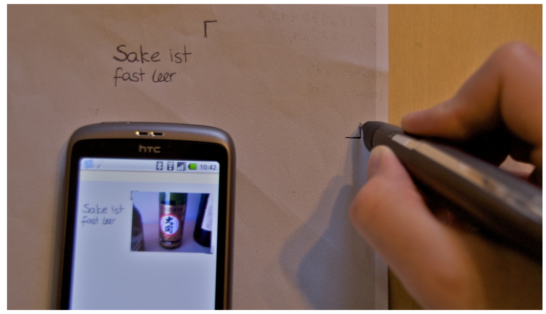


**Figure 1. The crop mark selection technique in our prototype**

Some examples of interaction techniques for PPUIs have been introduced in the literature. For example, a region on paper can be selected by drawing crop marks [5] (see Fig. 1). Yet, how can PPI interaction techniques be modeled? There is a lack of theoretical models for describing and categorizing such interaction techniques. Such a model should be able to guide researchers in finding new interaction techniques and help interaction designers in selecting appropriate techniques for PPI based applications. Ultimately, an appropriate model could inform the design of development toolkits for PPUIs.

Our contribution to solving this problem is threefold: First, we introduce the $W^5$ meta-model for modeling PPI. Second, based on this meta-model, we establish a set of primitives that suffice to describe and classify a significant portion of the design space for PPI interaction techniques. Third, we present a proof-of-concept implementation of a PPUI toolkit based on $W^5$ and the identified primitives to validate our theoretical concepts.

## RELATED WORK

The primacy of interaction design in the development of interactive computer systems has been emphasized by Beaudouin-Lafon [1]. Toolkits and models of interaction support the


*EICS 2011*, June 13-16, 2011, Pisa, Italy.

designer in the development of appropriate interaction techniques, by providing structuring and reusable components. Basically, PPI is a subset of tangible interaction, for which models exist, e.g. TAC [6]. However, expressing PPI with these models is cumbersome as primitives relevant for PPI, like gesture etc., must be constructed out of generic primitives for tangible interaction. The Resource-Selector-Link (RSL) model proposed by Signer and Norrie [7] has been used to describe PPI, e.g. in the iServer and iPaper framework [8]. Here paper artifacts are modeled as resources, to which links can be established using a certain selector. Although this general link model can be used to describe a broad range of different cross-media links, it does not explicitly model the interaction. To model interaction techniques employed in PPUIs, Steimle proposed a conceptual framework grounded on empirical research [9]. It 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. Conceptual activities are annotating, linking and tagging among others.

$W^5$ is grounded on this prior work. It forms around the general associative paradigm for PPI modeled in iServer and iPaper [8] to describe its interaction techniques. The semantic and syntactic level described by Steimle [9] are also included. $W^5$ generalizes the syntactic level, as it uses three dimensions (spatial, temporal, content) which are, in contrast to [9], independent from any application domain. The model of Steimle thus can be derived from our model by picking appropriate representatives from each dimension. However, the two models are not isomorphic: $W^5$ allows to express interaction techniques that cannot be expressed in the framework introduced by Steimle, e.g. temporal sequences. In contrast to [9], $W^5$ models PPI, without aspects of tangible interaction. Only input created by touching the paper with a pen is considered, as done naturally while writing or drawing; yet also "clicking" the paper can be described. Input by folding or rearranging paper is not addressed by $W^5$, although it could be expressed using the external primitive introduced below.

Software support for the interaction designer is also a problem, because in contrast to GUIs, there is no abundance of toolkits for PPUIs. The PaperToolkit [13] aims to provide a generic toolkit, however, it does not explicitly address interaction modeling. As a result, PaperToolkit does not lend itself for combining PPI input primitives into more complex input as it is typical for PPI. Other systems supporting PPI, such as Letras [3] or iServer and iPaper [8] provide the infrastructure needed to support PPI in applications, yet fail to provide the developer and interaction designer appropriate support when it comes to the development of novel interaction techniques.

## THE W⁵ META-MODEL

The purpose of $W^5$ is to describe actions executed by the user with a digital pen in the physical world, that can serve as input to digital applications, i.e. trigger digital functionality. This corresponds to the associative nature of the RSL model [7] introduced by Signer and Norrie, which has been successfully applied to PPI [8]. The application functionality itself and its connection to the input is not described by $W^5$.

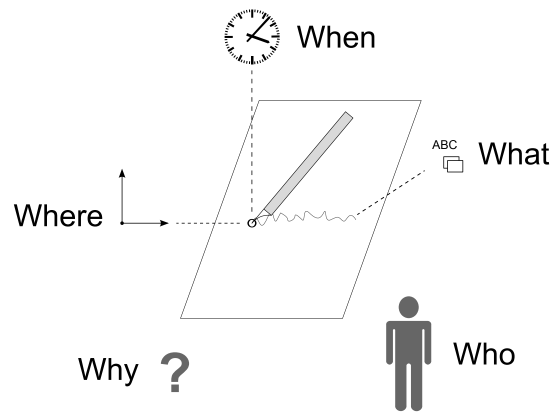


**Figure 2. Dimensions of the W⁵ Framework**

### W⁵ Dimensions

$W^5$ describes and classifies PPI based on elementary user actions, or *input primitives*, as well as their combinations into more elaborate input expressions. Interaction techniques are combinations of primitives with appropriate feedback. Each action detectable by a PPUI can be classified according to the following five dimensions, as illustrated in Figure 2

- **Where**: Spatial dimension ($W_1$)
- **When**: Temporal dimension ($W_2$)
- **What**: Content dimension ($W_3$)
- **Who**: Originator dimension ($W_4$)
- **Why**: Contextual task dimension ($W_5$)

All input primitives fully coincide with one of these dimensions. Spatial primitives (Where) use the location where the user touches the paper with a pen. Temporal primitives (When) use the time when the user touches the paper with a pen. Primitives from the content dimension (What) use the content created by the user with the pen on paper. Gestures or written commands belong to this dimension.

As an example, we consider a spatial input primitive. This input primitive describes touching a certain paper region with the pen. It can be directly used as an interaction technique, e.g. in [8, 2]. In these systems so called paper widgets, or *pidgets*, are used to trigger application functionality. Pidgets are typically marked on the paper with a small iconographic representation of the corresponding application functionality. Another example is a temporal primitive, where 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.

As shown in [9], the perspective on the information ecology cannot be neglected, so the user (Who) and the executed tasks (Why) are also reflected in $W^5$. These dimensions can also contribute to the interaction techniques supported by a PPUI. Depending on the originator, the system might, e.g., accept or reject a command. The task at hand also can influence available functionality, allowing for example modal interaction in PPUIs. We suggest to use these five dimensions as basis for the design space, i.e., define input primitives for PPI that fully coincide with one of these dimensions.

#### *Composition of Input Primitives*
The above input primitives are absolute, i.e., the pen data is compared to an absolute value. Specifying all input by means of absolute input primitives is problematic. To describe more complex input we need means to relate user actions with other user actions. We do so by introducing relative input primitives. Again we consider only input primitives that coincide with one of the five dimensions above.

An example for a relative spatial input primitive is "above" meaning that one input must be performed above another. Another example for a relative input primitive, from the temporal dimension, is "after", meaning that two inputs must be performed in sequence. For example, one can combine two spatial input primitives with the temporal sequence primitive (one shortly after the other) to have something similar to the double-click input known from GUI systems.

### $W^5$ Semantics and Notation
The semantics of $W^5$ are derived from logic programming. We regard the entire set of pen data $PD$ as facts. Each absolute input primitive is a unary predicate. For example, $At_R$ is *true* whenever $PD$ contains data points that lie inside the rectangle $R$. $At$ is an input primitive from the spatial dimension. The predicate can also be used with a variable, i.e., $At_R(X)$. In this case, all bindings for $X$ that let $At_R$ evaluate to *true* are returned.

Relative input primitives are n-ary predicates. For example, the $Abv$ primitive spatially relates two facts. $Abv(X,Y)$ returns all bindings to $X$ and $Y$ so that $X$ is "above" $Y$.

#### *Example*
To illustrate the concept, we apply it to an interaction technique from the literature. In the PapierCraft system, introduced by Liao et al. [5], the user can copy a portion of a document by marking its upper left and lower right corners with a so called crop gesture (see Figure 1 for an example of the crop gesture) followed by a pigtail gesture oriented to the right (east). Using the $W^5$ framework this is expressed as

$$G_{cs}(X) \wedge G_{ce}(Y) \wedge G_{pgt}(Z) \wedge \curvearrowright (X,Y,Z)$$

Where $G_{cs}$, $G_{ce}$ and $G_{pgt}$ are input primitives in the content dimension for the two crop marks and the pigtail symbol respectively. $\curvearrowright$ is a temporal relative input primitive indicating a temporal sequence. This can be abbreviated, if we do not reuse any of the variables, emphasizing the hierarchic composition of primitives: $\curvearrowright (G_{cs}, G_{ce}, G_{pgt})$

#### *The $EXT$ Predicate*
Because $W^5$ only describes PPI input, it is impossible to combine non-PPI input with PPI input in $W^5$. To remedy this limitation we introduce a special predicate called $EXT$, that evaluates to true when any relevant non-PPI input happens. This could, e.g., be pressing a button on a keyboard to switch between inking mode and command mode for the UI as in [5].

## BASIC INPUT PRIMITIVES
$W^5$ provides a way to model input in PPUIs. It supports the designer in constructing complex input out of given input primitives. The question arises, which primitives are relevant for implementing PPI? This question has high practical relevance, as such a set could be used to inform the design of PPUI toolkits.

### Derivation of Input Primitives
We derive a set of primitives by analysis of interaction techniques described in the literature. In particular we use the interaction techniques described in PapierCraft [5], PaperPoint [8], PaperProof [11] and CoScribe [9].

#### *PapierCraft*
Liao et al. describe a set of interaction techniques used in the PapierCraft system [5]. Their interaction techniques use a gesture based command system. Normally, the user uses the digital pen to generate digital ink and annotate a document. In order to invoke some functionality, the user presses a "gesture" button and then specifies a command. A command consists of a sequence of a *command scope* followed by an *intermediate delimiter* and a final *command type*. Commands can be constructed in sequences, e.g., a copy command followed by a paste command. We chose to analyze a copy and paste command using the crop mark selection gesture with an explicit written command, a hyperlink command with a margin bar selection gesture and a stitching gesture to combine two paper artifacts. Other interaction techniques in PapierCraft consist of (sub-)portions of these, or combine them sequentially.

Table 1 presents the employed composition of primitives in formal notation. Copy & paste and hyperlink are split in two lines for the sake of brevity. Formally, the two lines are connected using the temporal sequence primitive described below. Classified in their respective dimensions, used primitives are

- $W_1$ In the spatial dimension, only the absolute $At_X$ primitive has been used, where $X$ is a region in the absolute paper coordinate system (in the example above $R1$ and $R2$ for two distinct pages)
- $W_2$ In the temporal dimension, two relative primitives have been used: the simultaneous or parallel primitive $\|$ and the primitive for a temporal sequence $\curvearrowright$. Both have been used in k-ary definitions of sequences or parallelisms of arbitrary length.
- $W_3$ In the content dimension, two types of primitives have been used: gestures $G_X$ and words $W_X$, where $X$ cor-

| Technique | Formalization |
|---|---|
| copy & paste | $\Vert (EXT, \curvearrowright (G_{cs}, G_{ce}, G_{pgt}, G_E, W_{cp}))$ |
| | $\Vert (EXT, \curvearrowright (G_{cs}, G_{ce}, G_{pgt}, G_W))$ |
| hyperlink | $\Vert (EXT, \curvearrowright (G_{mb}, G_{pgt}, G_N))$ |
| | $\Vert (EXT, \curvearrowright (G_{pgt}, G_S))$ |
| stitch | $\Vert (EXT, \curvearrowright (At_{R1}, At_{R2}, At_{P1}) \wedge \curvearrowright (G_{st}, G_{pgt}, G_S))$ |

**Table 1. Interaction Techniques in PapierCraft**

responds to the specific gesture or word respectively. In theory, this is no limitation, as one can define arbitrarily many primitives. However, for practical considerations, the interpretation of $X$ as a parameter for such a primitive might prove useful.

### *PaperPoint and PaperProof*

The PaperPoint [8] and PaperProof [11] system were developed based on the iServer and iPaper infrastructure for PPI. PaperPoint only uses the pidget interaction technique. The input understood by the system can be modeled by using the absolute $At_R$ primitive in the spatial dimension. PaperProof employs gesture based interaction techniques, comparable to PapierCraft. We selected the two interaction techniques for annotation and move described in PaperProof to represent the used interaction techniques. Interestingly enough, the informal notation used in [11] to describe the employed interaction techniques, resembles the structure used in the $W^5$ meta-model and can be readily transcribed.

As can be seen in Table 2, the description of these interaction technique uses only the $DI$ primitive in addition to the primitives already defined. The $DI$ primitive belongs to the content dimension, and characterizes the need for specifying digital ink, i.e., handwriting or drawing, as part of an interaction technique.

### *CoScribe*

The CoScribe system described by Steimle [9] also introduced a set of PPI techniques. CoScribe differs from the other systems as it incorporates the contextual task domain. Again, we chose a representative subset of the interaction techniques and omitted techniques that either use the same primitives as in PapierCraft, e.g. stitching, or exist as subtechniques or combinations of the chosen techniques. The examined techniques include a technique for creating links between documents and a technique for tagging based on links and an optional written label.

| Technique | Formalization |
|---|---|
| annotation | $\curvearrowright (G_{cs1} \vee G_{cs2} \vee G_{<}, G_{ce1} \vee G_{ce2} \vee G_{>}, DI)$ |
| move | $\curvearrowright (G_{cs1} \vee G_{cs2} \vee G_{<}, G_{ce1} \vee G_{ce2} \vee G_{>}, G_N)$ |

**Table 2. Interaction Techniques in PaperProof**

| Technique | Formalization |
|---|---|
| hyperlink | $T_L \wedge \curvearrowright (For_t(At_{R1}), At_{R2})$ |
| tag | $T_T \wedge \curvearrowright (At_{R1}(W_T), For_t(At_{R1}), At_{R2})$ |

**Table 3. Interaction Techniques in CoScribe**

The composition of these techniques into input primitives is shown in Table 3. As can be seen, the only primitives used in addition to the primitives defined above are the $For_t$ and the $T_X$ primitives. The former describes an absolute input primitive in the temporal dimension, where $t$ means "executed for a duration of $t$". The latter describes the contextual task $X$, which is in this example either "linking" ($L$) or tagging ($T$). As can be seen, the only distinction between the two interaction techniques (if we omit the optional label writing action $At_{R1}(W_T)$) here is the contextual task.

### *Other Systems*

Other systems proposed in the literature offer interaction techniques that can be described using the above primitives. Knotty gestures [10] and their associated interaction techniques of tapping, holding, circling and marking are an example for using gestures $G_X$ in combination with absolute spatial $At_R$ and temporal $For_T$ primitives. An interesting observation here is, that the user "creates" the regions for the spatial $At_R$ primitive at run time, i.e. the knot which is used in other techniques, e.g. by tapping. NiceBook [2] presents another recent PPI based note taking application. The used interaction techniques include pidgets, a tagging system comparable to the one described by Steimle [9] and a dog-ear mark. All of these techniques can be described using the previously established primitives. ButterflyNet [12] presents 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. Of these, only automatic time based correlation requires a primitive not described so far: It associates two actions, i.e. taking a photo and writing something, iff these actions occur within a time interval. This is something less restrictive than the temporal $\Vert$ primitive defined above, so we denote this as a primitive $Int_T$, where $T$ marks the interval length.

## Overview of Basic Primitives

As we have shown, a relatively small set of *basic primitives*, suffices to model a broad variety of interaction techniques. The overview of basic primitives in the core dimensions is given in Table 4. It must be pointed out, however, that this set of basic primitives does not form a complete basis of the design space. So we conclude, that these primitives are necessary components of a PPUI toolkit – others might however be needed. Regarding the design of PPUI toolkits, we conclude that toolkits need to

(i) support the basic primitives described above

(ii) support be extensibility in terms of new primitives

| Primitive | | PapierCraft | PaperPoint | PaperProof | CoScribe | Knotty Gest. | NiceBook | ButterflyNet |
|---|---|---|---|---|---|---|---|---|
| **$W_1$** | $At_R$ | X | X | X | X | X | X | X |
| **$W_2$** | $Int_t$ | - | - | - | - | - | - | X |
| | $\parallel$ | X | - | - | - | - | - | - |
| | $\curvearrowright$ | X | - | X | X | X | X | X |
| | $For_t$ | - | - | - | X | X | X | - |
| **$W_3$** | $G_X$ | X | - | X | X | X | - | X |
| | $W_X$ | X | - | X | X | - | - | - |
| | $DI$ | X | X | X | X | - | X | X |
| **$W_4$** | $T_X$ | - | - | - | X | - | - | - |
| **$W_5$** | - | - | - | - | - | - | - | - |

**Table 4. Use of primitives in PPI based systems**

## PROOF-OF-CONCEPT IMPLEMENTATION

To validate our approach and the derived basic primitives, we have implemented a lightweight framework for PPUI development. This framework supports modeling interaction based on the $W^5$ meta-model and offers system support for basic primitives. Based on it, we have developed three applications employing interaction techniques described in the literature.

The system was implemented based on Letras [3]. Letras provides a generic PPI processing pipeline for Anoto[1] digital pens. Here, we used its pen drivers in combination with a Nokia SU-1B Digital pen. As most parts of Letras itself, our $W^5$ reference implementation is written in java. We designed it to support rapid development of PPUIs for mobile phones. Therefore, we deployed it on the Android[2] platform for smart phones. It is based on the Android 2.1 API version and has been tested on the Motorola Milestone and the HTC Desire smartphones.

To Letras, we added support for the basic primitives and the $W^5$ dimensions. Our implementation splits the actual recognition and the structure imposed by the formal description of interaction techniques into separate concerns. First a set of recognizers for the core dimensions allows to detect events along these dimensions. This directly provides support for the absolute primitives, e.g. $At_R$. All recognizers can be configured to recognize several primitives, e.g. several regions or gestures. When a primitive is recognized, the recognizer issues a corresponding event. Second, relative primitives and complete interaction techniques are modeled as rules. These rules receive the events emitted by the recognizers and fire, iff all required events have been received. This then triggers digital functionality in the application.

### *Implemented Interaction Techniques*

In order to validate the usefulness of our approach, we implemented 3 applications. The first application allows the user take notes using a PPUI and combine these notes with photographs taken with the phone camera. Here, we employed the hotspot association interaction technique described by Yeh [12] as part of ButterflyNet (see Figure 1). The second application allows the user to draw on paper, with a printed palette of various stroke widths and colors. Here we used pidgets to set the drawing mode of the pen, as done e.g. in NiceBook [2]. In the third application, the user can establish cross-media hyperlinks, as described by Steimle [9]. Here we use a similar interaction technique as in CoScribe, where the user draws a vertical line beside the part of the page that should be linked and then associates this page to a web page displayed on the smartphone by a similar gesture, this time using the finger instead of the pen (see Figure 3).

[1] `http://www.anoto.com`
[2] `http://www.android.com`

## DISCUSSION

A limitation of the $W^5$ meta-model is that feedback is currently not included. Nevertheless, the input primitives also represent the smallest units capable of providing feedback to the user. Theoretically, the model could be extended here to associate feedback with primitives, however, further research is needed in this direction. Implementations using the $W^5$ meta-model can assure that it is possible to add hooks in primitives to provide feedback on this level.

An open question is the completeness of the proposed dimensions. The selection has been grounded on study of interaction techniques in the literature, our practical knowledge of PPI design and prior work, as discussed above. However, no final statement on the completeness can be made. Nevertheless, we believe the current selection to represent the lion share of the design space for PPI and to be sufficient for all practical purposes.

The same must be said for the proposed set of primitives. As we have shown, it suffices to model many existing interaction techniques. However, we believe that this set is not complete. Nevertheless, the structuring provided by $W^5$ may help designers to systematically identify new primitives in the design space. For example, one can clearly see that relative spatial primitives, such as *above*, *below*, *close* etc., have been neglected so far. Additionally, the introduced primitives make only use of four dimensions of $W^5$. A remain-

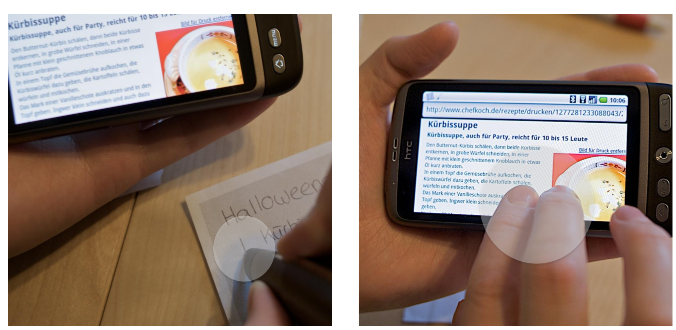


**Figure 3. A cross-media linking technique of CoScribe in our prototype: a) drawing a marker on paper and b) on the smartphone in sequence to establish a link**

ing question is, how these can be extended to systematically incorporate the contextual task (Why) and the originator (Who) dimensions. For example, a PPI based application could develop a user ID primitive in the originator dimension to allow users different actions based on their respective access rights.

Our implementation satisfies the need for supporting the basic primitives, with the exception of the word recognition primitive $W_X$ and the user task primitive $T_X$. This is because it would require a handwriting recognition sub-system or an explicit task model respectively, which is beyond the scope of a proof-of-concept implementation. The need for extendability in terms of primitives is, however, satisfied: The framework can be extended on the recognizer level, to support additional absolute primitives, and on the rule-level, to support relative primitives.

The implementation using a rule-based system, however, still has drawbacks in terms of developer support: the interaction designer has to maintain active knowledge how the recognizer affects the rules in the system and rules have to be specified programmatically, while having a low re-use factor. This comes from the fact that rules actually serve defining relative predicates along with application logic. A possible solution to this problem would be a domain specific language based on logic programming, that allows to specify interaction techniques more directly and offers a cleaner separation of concerns.

Our proof-of-concept implementation only shows, that the $W^5$ meta-model and the derived basic primitives can be used to design interaction techniques and explore the design space systematically. What it does not answer, is how huge the benefit in terms of system support for developers actually is. Although the ease of development in our setting proved to be considerable, a comparative study has to be conducted as part of future research.

## CONCLUSION

We have presented $W^5$, a meta-model to describe PPI in interactive computer systems, along with a set of basic primitives. Essentially, $W^5$ presents a way how the designer can look at and talk about PPI. It can 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 primitives or combinations thereof that have not been used so far. Additionally, $W^5$ can be used to inform the design of PPUI toolkits, as we have shown in our proof-of-concept implementation of such a toolkit.

## ACKNOWLEDGMENTS

Thanks to Niklas Lochschmidt for his contribution to the implementation of the $W^5$proof-of-concept. Part of this research was conducted within the ADiWa project funded by the German Federal Ministry of Education and Research (BMBF) under grant number 01IA08006.

## REFERENCES

1. M. Beaudouin-Lafon. Designing interaction, not interfaces. In *Proc. AVI '04*, pages 15–22, New York, NY, USA, 2004. ACM.
2. P. Brandl, C. Richter, and M. Haller. Nicebook: supporting natural note taking. In *Proc. CHI '10*, pages 599–608, New York, NY, USA, 2010. ACM.
3. F. Heinrichs, J. Steimle, D. Schreiber, and M. Mühlhäuser. Letras: An architecture and framework for ubiquitous pen-and-paper interaction. In *Proc. EICS '10*, pages 193–198, New York, NY, USA, 2010. ACM.
4. K. Hinckley. *The Human Computer Interaction Handbook*, chapter Input Technologies and Techniques, pages 161 – 176. Lawrence Erlbaum Associates, Mahwah, NJ, USA, 2 edition, 2007.
5. C. Liao, F. Guimbretière, K. Hinckley, and J. Hollan. Papiercraft: A gesture-based command system for interactive paper. *ACM Trans. Comput.-Hum. Interact.*, 14(4):1–27, 2008.
6. O. Shaer, N. Leland, E. H. Calvillo-Gamez, and R. J. K. Jacob. The tac paradigm: specifying tangible user interfaces. *Personal Ubiquitous Comput.*, 8:359–369, September 2004.
7. B. Signer and M. C. Norrie. As we may link: a general metamodel for hypermedia systems. In *Proc. ER'07*, pages 359–374, Berlin, Heidelberg, 2007. Springer-Verlag.
8. B. Signer and M. C. Norrie. Paperpoint: a paper-based presentation and interactive paper prototyping tool. In *Proc. TEI '07*, pages 57–64, New York, NY, USA, 2007. ACM.
9. J. Steimle. Designing pen-and-paper user interfaces for interaction with documents. In *Proc. TEI '09*, pages 197–204, New York, NY, USA, 2009. ACM.
10. T. Tsandilas and W. E. Mackay. Knotty gestures: subtle traces to support interactive use of paper. In *Proc. AVI '10*, pages 147–154, New York, NY, USA, 2010. ACM.
11. N. Weibel, A. Ispas, B. Signer, and M. C. Norrie. Paperproof: a paper-digital proof-editing system. In *Proc. CHI '08*, pages 2349–2354, New York, NY, USA, 2008. ACM.
12. R. Yeh, C. Liao, S. Klemmer, F. Guimbretière, B. Lee, B. Kakaradov, J. Stamberger, and A. Paepcke. Butterflynet: a mobile capture and access system for field biology research. In *Proc. CHI '06*, pages 571–580, New York, NY, USA, 2006. ACM.
13. R. B. Yeh, A. Paepcke, and S. R. Klemmer. Iterative design and evaluation of an event architecture for pen-and-paper interfaces. In *Proc. UIST '08*, pages 111–120, New York, NY, USA, 2008. ACM.