Mandy Korzetz, Romina Kühn, Karl Kegel, Leon Georgi, Franz-Wilhelm Schumann, Thomas Schlegel

Milky Way: **A Toolbox for Prototyping Collaborative Mobile-Based Interaction Techniques**

SLUB
Wir führen Wissen.

TECHNISCHE UNIVERSITÄT DRESDEN

QUCOSA
Quality Content of Saxony

MilkyWay: A Toolbox for Prototyping Collaborative Mobile-Based Interaction Techniques

Mandy Korzetz[1], Romina Kühn[1], Karl Kegel[1], Leon Georgi[1], Franz-Wilhelm Schumann[1], and Thomas Schlegel[2]

[1] TU Dresden, Software Technology Group, Dresden, Germany
{firstname.lastname}@tu-dresden.de
[2] Karlsruhe University of Applied Science, Institute of Ubiquitous Mobility Systems, Karlsruhe, Germany
thomas.schlegel@hs-karlsruhe.de

Abstract. Beside traditional multitouch input, mobile devices provide various possibilities to interact in a physical, device-based manner due to their built-in hardware. Applying such interaction techniques allows for sharing content easily, e.g. by literally pouring content from one device into another, or accessing device functions quickly, e.g. by facing down the device to mute incoming calls. So-called mobile-based interaction techniques are characterized by movements and concrete positions in real spaces. Even though such interactions may provide many advantages in everyday life, they have limited visibility in interaction design due to the complexity of sensor processing. Hence, mobile-based interactions are often integrated, if any, at late design stages. To support testing interactive ideas in early design stages, we propose *MilkyWay*, a toolbox for prototyping collocated collaborative mobile-based interaction techniques. *MilkyWay* includes an API and a mobile application. It enables easily building up mobile interactive spaces between multiple collocated devices as well as prototyping interactions based on device sensors by a programming-by-demonstration approach. Appropriate sensors are selected and combined automatically to increase tool support. We demonstrate our approach using a proof of concept implementation of a collaborative Business Model Canvas (BMC) application.

Keywords: Prototyping · Mobile Devices · Device-Based Interaction · Collocated Interaction · Collaboration · Interaction Design.

1 Introduction

Smartphones as mobile devices have become a companion of our everyday life. Current devices innately provide numerous built-in sensors such as accelerometers and gyroscopes for sensing motions or orientation sensors and magnetometers for determining positions. Thus, mobile devices cover a wide range of interaction techniques and can support users in various situations. For example,

1

performing a short shaking gesture facilitates to provide feedback to map applications. Furthermore, turning a mobile device enables muting the device instead of pushing hardware buttons or navigating through complex menus. In addition, working and learning situations often involve collaborative activities which are increasingly supported by mobile device usage to avoid digital disruptions [19]. Collaborative activities comprise joining and leaving a group, creating and editing content, presenting and comparing results as well as sharing content.

In recent years, research has investigated various specific mobile-based interaction techniques to support these activities, for example, connecting devices by touching another device to add it to the group [9], comparing by ordering devices in a row to rank displayed content [17], or sharing by figuratively pouring content from one device to another [15]. Integrating mobile phones in a physical manner aims at fostering face-to-face collaboration because users are enabled to pay more attention to group activities instead of looking at screens. Although, there are numerous examples from research, mobile device-based interaction techniques still have limited visibility in interaction design [16].

As mobile-based interaction techniques base on sensors for connecting and building groups of collocated devices and also for detecting physical and motion-based interactions, designers and developers need advanced technical knowledge about hardware. Hence, mobile-based interactions are often integrated, if any, at late design stages. But to develop highly usable and useful systems along with addressing the user needs as a whole, it is particularly necessary to test and evaluate ideas in early design stages with interactive prototypes. Existing prototyping tools focus on single-device interactions with mobile phones [2] or specific device sensors [11]. Nevertheless research shows that mobile-based interaction techniques are more versatile. Leigh et al. [21] generally describe that smartphones are used as a tangible interfaces. Rico and Brewster [26] illustrate interactions that involve touching and moving mobile devices directly regardless of specific sensors. Furthermore, Lucero et al. [24,23,22] demonstrate various interactions with multiple devices where users share their mobile phones in collocated situations. As these examples show the versatility of these interaction techniques, prototyping tools are currently limited because they only address specific aspects of mobile-based interaction possibilities.

The *MilkyWay* toolbox is intended to support prototyping interactions that are generally invoked by deliberate device movements. In contrast to other tools, *MilkyWay* tries to minimize the required technical know-how, which means designers of mobile-based interactions can create and adjust prototyped interactions without any knowledge of sensor characteristics. Additionally, our toolbox supports developers with an API for easily building collaborative multi-device applications without further equipment, so-called mobile co-spaces [16]. Mobile co-spaces are formed by connecting multiple mobile devices that allow collaboration of multiple collocated users.

To describe our approach, we structured the remainder of the paper as follows: First, we give an overview of related work focusing on several kinds of prototyping tools. Then, we define design goals that consider the user require-

ments for implementing this kind of interaction techniques. Taking the design goals into account, we present our *MilkyWay* toolbox describing the including tools that target the development support. We illustrate and discuss the usage of the toolbox by describing the mobile application *MobiLean*. Concluding, we give an outlook on next steps.

2 Related Work

As stated by Ledo et al. [20], toolkit research plays an important role in the HCI community. In their work, they collected several representative papers that focus on toolkits or toolkit research. Thereby, they give a broad overview of related research but mainly investigate how these papers address evaluation aspects. For our work, it is important how toolkits facilitate easy prototyping of movement-based interactions for mobile phones without further equipment and devices, e.g. desktop PCs, but also allow prototyping of mobile co-spaces.

Interactions in ubiquitous environments, mid-air gestures and also tangible interfaces are related to mobile-based interaction techniques because they also deal with human movements. In the following, we therefore describe approaches that address prototyping in these research areas. One of the early works is the iStuff approach [4]. This extensive toolkit supports interaction designer to quickly create events in ubiquitous environments. Using additional equipment (so-called PatchPanels), they can complement everyday objects physically and utilize them to execute different actions, e.g. turning on and off the light by adding PatchPanels wherever wanted within a room. Keller et al. [10] extend Ballagas' et al. approach by providing virtual interactive surfaces to interact with everyday objects in a ubiquitous way. On top of the iStuff approach, Ballagas et al. [3] built iStuff mobile that is a visual programming environment, which enables the low-fidelity prototyping of mobile interactions. The toolkit provides several predefined sensors and interactions that can be combined individually. However, this limits the way the devices can be used because new sensors have to be added before they can be used. This leads to high maintenance effort due to the need of constantly updating the tool.

The requirements for authoring mid-air gestures are similar to the prototyping of mobile device-based interactions, because they also deal with human movements. Baytaş et al. [5] give an overview of tools for authoring mid-air gestures. They distinguish between tools that use graphs of the data from the sensors that detect movement, tools that use an own visual markup language, and tools that provide a timeline of frames. MAGIC [2] captures mid-air gestures of on-body devices such as wristwatches and visualize the data in form of sensor graphs within a desktop application. Developing motion-based gestures is done by demonstrating but allows only single device and capturing of a wristwatch device. EventHurdle [12] is a visual tool for authoring gesture events. The gestures are drawn in a separate application, code is generated and integrated into the prototype to trigger the interaction. EventHurdle is specialized on simple two-dimensional gestures, e.g. for touch pads, but lacks more complex three-

dimensional gestures. In contrast, M.Gesture [11] allows authoring directly on a mobile device by using a visual metaphor of a mass-spring, but concentrates on gestures based solely on accelerometers. As Baytaş et al. propose, we combine the two programming approaches demonstration and declaration to prototype mobile-based interactions.

Hartmann et al. [8] describe their approach on authoring sensor-based interactions by demonstrating using direct manipulation and pattern recognition. Their tool *Exemplar* visualizes data streams of connected sensors and enables a direct manipulation of the data streams using the PC's mouse. Although their tool support is extensive, recording and editing the interactions is done on another device, a desktop PC. This leads to a device-switch with a higher learning and execution effort than authoring directly with the sensor device. Klompmaker et al. [14] present the INDiE approach that consists of a network protocol, a device abstraction and a software development kit (SDK). Whereas the network protocol facilitates connection establishment and data exchange, the device abstraction approach aims at supporting rapid prototyping of multimodal interactions with focus on virtual reality (VR). Similar to Hartmann et al. [8], this approach needs a server component to manipulate and aggregate the received data from the sensors. This leads to a higher dependency on a fast WiFi connection to communicate with the server component as well as the server itself. Ajaj et al. [1] describe their Real/Virtual-Device/Task (RVDT) model that aims at supporting the design of multimodal/multi-view interfaces. The presented design space binds input devices to spatial tasks. Although the approach is extensive in terms of the degrees of freedom, it only addresses graphical output modalities. Klemmer et al. [13] present a Wizard of Oz (WOz) approach for tangible user interfaces to support designers by providing WOz generation and removal of input. This approach is useful for simulating hardware that is not (yet) available.

Although, there are already various approaches that present toolkits and frameworks to support developers, current work lacks approaches that allow for prototyping mobile-device interactions which on the one hand are based on motions and on the other hand are often combined with other mobile devices to allow collaboration. Moreover, most approaches require additional applications and PCs, so that learning is more difficult and end-users can not be involved easily.

3   Design Goals

Klemmer et al. [13] derived functionality that should be provided for tangible user interfaces. We extended their work and tailored the functional requirements to collaborative mobile-based interaction techniques. Geiger et al. [6] also describe requirements for an easy-to-use framework for prototyping hybrid user interfaces that combines 2D, 3D and haptic interfaces. Based on these works, we derived the following requirements for the design of the *MilkyWay* toolbox.

Collaborative Mobile-Based Interaction Techniques. For a better understanding of the domain of mobile-based interaction techniques, we investigated several
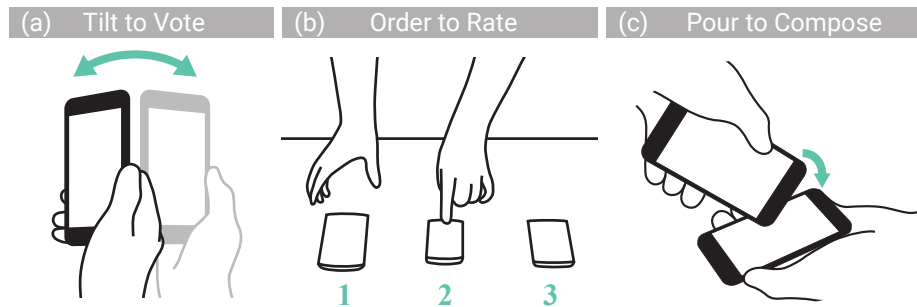
Fig. 1: Examples for mobile-based interaction techniques: (a) Tilt to Vote [18], (b) Order to Rate [17], and (c) Pour to Compose [15].

interactions from literature (see Fig. 1). Interaction characteristics can be distinguished between interactions that are for individual use, e.g. tilting the device to accept or reject a solution (Fig. 1a) or facing down the device to mute incoming calls, as well as for collaborative use, e.g. ordering devices on a table to rate the displayed content (Fig. 1b) or merging content by literally pouring content from one device to another (Fig. 1c). Prototyping tools should support easily building up such mobile interactive spaces between multiple collocated devices. The examples also show the important role of *spatiality* and *motion*. The intended interaction either can depend on relative positions between or a specific arrangement of devices that trigger system functions as shown in Figure 1b, or is caused by a movement (Fig. 1a).

User-Centered Design of Interactions. Nielsen et al. [25] stated that gestures and thus mobile-based interactions should not be motivated by the easiness of implementation (technology-based approach), but instead should be developed together with end-users to ensure an intuitive and ergonomic interface. Our tools aim at providing user involvement at different prototyping stages. This enables collecting interaction ideas to specific system functions with participants of a user group, improving interim results, for example, by gathering different variants of how to perform interactions, and evaluating interactions to either test the interaction itself or to test the interplay with other interactions and system functions.

Facilitating the Processing and Use of Device Sensors. Especially designers but also developers often have an extensive knowledge about graphical aspects of creating mobile user interfaces, but lack advanced technical knowledge in terms of sensor hardware. Prototyping of mobile-based interactions which are based on device sensors therefore should reduce the programming effort. To tackle this issue, we propose an easy programming-by-demonstration approach where relevant sensors are selected and combined automatically. By using the respective device for prototyping, there is no need for additional equipment such as a PC

for editing the prototype or an additional server for collecting sensor data. Furthermore, the usage of the respective devices reduces effort both regarding the need of applying different devices for prototyping and to find metaphors that create an understanding of how the mobile device is utilized. The toolbox should also facilitate the reuse of prototyped interactions.

4   The *MilkyWay* Toolbox

The *MilkyWay* toolbox consists of the *Spaces* API and the *Gaia* mobile application. The *Spaces* API facilitates implementing so-called mobile co-spaces, where multiple nearby mobile devices are enabled to communicate equally privileged with the objective to collaborate. This aim includes, for example, sharing created content within a group or voting on several working results. The *Gaia* application aims at mainly supporting designers of mobile device-based interaction techniques by enabling an easy recording by demonstration of interactions. The tools can be used in parallel by designers and developers of mobile-based interaction techniques. The following sections describe the tools in detail as well as how they can be utilized for prototyping.

4.1   The *Spaces* API

The application programming interface (API) *Spaces* supports developers in prototyping group communication between multiple mobile devices. *Spaces* provides developers with methods to easily built-up and maintain stable connection within mobile co-spaces. Further, the API automates handling connection-based functionality such as sending and receiving data or joining and leaving co-spaces. The *Spaces* API is based on the Nearby Connections 2.0 API[1], an API that is originally intended to enable position-based information to users, e.g. for advertising. For convenience, connections between mobile devices are fully-offline by combining Bluetooth, Bluetooth Low Energy (BLE), and WiFi and therefore enable fast, secure data transfers. Moreover, users are not prompted to turn on Bluetooth or WiFi, because the API enables these features as they are required. The Nearby Connections 2.0 API provides very extensive functions specialized for position-based advertising, but for prototyping and implementing mobile co-spaces developers have to write a lot of boilerplate code[2] and the effort for configuring peer-to-peer networks which is needed for the communication within a mobile co-space is very high. *Spaces* combines the API functionality to multi-device communication so that configuration effort is low and using the API becomes more simple and intuitive. As collaborative scenarios require a stable connection between mobile devices, the *Spaces* API enables a fully automated reconnection management in case of connection failures.

---

[1] `https://developers.google.com/nearby/connections/`
[2] Boilerplate code is code that has to be included in many places with little or no alteration.

4.2 The *Gaia* Mobile Application

Current mobile devices include numerous built-in sensors to measure motion, orientation and other environmental conditions[1]. To support designers and developers with little technical knowledge about sensor characteristics, the *Gaia*[2] application facilitates the development of mobile-based interactions. Designers, developers, and also end-users can easily record interactions with the mobile application. For this purpose, app users have to demonstrate the respective movement or device constellation they want to achieve by directly manipulating the device(s). To ensure a higher recognition rate, the interaction has to be repeated a couple of times, approximately 5 to 10 repetitions. *Gaia* estimates dynamically which sensors are appropriate to recognize the demonstrated interaction by an algorithm which compares absolute sensor values. Advanced *Gaia* users can adjust the suggested sensors post hoc. Interrelated interactions can be grouped, e.g. simple moving left, right, up and down on a table can be grouped as "sliding interactions". To compare a stored recorded interaction with a new record, *Gaia* uses an optimized dynamic time warping algorithm for each relevant sensor type. Advanced users can have a look at the sensor graphs to understand single recordings.

For prototyping collaborative interactions that consist of multiple parts distributed on multiple devices, e.g. pouring content from one device to another device [15], *Gaia* allows composing interaction parts. Users of the *Gaia* application can choose from a list which interaction parts belong together. Composing interaction parts is implemented as simple event processing architecture, which means if two or more interaction events occur, the composed interaction is recognized. For this purpose, *Gaia* itself uses the *Spaces* API to easily exchange event messages.

4.3 Prototyping with the *MilkyWay* Toolbox

The prototyping process by using *MilkyWay* is structured as shown in Figure 2. Whereas developers are especially supported by using the *Spaces* API to prototype system functionality for collaborative interactions, designers can work with the *Gaia* mobile application to prototype the mobile-based interactions related to the system functions. Designers typically collect different ideas for interacting mobile-based with the system. The ideas can be demonstrated with *Gaia* once to capture it. To refine first ideas and capture performing variants, interaction ideas have to be recorded multiple times. In case the interaction involves multiple devices, each part has to be repeated about 5 to 10 times. The last step includes composing single interaction parts to the collaborative mobile-based interaction. Designers and developers of mobile device-based interactions can work efficiently in parallel. At any stage they can exchange interim results,

---

[1] As example the sensor types of typical mobile devices with Android operating system: `https://developer.android.com/guide/topics/sensors/sensors_overview`

[2] Gaia is a spacecraft of the European Space Agency (ESA) for astrometry and measures the positions, distances and motions of stars with unprecedented precision.
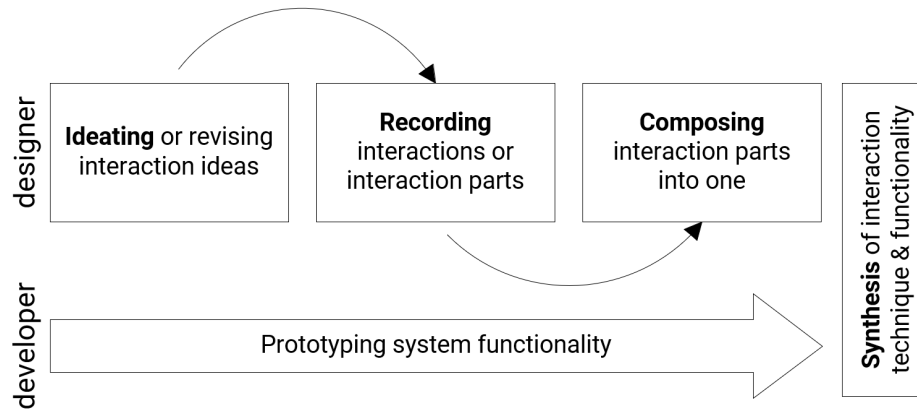
Fig. 2: Prototyping process with the *MilkyWay* toolbox. Designers and developers of mobile device-based interactions can work efficiently in parallel and exchange interim results. Users can be involved at any prototyping stage.

e.g. integrating first ideas for a mobile-based interaction, or vice versa, working results of test functionality can be used to generate mobile-based interactions with users itself. To enable a user-centered design approach, users can be involved at any prototyping stage to develop easy to perform and remember interactions. In the first stage, users can directly demonstrate their interaction ideas by using the *Gaia* app either by simulating system functionality or using prototyped functionality from the developer. In the next stage, the recording stage, users can be involved to record variants of concrete interaction ideas. The main goal of integrating users at this stage is to ensure that different performances can be recognized and concrete interactions and interaction parts can be refined. At the third stage, users should be involved to test the prototyped interactions and give feedback for improvements. The last step of prototyping comprises combining prototyped system functionality and revised mobile-based interaction techniques to an extensive interactive mobile app prototype.

5   Applying *MilkyWay* to a Mobile System for Creating BMCs Collaboratively

Collaboration usually includes working together as well as working individually, so-called mixed-focus collaboration [7]. Kuehn and Schlegel [19] identified several activities that are typically performed in mixed-focus collaboration, e.g. creating and sharing content. Furthermore, they presented an example scenario for the key activities addressing the creation process of a business model canvas[1] (BMC) with several building blocks. In their example, a group of students performs this

---

[1] The traditional paper tool for BMC can be found on `https://strategyzer.com/`

(a) Initiating collaboration.

(b) Creating and presenting results.

(c) Selecting content.
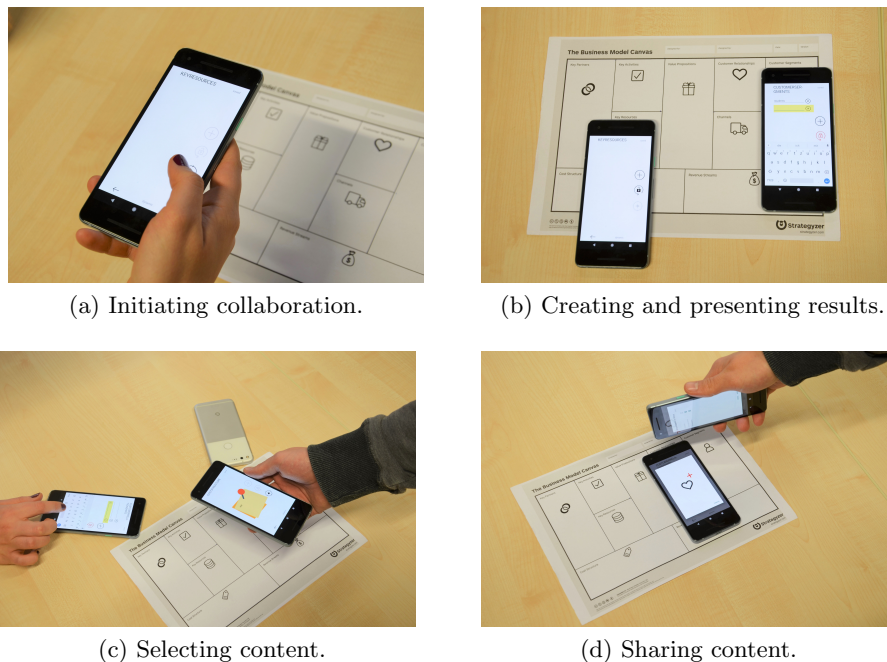
(d) Sharing content.

Fig. 3: Initiating collaboration by detecting blocks using image recognition (Fig. 3a). Creating text entries collaboratively by adding, editing or removing content and presenting results by highlighting important content (Fig. 3b). Selecting relevant content for sharing (Fig. 3c). Sharing content by merging (preselected) entries from different blocks after working individually (Fig. 3d).

collaboration process using mobile phones. The activities involve initiating collaboration and assigning blocks to group members, creating text entries within the blocks, presenting and discussing their interim results, sharing and merging individual results to receive an overall solution and finishing collaboration. We use this example scenario to illustrate how the *MilkyWay* toolbox supports several collaboration activities and respective prototyping of mobile-based interaction techniques.

We implemented *MobiLean* (Fig. 3), a mobile application for Android devices that enables the creation of BMCs collaboratively. The usage of *MilkyWay* supports developers in implementing collaborative interactions and designers in trying several interaction techniques for different activities. The application has the following functionalities according to the *collaboration activities* [19]:

– Initiating collaboration (Fig. 3a): connecting devices; joining sessions; inviting to sessions.
– Creating content (Fig. 3b): adding, editing, and removing text entries within BMC blocks; creating new projects and sessions.

- Presenting content (Fig. 3b, right): highlighting text entries.
- Comparing results: voting on results.
- Sharing content (Fig. 3d): selecting content; sharing complete entry sets; merging entry sets with other entries.
- Finishing collaboration: disconnecting devices; leaving sessions.

While we implemented core functionality that is specific for BMCs, e.g. building different blocks and editing block entries, we used the *MilkyWay* toolbox to create functionalities for collaborating and to establish mobile interactive spaces (mobile co-spaces) between several devices. *MobiLean* uses the *Spaces* API to connect the devices easily. A device can act as host and can invite further devices (guests) to join a session. Each guest can easily join such hosted sessions. As it is common in mixed-focus collaboration to work individually, too, re-joining a session can be performed automatically. Especially for designers and developers, the *Spaces* API is an easy tool to establish multi-device connections.

The *Gaia* app is used to teach the interaction techniques by applying the programming-by-demonstration approach. In the current version of *MobiLean*, direct voting interactions [17] and merging interactions [15] (see Fig. 3d) are implemented. Applying the *Gaia* app, these interaction techniques could already be recorded to be used during the BMC creation. As described in section 4.2, each interaction was performed several times while *Gaia* estimated the utilized sensors. Especially for developers, there is an advanced setting to adjust the suggested sensors post hoc. For example, for *Shift to Vote*, one of Kühn's et al. voting interaction technique [17], several single up and down movements of the device on a table are grouped as the mentioned interaction. Furthermore, designers can group interactions that belong to the category of "voting". Designers can use this functionality to create and refine interaction techniques. Furthermore, end-users can record different interactions to extend the variety of the way how interactions can be performed. The *MilkyWay* toolbox simplifies the development of several interaction techniques tremendously, especially for those, who are not familiar with programming. We aim at extending *MobiLean* with additional interaction techniques to further investigate our toolbox and its application. While implementing *MobiLean* we could observe, so far, that prototyping collaborative mobile-based interaction techniques can be speed up with *MilkyWay*. We plan to investigate the exact speed up rate in future work.

6   Discussion

We implemented *MobiLean* as proof of concept application to investigate the usage of our *MilkyWay* toolbox. Although, the application is still in development, we got a better understanding of how the toolbox can support designers, developers, and end-users of mobile-based interaction techniques. Based on our design goals for *MilkyWay* in section 3 and the *MobiLean* implementation (section 5), we discuss the following aspects.

6.1   Coverage of Collaborative Mobile-Based Interaction Techniques

The *MilkyWay* toolbox allows prototyping of interaction techniques that base on movements of one or more devices, but also on more static constellations of one or multiple devices if built-in sensors are involved. The toolbox therefore covers a wide range of possible mobile-based interactions that are appropriate for individual but also for collaborative use. As we build on all available sensors of a device and automatically select and combine them, designers and developers are not restricted concerning the used sensor data.

6.2   Involving Users to Interaction Design

To ensure intuitive and ergonomic interactions with the system, that has to be developed, *MilkyWay* enables to involve users at any stage of interaction design. They can be involved (1) to collect a set of interaction ideas, (2) to record variants of interactions, and (3) to conclusively evaluate interactions as described in section 4.3. As interactions are captured by directly manipulating the mobile devices themselves, end-users of mobile-based interactions can easily be involved.

6.3   Facilitating the Use of Device Sensors

Understanding sensor characteristics needs advanced technical knowledge, which is seldom pronounced within the group of designers and developers of mobile applications. *MilkyWay* simplifies the usage of sensor data by allowing a fully automated selection and combination of sensors. Users only have to demonstrate their interaction idea with the respective device and *Gaia* handles the translation into code. In doing so, no additional equipment is needed such as a PC/server for, e.g. adjusting or processing raw sensor data. Moreover, implementing multi-device communication is facilitated by the *Spaces* API by providing and enclosing methods that are needed for collaborative interactions.

6.4   Limitations and Future Work

There remains some future work to improve the usability of static mobile-based interactions (e.g. *Order to Vote* [17]). As interactions have to be demonstrated for prototyping, each possible arrangement has to be performed. So, prototyping of static interactions is possible but cumbersome. We would like to find strategies to simplify prototyping of interactions such as *Order to Vote* with an improved version of *MilkyWay*. Combining parts of interactions with *Gaia* is currently realized by a simple "if this than that" rule. We could imagine that there are cases that require more flexibility. So, we plan to implement algorithms to allow further combining strategies. We also plan to improve reusing of prototyped mobile-based interactions. Currently, recognition of interactions works best on the device type where they were recorded because sensor data of each device type slightly differs. In next steps we will test how this can be improved, e.g. with mixing training data of different devices to enhance accuracy to enable reusing of interaction data for further development stages.

7   Conclusion

We presented *MilkyWay*, a toolbox for prototyping mobile-based interaction techniques already in early design stages. The toolbox aims at supporting designers and developers to prototype complex mobile-based interaction techniques based on movements and/or relative positions detected by the devices' built-in sensors. Therefore, users of *MilkyWay* need no or little knowledge of sensor characteristics. And as interactions are prototyped with the respective device, end-users can be easily involved. Developers are supported by an API that facilitates and aims at speeding up the implementation of collaborative communication between devices. As a proof of concept, we implemented *MobiLean*, a collaborative BMC application that applies the *MilkyWay* toolbox.

As described in section 6 we plan to facilitate prototyping of static mobile-based interactions, expand combination methods of interaction parts, and improve reusing interaction data. We would also like to use parts of *MilkyWay* to enable individualizing interaction techniques for end-users in mobile applications that are production-ready or already on the market.

We currently use *MilkyWay* to prototype collaborative mobile-based interactions for creating BMCs. We intend to implement further techniques, e.g. anonymous voting interactions [18]. Our next steps will also include an evaluation with mobile designers and developers to get additional impressions and suggestions for improvements of the *MilkyWay* toolbox, and further ideas for potential additional tools to support prototyping of mobile-based interaction techniques.

References

1. Ajaj, R., Jacquemin, C., Vernier, F.: RVDT: A Design Space for Multiple Input Devices, Multiple Views and Multiple Display Surfaces Combination. In: Proceedings of the International Conference on Multimodal Interfaces (ICMI-MLMI '09). pp. 269–276 (2009). https://doi.org/10.1145/1647314.1647372
2. Ashbrook, D., Starner, T.: MAGIC: A Motion Gesture Design Tool. In: Proceedings of the 28th international conference on Human factors in computing systems (CHI '10). pp. 2159–2168. ACM (2010). https://doi.org/10.1145/1753326.1753653
3. Ballagas, R., Memon, F., Reiners, R., Borchers, J.: iStuff Mobile: Rapidly Prototyping New Mobile Phone Interfaces for Ubiquitous Computing. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07). pp. 1107–1116 (2007). https://doi.org/10.1145/1240624.1240793
4. Ballagas, R., Ringel, M., Stone, M., Borchers, J.: iStuff: A Physical User Interface Toolkit for Ubiquitous Computing Environments. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03). pp. 537–544 (2003). https://doi.org/10.1039/F29767200988

5. Baytaş, M.A., Yemez, Y., Özcan, O.: User Interface Paradigms for Visually Author-ing Mid-Air Gestures: A Survey and a Provocation. In: 1st International Workshop on Engineering Gestures for Multimodal Interfaces (EGMI '14). pp. 8–14 (2014), `http://ceur-ws.org/Vol-1190/`.

6. Geiger, C., Fritze, R., Lehmann, A., Stöcklein, J.: HYUI: A Visual Framework for Prototyping Hybrid User Interfaces. In: Proceedings of the 2nd international conference on Tangible and embedded interaction (TEI '08). pp. 63–70 (2008). https://doi.org/10.1145/1347390.1347406

7. Gutwin, C., Greenberg, S.: Design for Individuals, Design for Groups: Tradeoffs Be-tween Power and Workspace Awareness. In: Proceedings of the 1998 ACM Confer-ence on Computer Supported Cooperative Work (CSCW '98). pp. 207–216 (1998). https://doi.org/10.1145/289444.289495

8. Hartmann, B., Abdulla, L., Mittal, M., Klemmer, S.R.: Authoring Sensor-based Interactions by Demonstration with Direct Manipulation and Pattern Recogni-tion. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07). pp. 145–154 (2007). https://doi.org/10.1145/1240624.1240646

9. Jokela, T., Chong, M.K., Lucero, A., Gellersen, H.W.: Connecting Devices for Collaborative Interactions. Interactions **22**(January 2016), 1–5 (6 2015). https://doi.org/10.1145/2776887

10. Keller, C., Kühn, R., Engelbrecht, A., Korzetz, M., Schlegel, T.: A Prototyping and Evaluation Framework for Interactive Ubiquitous Systems. In: Proceedings of the First International Conference on Distributed, Ambient, and Pervasive Inter-actions. pp. 215–224 (2013). https://doi.org/10.1007/978-3-642-39351-8_24

11. Kim, J.W., Kim, H.J., Nam, T.J.: M.Gesture: An Acceleration-Based Gesture Au-thoring System on Multiple Handheld and Wearable Devices. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16). pp. 2307–2318. ACM (2016). https://doi.org/10.1145/2858036.2858358

12. Kim, J.W., Nam, T.J.: EventHurdle: Supporting Designers Exploratory Interaction Prototyping with Gesture-Based Sensors. In: Proceedings of the SIGCHI Confer-ence on Human Factors in Computing Systems (CHI '13). pp. 267–276 (2013). https://doi.org/10.1145/2470654.2470691

13. Klemmer, S.R., Li, J., Lin, J., Landay, J.A.: Papier-Mâché: Toolkit Sup-port for Tangible Input. In: Proceedings of the SIGCHI Conference on Hu-man Factors in Computing Systems (CHI '04). pp. 399–406. Vienna (2004). https://doi.org/10.1016/j.jjcc.2014.08.007

14. Klompmaker, F., Schrage, K., Reimann, C.: INDiE: A Framework for Human Com-puter Interaction in Distributed Environments. In: Proceedings of the 6th Inter-national Conference on Mobile Technology, Application & Systems (Mobility '09). pp. 26–29 (2009). https://doi.org/10.1145/1710035.1710061

15. Korzetz, M., Kühn, R., Heisig, P., Schlegel, T.: Natural Collocated Inter-actions for Merging Results with Mobile Devices. In: Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '16). pp. 746–752. ACM (2016). https://doi.org/10.1145/2957265.2961839

16. Korzetz, M., Kühn, R., Schlegel, T.: Turn it, Pour it, Twist it: A Model for De-signing Mobile Device-Based Interactions. In: Proceedings of the 5th International Conference on Human-Computer Interaction and User Experience in Indonesia (CHIuXiD '19). ACM (2019)

17. Kühn, R., Korzetz, M., Büschel, L., Korger, C., Manja, P., Schlegel, T.: Nat-ural Voting Interactions for Collaborative Work with Mobile Devices. In: Pro-

13

ceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16). pp. 2570–2575. ACM (2016). https://doi.org/10.1145/2851581.2892300

18. Kühn, R., Korzetz, M., Büschel, L., Schumann, F.W., Schlegel, T.: Device-Based Interactions for Anonymous Voting and Rating with Mobile Devices in Collaborative Scenarios. In: Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia (MUM '16). pp. 315–317. ACM (2016). https://doi.org/10.1145/3012709.3016067

19. Kühn, R., Schlegel, T.: Mixed-Focus Collaboration Activities for Designing Mobile Interactions. In: Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '18). pp. 71–77. ACM (2018). https://doi.org/10.1145/3236112.3236122

20. Ledo, D., Houben, S., Vermeulen, J., Marquardt, N., Oehlberg, L., Greenberg, S.: Evaluation Strategies for HCI Toolkit Research. In: Proceedings of the 2018 Conference on Human Factors in Computing Systems (CHI '18). p. 17 (2018). https://doi.org/10.1134/S1819712414020056

21. Leigh, S.w., Schoessler, P., Heibeck, F., Maes, P., Ishii, H.: THAW: Tangible Interaction with See-Through Augmentation for Smartphones on Computer Screens. In: Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '14). pp. 89–96. ACM (2015). https://doi.org/10.1145/2677199.2680584

22. Lucero, A., Holopainen, J., Jokela, T.: Pass-Them-Around: Collaborative Use of Mobile Phones for Photo Sharing. In: Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems (CHI '11). pp. 1787–1796 (2011). https://doi.org/10.1145/1978942.1979201

23. Lucero, A., Keränen, J., Jokela, T.: Social and Spatial Interactions: Shared Colocated Mobile Phone Use. In: CHI '10 Extended Abstracts on Human Factors in Computing Systems (CHI EA '10). pp. 3223–3228. ACM (2010)

24. Lucero, A., Keränen, J., Korhonen, H.: Collaborative Use of Mobile Phones for Brainstorming. In: Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '10). pp. 337–340. ACM (2010). https://doi.org/10.1145/1851600.1851659

25. Nielsen, M., Störring, M., Moeslund, T.B., Granum, E.: A Procedure for Developing Intuitive and Ergonomic Gesture Interfaces for HCI. In: Camurri, A., Volpe, G. (eds.) Gesture-based Communication in Human-Computer Interaction. vol. 2915/2004, pp. 409–420. Springer (2003). https://doi.org/10.1007/978-3-540-24598-8_38

26. Rico, J., Brewster, S.: Usable gestures for mobile interfaces: evaluating social acceptability. In: Proceedings of the 28th international conference on Human factors in computing systems (CHI '10). p. 887. ACM (2010). https://doi.org/10.1145/1753326.1753458