Towards Gesture-Based Process Modeling on Multi-touch Devices

Jens Kolb, Benjamin Rudner, and Manfred Reichert

Institute of Databases and Information Systems
Ulm University, Germany
{jens.kolb,benjamin.rudner,manfred.reichert}@uni-ulm.de
http://www.uni-ulm.de/dbis

Abstract. Contemporary tools for business process modeling use menu-based interfaces for visualizing process models and interacting with them. However, pure menu-based interactions have been optimized for applications running on desktop computers and are limited regarding their use on multi-touch devices. At the same time, the increasing distribution of mobile devices in business life as well as their multi-touch capabilities offer promising perspectives for intuitively defining and adapting business process models. Additionally, multi-touch tables could improve collaborative business process modeling based on natural gestures and interactions. In this paper we present the results of an experiment in which we investigate the way users model business processes with multi-touch devices. Furthermore, a core gesture set is suggested enabling the easy definition and adaption of business process models on these devices. Overall, gesture-based process modeling and multi-touch devices allow for new ways of (collaborative) business process modeling.

Keywords: process modeling, gestures, user-centered process modeling, multi-touch application.

1 Introduction

Multi-touch devices have been increasingly used in companies during the last years and have become more and more useful for the daily work of business people. Multi-touch devices used in companies range from smartphones and tablets to multi-touch tables and walls. Obviously, screen size affects both mobility and application areas. While smartphones are primarily used for mobile communication (e.g., mailing and information retrieval), tablets and multi-touch tables are applied for enabling collaborative tasks (e.g., joint editing of a business document). Consequently, multi-touch devices may be applied to business process modeling as well. In particular, portable multi-touch devices can be used to capture and model business processes while interviewing process participants. Multi-touch devices with larger screens (e.g., multi-touch tables), in turn, can be used to collaboratively model, discuss and change business processes [1]. However, traditional process modeling tools have not been designed with multi-touch

M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 280–293, 2012.

devices in mind and do not take their specific properties (e.g., small screen size) and interaction possibilities (e.g., gesture-based interaction) into account [2,3]. In this paper we introduce a set of gestures to easily and intuitively model business processes using multi-touch devices. This gesture set enables process designers to define, visualize and adapt business processes in an intuitive and comprehensible way. The focus of the paper is on the introduction of this core gesture set, and less on a concrete user interface design. The suggested core gesture set is applicable to all screen sizes of multi-touch devices. It has been developed in the context of the proView[1] project [4], which aims at user-centered process management. This includes techniques enabling personalized process model visualizations (e.g., process views [5,6]) as well as intuitive process model representations (e.g., diagrams and trees [7]). Gesture-based modeling complements this work with sophisticated and intuitive concepts for interacting with process designers.

The remainder of this paper is structured as follows: Section 2 gives background information on multi-touch applications and summarizes the abstract modeling operations to be supported by the gesture set. Section 3 presents the results of an experiment we conducted to analyze how users interact with multi-touch devices and which kind of gestures they are using to model and change processes. These results provide the foundation of the core gesture set introduced in Section 4. Finally, Section 5 concludes the paper with a short summary and outlook.

2 Background

This section describes background information needed for the understanding of this paper. Section 2.1 introduces general interaction concepts, which can be used to interact with an application on a multi-touch device. Section 2.2 then lists common characteristics of multi-touch applications. Finally, Section 2.3 presents a core function set required for modeling and adapting business processes.

2.1 Multi-touch Interaction Concepts

Generally, users may interact in different ways with a multi-touch application. First, concepts used in the context of conventional desktop applications may be applied to multi-touch application as well; e.g., *menu-based interactions* use menus and toolbars to provide available functions to users. As an advantage, this concept allows users to easily *explore* the application when searching for a specific function. However, displaying menus and toolbars requires space on the screen, which is limited on devices with small screens (e.g., smartphones). Hence, menu-based interaction should primarily be used for multi-touch applications running on larger screens.

[1] http://www.uni-ulm.de/in/iui-dbis/forschung/projekte/proview-project.html

Gesture-based interaction, in turn, uses gestures for selecting functions of the multi-touch application as well as for interacting with them. Thereby a (multi-touch) *gesture* constitutes a movement of one or more fingers on the screen of the multi-touch device. This movement is then recognized and interpreted by the application. For example, in a slideshow of pictures the *wipe gesture* (i.e., wiping with one finger from right to left) can be used to display the next picture. Generally, gestures can be categorized into *physical gestures* and *symbolic gestures* [8]. Physical gestures manipulate the virtual objects on the screen directly, e.g., by dragging a virtual element on the screen. Symbolic gestures, in turn, are related to the function to be executed (e.g., drawing a plus to add an object). As opposed to menu-based interactions, gestures do not require any space on the screen in order to display menus or other graphical elements. However, as a barrier for their application, users do not always know which functions are supported by a multi-touch application and with which gestures they can be selected. The latter is especially crucial for novices and unexperienced users of the multi-touch application. Usually, this problem is addressed through a tutorial provided the first time the user starts the application. Finally, a *hybrid interaction* based on both menu- and gesture-based interactions can be realized. For example, a menu bar may be displayed at the top or bottom of the screen offering the most important functions. Additionally, well-known gestures may be supported, e.g., the above mentioned wipe gesture.

2.2 Characteristics of Multi-touch Applications

Besides their interaction concepts multi-touch applications—or to be more precise their multi-touch capabilities—can be described through six properties.

C1 (Screen Occlusion): The user interface design of a multi-touch application should consider that users directly interact with the application, but might cover screen areas with their hand [9]. For example, dragging an object from the top-left corner of a tablet to its bottom-right corner will be a difficult task if large parts of the screen are covered by the user's hand.

C2 (Handling Precision): Conventional desktop applications, require a mouse pointer to interact with them. This pointer scales with the screen resolution in order to ensure high precision of the interactions. In multi-touch applications, in turn, the 'interaction device' is the user's finger [10]. As opposed to the mouse pointer, a finger does not scale up or down when changing screen sizes. Accordingly, the handling precision of a multi-touch application changes with the size of an object the user has to touch. Studies have shown that an object should have a size of 11.52 mm or more to have a hit probability of at least 95% [11]. Regarding multi-touch process modeling, it will be crucial to achieve a high handling precision.

C3 (Fatigue of Extremities): When using a computer mouse, usually, the hand does not move a lot. Typically, the arm lays on the table and the mouse is moved with the fingers to reach the corners of the screen with the mouse pointer. Interacting with multi-touch applications, in turn, frequently requires the movement of the whole arm. Further, the arm does not lay on a table, but has to be hovered in the air during the interaction with the multi-touch application [12]. Obviously, the degree of movement strongly depends on the screen size.

C4 (Number of Supported Fingers): Compared to conventional desktop applications with a mouse pointer, a multi-touch application typically supports more than one touch point. Generally, the maximum of touch points an application supports is limited through the underlying hardware and operation system. For example, Apples iPad distinguishes between 11 touch points (i.e., fingers) [13]. While this number is sufficient for single user applications, multi-user applications should be able to even support more touch points, e.g., in order to allow for the concurrent modeling of processes on a multi-touch table.

C5 (Number of Concurrent Users): Multi-touch applications may be concurrently used by multiple users. Obviously, the degree of concurrency is limited by the screen size. While concurrent interactions with a tablet would effect each other, concurrent use of an application on a multi-touch table (e.g., joint editing or modeling) might improve the way of working [14]. Especially in the context of concurrent process modeling, this should be exploited, particularly in process management projects involving multiple modelers [15].

C6 (Usage of Common Interaction Concepts): As known from conventional desktop applications, there are interaction concepts representing de-facto standards. Examples include the menu bar on the top of the application window or *File* as first entry of such a menu bar. Such recurrent patterns help users to intuitively interact with applications. The same applies to multi-touch applications and especially gesture-based interactions; e.g., the *pinch gesture* is typically used for zooming. However, there are only few gestures that have been used in a consistent manner so far. Therefore, [16] suggests a gesture dictionary with the intention that different applications show similar behavior in connection with a specific gesture.

The introduced characteristics of multi-touch devices show their wide range of possible applications. Obviously, these characteristics should be taken into account when designing gestures for business process modeling.

2.3 Required Functions for Modeling Business Processes

This section introduces a core function set required to model and change processes. This function set is derived from the change patterns presented in [17,18]. In the context of this paper we assume that process models are well-structured and are based on an activity-centric modeling language. Furthermore, a process model is displayed to process designers by using advanced layout algorithms, i.e., the process designer cannot position single process elements arbitrarily on the screen. Finally, an initial process model consists of a start and an end node connected through a control-flow edge. Based on modeling functions F1-F8, as introduced in the following, the (multi-touch) application ensures that the elements of a process model are connected.

Function *F1 (Insert Activity)* inserts a single activity into a process model between two existing nodes. *F2 (Insert Surrounding Block)* encloses a process fragment with an XOR (AND) block, i.e., an XOR (AND) split with corresponding XOR (AND) join gateway. The enclosed process fragment may be also "empty" leading to the creation of an XOR (AND) block having one "empty" branch. Function *F3 (Insert Branch)* extends an existing XOR (AND) block with an empty branch between the splitting and joining gateway.

Function *F4 (Renaming Element)* changes the name of a process element, e.g., an activity or data element. To delete process elements, function *F5 (Delete Element)* can be applied. The process element may be an activity, a data element, or a gateway. Comparable to F1, function *F6 (Insert Data Element)* adds a new data element to the process model. This data element, in turn, can be connected to activities using function *F7 (Insert Data Edge)*. Finally, function *F8 (Extract/Inline Sub-process)* extracts a sub-process based on a selected set of activities or inlines this sub-process [19]. This function is especially required when modeling complex process hierarchies. An overview of the core function set is given in Table 1.

Table 1. Core Function Set

Core Function Set	
F1 Insert Activity	F5 Delete Element
F2 Insert Surrounding Block	F6 Insert Data Element
F3 Insert Branch	F7 Insert Data Edge
F4 Renaming Element	F8 Extract/Inline Sub-process

In this paper, we restrict ourselves to this core function set, which provides elementary functions required to model and adapt processes. Obviously, additional functions are required for a sophisticated process modeling tool (e.g., to specify attributes or to undo/redo modeling steps).

3 An Experiment on Multi-touch Process Modeling

Experimental research in the context of business process management has already shown promising results regarding user-centric process support [20,21,22,23]. In our research, we conducted, a user experiment on multi-touch process modeling before developing a corresponding gesture set. Using the *Goal Definition Template* as presented in [24], the goal of the experiment is defined as follows:

Analyze	*multi-touch process modeling interaction*
for the purpose of	*evaluating*
with respect to their	*intuitive usage*
from the point of view of	*the researchers and developers*
in the context of	*students and research assistants.*

Taking this goal definition, our experiment evaluates multi-touch interactions applied by users—or to be more precise by students and research assistants—when modeling and changing processes on multi-touch devices. The results of this experiment are then used to develop a consistent set of modeling gestures covering the core process modeling functions introduced in Section 2.3. Based on this goal definition, the following hypothesis is derived:

Users intuitively use multi-touch gestures when modeling and changing process models on multi-touch devices.

Section 3.1 refines the goal and explains the experiment setting. Section 3.2 discusses the results of the experiment.

3.1 Experiment Setting

The experiment is designed as a *multi-test within object study* [24], i.e., having two groups of participants and one object. *Students* (as first group) do not have a broad background in process modeling and process management, whereas the second group consists of research assistants being more experienced with process modeling. Furthermore, the experiment is divided into two parts. In the first one, demographic and background information of the participants is collected (i.e., gender, experience with multi-touch devices, profession). In the second part of the experiment, each participant has to modify an existing process model (i.e., object) in eight different steps. Each step includes a description explaining to participants what they have to do in order to perform the respective process modeling task (cf. Section 2.3). For example, step "Think of a way to create a new activity between *Make Up Package* and the *XOR Branch*" can be accomplished using core function *F1 (Insert Activity)*.

The response variable of our experiment is the *interaction category* used to perform the required change of the process model. Therefore, for each step the variable can have one of the following values: *picture-based*, *gesture-based*, or *menu-based interaction*.

As instrumentation of the experiment, an Apple iPad and the multi-touch drawing application Doodle Buddy [25] is used. For each step, this application displays an image to the participant. On the image, a process model is shown serving as starting point for the model change to be conducted in the respective step. Furthermore, the aforementioned textual explanation is displayed in this context. Finger movements of participants are captured through lines overlaying the image (cf. Figure 1). Additionally, the gesture is captured through a video camera. The latter records the screen of the device as well as the hands of the participant and their movements. Furthermore, participants are asked to *think aloud* about what they are doing and what they are thinking about [26]. This information is used for classifying and interpreting results afterwards (cf. Section 3.2). Furthermore, the supervisor of the experiment stays in the same room as the participant, motivates him or her to *think aloud*, and provides assistance in case of emerging questions.

3.2 Analysis and Results

26 participants attended the experiment. Table 2 summarizes background information on them. It is worth mentioning that the number of participants being experienced with multi-touch devices (e.g., smartphones) and participants not using multi-touch devices is almost equal.

Table 2. Participants' Background

	Gender	**Profession**	**Handedness**	**Multi-Touch Experience**
26 Participants	19 Male 7 Female	17 Students 9 Res. Assist.	25 Right 1 Left	14 Yes 12 No

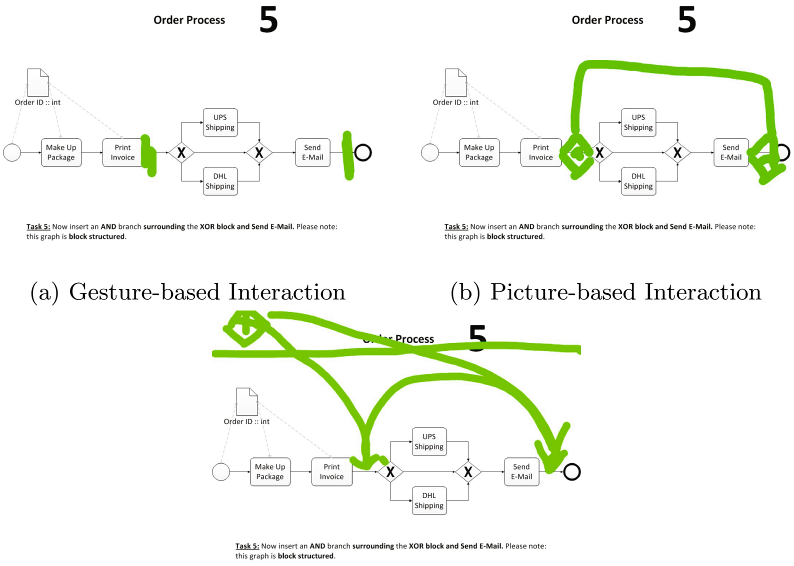
We classify the interactions a participant applies in the context of a concrete modeling step into three *interaction categories*. Note that this classification is accomplished by two persons and is therefore a subjective measurement. As basis of this measurement, we choose the drawing overlaying the image of the respective modeling step as well as the video capturing the actual movement of the participant's hand.

The first category groups *gesture-based interactions*. This kind of interaction uses simple movements on the multi-touch screen changing the process model (as physical gestures described in Section 2.1). Figure 1a exemplarily shows a gesture-based interaction applied to insert an XOR block surrounding a particular process fragment in a process model. In this scenario, the participant moves two fingers simultaneously up and down in order to insert the surrounding block.

The second category comprises *picture-based interactions*, i.e., all interactions drawing a realistic representation of the final result expected from the application of the respective modeling function (as symbolic gestures, cf. Section 2.1). Figure 1b shows an example of this category.

The third category captures all interactions presuming the presence of a menu bar or context menu. Figure 1c exemplarily shows the result of a participant who is drawing a toolbar at the top of the screen and is dragging & dropping the required process elements on the depicted process model. This category, therefore, covers *menu-based interactions*.

Figure 2 summarizes the results of the experiment and visualizes the distribution of the interaction categories. On the x-axis, each step of the experiment



Task 5: Now insert an **AND** branch **surrounding** the **XOR block and Send E-Mail.** Please note: this graph is **block structured**.

Task 5: Now insert an **AND** branch **surrounding** the **XOR block and Send E-Mail.** Please note: this graph is **block structured**.

Task 5: Now insert an **AND** branch **surrounding** the **XOR block and Send E-Mail.** Please note: this graph is **block structured**.

(a) Gesture-based Interaction

(b) Picture-based Interaction

(c) Menu-based Interaction

Fig. 1. Interaction Categories of the Experiment

is represented through its corresponding function. The y-axis, in turn, shows the number of participants using interactions from a specific category.

Obviously, there is no predominant interaction concept covering all process modeling functions. However, for certain functions one can observe a clear preference for a specific interaction concept. Function *F8 (Extract/Inline Sub-process)*, for example, is applied by 73% of the participants by using gesture-based interaction. For other functions (e.g., *F2 (Insert Surrounding Block)*), however, no dominating interaction can be identified. Hence, multiple interaction concepts should be provided to optimally support different user groups and users in applying respective functions.

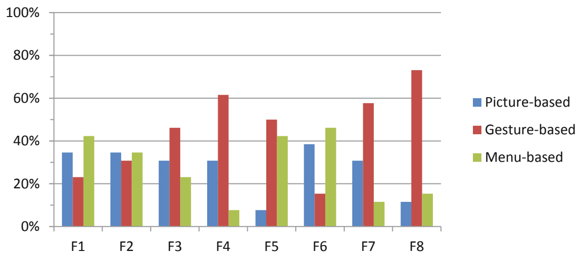
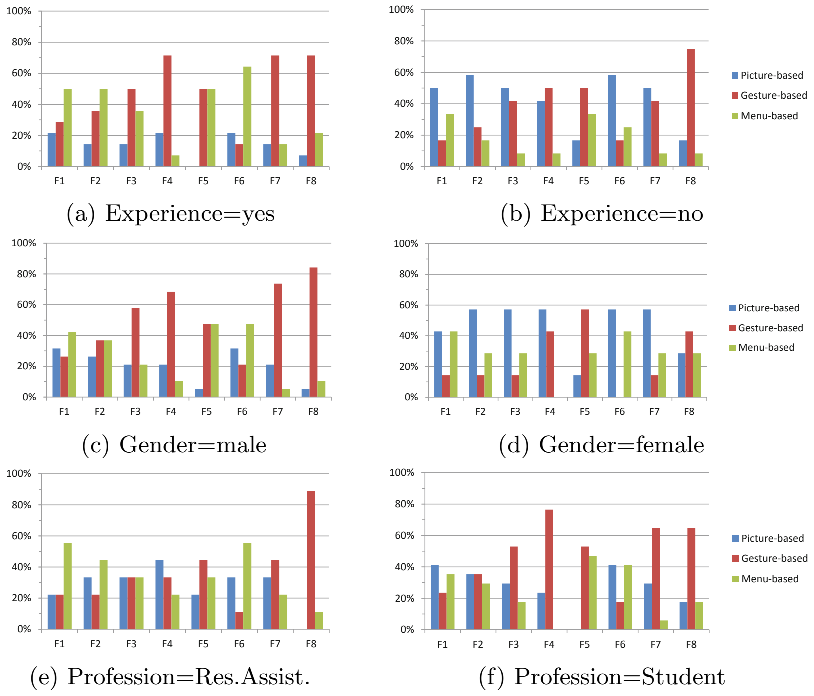


Fig. 2. Distribution of Interaction Categories

When drilling down the results, some insights into differences of the interactions applied by the two groups of participants can be identified. Analyzing the influence of multi-touch experience, for example, shows that experienced participants use gesture-based interactions more often than picture- and menu-based interactions (cf. Figure 3a). By contrast, unexperienced participants more often tend to predominantly use picture-based interactions, but do hardly use menus (cf. Figure 3b). Probably, this can be explained by the fact that unexperienced participants are unaware of common interaction concepts as provided by multi-touch devices (cf. Section 2.2).

Comparing the results of the two gender groups (cf. Figures 3c+3d), female participants predominantly use picture-based interactions. Moreover, none of the female participants uses gesture-based interactions when applying function *F6 (Insert Data Element)*. Male participants, in turn, show a clear preference for gesture-based interactions.

Finally, comparing the results of the professions, participants with low experience on process modeling (i.e., students) slightly prefer gesture-based interactions (cf. Figure 3f). Opposed to this, participants having modeling experience (i.e., research assistants) do not clearly prefer a specific interaction except for function F8 (cf. Figure 3e). Overall, here no clear distinction can be made.



(a) Experience=yes

(b) Experience=no

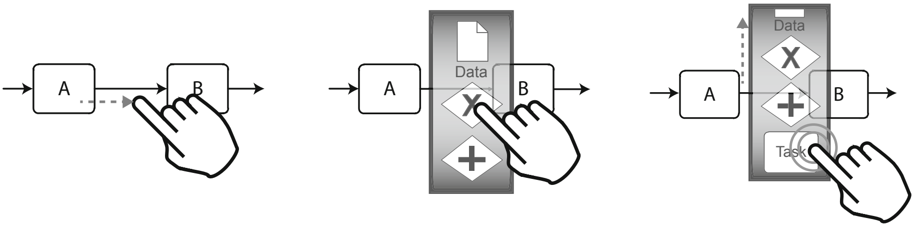
(c) Gender=male

(d) Gender=female

(e) Profession=Res.Assist.

(f) Profession=Student

Fig. 3. Experiment Results



(a) Wipe to Right (b) Scroll the Menu (c) Tap on Element

Fig. 4. Gesture for Inserting New Elements

Note that the results of our experiment are only to a certain degree representative due to the rather low number of participants and the chosen experiment setting. Nevertheless, it can be concluded that process modeling experience does not influence the way how processes are modeled on multi-touch devices. By contrast, being experienced with multi-touch devices affects the way of modeling processes on them. This can be explained by the non-availability of standardized gestures on multi-touch devices (cf. Section 2). Therefore, when developing a multi-touch application, its target group should be taken into account as well; e.g., it might be relevant to know whether members of the target group use their own devices or multi-touch devices are handed out to them.

In the following, we focus on end-users being experienced with multi-touch devices, and propose a core gesture set enabling intuitive process modeling.

4 A Core Gesture Set for Process Modeling

Taking into account the results of our experiment, we propose a core gesture set for modeling and adapting processes. In this context, we have to cope with the existing trade-off between common interaction concepts provided by multi-touch devices and the results of our experiment. Our goal is to provide an appropriate gesture set suited for modeling and adapting processes on multi-touch devices.

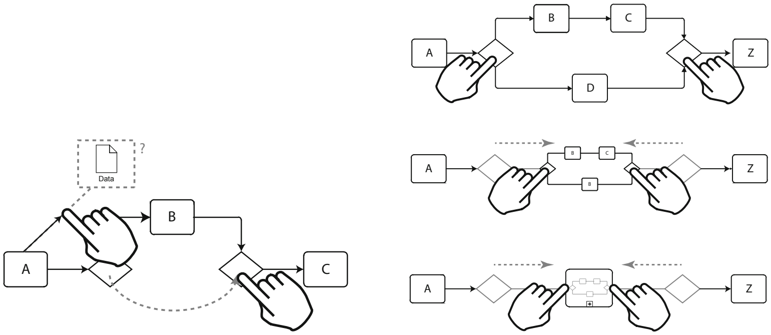
4.1 Core Gesture Set

Our experiment shows that users prefer a menu-based interaction when adding new elements to a process model (cf. Figure 2). Therefore our core gesture set includes a slider menu enabling the insertion of activities, data elements, and surrounding control-flow blocks (cf. functions F1, F2, F6). To trigger the slider menu, users wipe with their finger from an existing process element right to insert the new element at the respective position (cf. Figure 4a). Following this, the slider menu appears at the position where the user releases his finger from the surface of the multi-touch device. Through up and down movements in the slider menu, the required process element can be chosen. For example, to insert an element, the user taps on the respective icon in the slider menu (cf. Figure 4c);

afterwards the slider menu disappears and the element is inserted. Obviously, when inserting a surrounding control-flow block, a second position needs to be selected for adding the corresponding join gateway. For this purpose, the process model editor shows valid positions in the process model where the joining gateway may be inserted; the user then chooses a position by tapping on it. To provide more sophisticated user support when inserting a control-flow block that surrounds an existing process fragment, abstractions in terms of *process views* are useful, i.e., abstracting the process model to a simpler one only comprising those activities being relevant for the insertion of a join gateway [5].

The ordering of the process elements depicted in the slider menu can be optimized by considering their usage frequency. For example, inserting an activity might be more often required compared to the insertion of a surrounding control-flow block. Therefore, the respective function should be positioned on a "central" position in the slider menu that can be quickly accessed.

Another core gesture allows inserting new edges into a process model (cf. Figure 5a). Since we presume a block-structured modeling approach with fixed layout, in general, only two types of edges need to be "manually" added: data edges and edges connecting split and join gateways to add "empty" branches. For example, the user has to draw a line with his finger from the splitting to the joining gateway in order to insert an "empty" branch between them (i.e., F3).



(a) Insert Edges (b) Extract/Inline Sub-process

Fig. 5. Gesture to Insert Data Edges and Extract/Inline Sub-processes

To insert a data edge, in turn, the user has to draw a line between the activity and the respective data element or vice versa (i.e., F7). The direction of the data edge indicates whether it represents a read or write access of the activity on the data element. When inserting a data edge, the gesture set supports users by suggesting a target element (see the left of Figure 5a). If the target element is the desired one the user can lift his finger and the edge will be automatically completed and inserted. Note that this gesture complies with our experiment results since for both functions (i.e., F3 and F7) the majority of participants prefer gesture-based interactions.

To extract or inline a sub-process (i.e., function F8), a two-handed gesture is introduced. The user pushes the start and end element of the process fragment to be extracted into a sub-process activity (Figure 5b). While pushing the elements towards each other, the movement needs to be animated to give direct feedback to the user. After completing the gesture, the new sub-process model has to be linked to the respective sub-process activity. The resulting sub-process can then be displayed through a double tap gesture. To inline a sub-process, in turn, the user pulls the sides of the activity representing the sub-process apart. When pulling the sides of this activity apart, the sub-process appears and is inlined into the process model again. Note that this gesture also matches the results of our experiment in which a majority of 73% of the participants use gesture-based interaction in the context of function F8.

Though 62% of the participants of the experiment use gesture-based interaction to rename a process element (i.e., free-hand writing on the screen), most mobile applications provide an on-screen keyboard for inserting or modifying text on multi-touch devices. The keyboard is hidden most of the time to save screen space and is displayed only when the user wants to add or modify text. To comply with common interaction (de-facto) standards on multi-touch devices, in our gesture set, function *F4 (Renaming Element)* can be activated through tapping on a process element. Afterwards the keyboard appears and the user may modify the text. After confirming the modified text, the keyboard disappears.

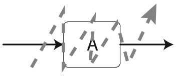


Fig. 6. Delete Element Gesture

Finally, the majority of participants delete process elements using gesture-based interactions. For realizing this function, a gesture crossing out the element to be deleted is suggested (cf. Figure 6). Generally, different variants for realizing this gesture could be envisioned (e.g., drawing two crossing lines or one line from bottom-left to top-right). Regarding our core gesture set, the user draws a connected zigzag-line on the respective element. Generally, this is more accurate regarding recognition compared to the drawing of two separate lines.

5 Summary and Outlook

In this paper, we presented the results of a first experiment we performed to investigate how users model processes on multi-touch devices in a natural and intuitive way. Experiment results indicate that gestures are useful for interacting with process modeling functions, but that for some functions menus are preferred when editing process models. Based on our experiment results as well as common interaction concepts of multi-touch applications a core gesture set has been suggested. This gesture set provides the basis for process modeling tools offering new ways of gesture-based process modeling and providing a high degree of usability. Note that it is no good idea to migrate existing modeling tools, which were designed for desktop computers, straight forward to multi-touch devices since this would affect the usability. The gesture set developed is independent from a concrete device type (e.g., table, tablet or smartphone), but has to be extended with gestures for supportive functions (e.g., undo/redo, copy, paste) in order to use it in a production environment.

As next step a proof-of-concept prototype will be developed. Using this modeling application, further experiments will be conducted to evaluate usability of the developed core gesture set and to elaborate emerging opportunities for modeling processes on multi-touch devices.

References

1. Frisch, M., Heydekorn, J., Dachselt, R.: Diagram Editing on Interactive Displays Using Multi-touch and Pen Gestures. In: Goel, A.K., Jamnik, M., Narayanan, N.H. (eds.) Diagrams 2010. LNCS, vol. 6170, pp. 182–196. Springer, Heidelberg (2010)
2. Gong, J., Tarasewich, P.: Guidelines for Handheld Mobile Device Interface Design. In: Proc. DSI 2004 Annual Meeting, pp. 3751–3756 (2004)
3. Wang, X., Ghanam, Y., Maurer, F.: From Desktop to Tabletop: Migrating the User Interface of AgilePlanner. In: Forbrig, P., Paternò, F. (eds.) HCSE/TAMODIA 2008. LNCS, vol. 5247, pp. 263–270. Springer, Heidelberg (2008)
4. Rudner, B.: Fortgeschrittene Konzepte der Prozessmodellierung durch den Einsatz von Multi-Touch-Gesten. Bachelor thesis, Ulm University (2011) (in German)
5. Reichert, M., Kolb, J., Bobrik, R., Bauer, T.: Enabling Personalized Visualization of Large Business Processes through Parameterizable Views. In: Proc. 26th Symposium on Applied Computing, SAC 2012, Riva del Garda, Trento, Italy (2012)
6. Bobrik, R., Bauer, T., Reichert, M.: Proviado – Personalized and Configurable Visualizations of Business Processes. In: Bauknecht, K., Pröll, B., Werthner, H. (eds.) EC-Web 2006. LNCS, vol. 4082, pp. 61–71. Springer, Heidelberg (2006)
7. Kolb, J., Reichert, M., Weber, B.: Using Concurrent Task Trees for Stakeholder-centered Modeling and Visualization of Business Processes. In: Oppl, S., Fleischmann, A. (eds.) S-BPM ONE 2012. CCIS, vol. 284, pp. 237–251. Springer, Heidelberg (2012)
8. Wobbrock, J.O., Morris, M.R., Wilson, A.D.: User-defined Gestures for Surface Computing. In: Proc. 27th Int'l Conf. on Human Factors in Computing Systems, CHI 2009, New York, USA, vol. 1083 (2009)
9. Brandl, P., Leitner, J., Seifried, T., Haller, M., Doray, B., To, P.: Occlusion-Aware Menu Design for Digital Tabletops. In: Proc. 27th Int'l Conf. on Human Factors in Computing Systems, CHI EA 2009 (2009)
10. Benko, H., Wilson, A.D., Baudisch, P.: Precise Selection Techniques for Multi-Touch Screens. In: Proc. SIGCHI Conf. on Human Factors in Computing Systems, CHI 2006, pp. 1263–1272 (2006)
11. Wang, F., Ren, X.: Empirical Evaluation for Finger Input Properties in Multi-Touch Interaction. In: Proc. 27th Int'l Conf. on Human Factors in Computing Systems, CHI 2009, pp. 1063–1072 (2009)

12. Yee, W.: Potential Limitations of Multi-touch Gesture Vocabulary: Differentiation, Adoption, Fatigue. In: Jacko, J.A. (ed.) HCI International 2009, Part II. LNCS, vol. 5611, pp. 291–300. Springer, Heidelberg (2009)
13. Gemmell, M.: iPad Multi-Touch (2010), `http://mattgemmell.com/2010/05/09/ipad-multi-touch/` (last visited: February 06, 2012)
14. Hornecker, E., Marshall, P., Dalton, N.S., Rogers, Y.: Collaboration and Interference: Awareness with Mice or Touch Input. In: Proc. 2008 ACM Conf. on Computer Supported Cooperative Work, CSCW 2008, pp. 167–176 (2008)
15. Edelman, J., Grosskopf, A., Weske, M., Leifer, L.: Tangible Business Approach Process Modeling: A New Approach. In: Proc. Int'l. Conf. on Engineering Design, ICED 2009, pp. 489–500 (2009)
16. Elias, J.G., Westerman, W.C., Haggerty, M.M.: Multi-Touch Gesture Dictionary. US Patent & Trademark Office, US7840912 (2007)
17. Weber, B., Reichert, M., Rinderle, S.: Change Patterns and Change Support Features - Enhancing Flexibility in Process-Aware Information Systems. Data & Knowledge Engineering 66(3), 438–466 (2008)
18. Rinderle-Ma, S., Reichert, M., Weber, B.: On the Formal Semantics of Change Patterns in Process-Aware Information Systems. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 279–293. Springer, Heidelberg (2008)
19. Weber, B., Reichert, M., Mendling, J., Reijers, H.A.: Refactoring Large Process Model Repositories. Computers in Industry 62(5), 467–486 (2011)
20. Weber, B., Mutschler, B., Reichert, M.: Investigating the Effort of Using Business Process Management Technology: Results from a Controlled Experiment. Science of Computer Programming 75(5), 292–310 (2010)
21. Recker, J., Dreiling, A.: Does It Matter Which Process Modelling Language We Teach or Use? An Experimental Study on Understanding Process Modelling Languages without Formal Education. In: Proc. 18th Australasian Conference on Information Systems, Toowoomba, Australia, pp. 356–366 (2007)
22. Recker, J., Safrudin, N., Rosemann, M.: How Novices Model Business Processes. In: Hull, R., Mendling, J., Tai, S. (eds.) BPM 2010. LNCS, vol. 6336, pp. 29–44. Springer, Heidelberg (2010)
23. Mutschler, B., Weber, B., Reichert, M.: Workflow Management versus Case Handling: Results from a Controlled Software Experiment. In: Proc. 23rd Annual ACM Symposium on Applied Computing, SAC 2008, Special Track on Coordination Models, Languages and Architectures, pp. 82–89 (2008)
24. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslen, A.: Experimentation in Software Engineering - An Introduction. Kluwer Academic Publishers (2000)
25. Pinger Inc.: Doodle Buddy (last visited: February 06, 2012), `http://itunes.apple.com/de/app/doodle-buddy/id313232441?mt=8`
26. Jaspers, M.W., Steen, T., van den Bos, C., Geenen, M.: The Think Aloud Method: A Guide to User Interface Design. Int'l. Journal of Medical Informatics 73(11), 781–795 (2004)