EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling (2010)
M. Alexa and E. Do (Editors)

Inkus: A Freehand Method of Creating Business Process Models

Nicolas Mangano[1] and Noi Sukaviriya[2]

[1]University of California, Irvine
[2]IBM TJ Watson Research Center, Hawthorne, NY

Abstract

Business process modeling has become an industry wide practice for business transformation consulting. Tools that support business process modeling are designed for experienced users to draw a process with precision and professional appearance. These tools are not conducive to sketching quick business design ideas. In this paper, we explore a non-intrusive business process sketching tool which allows free hand sketches of process ideas and slowly brings the users to the required common business vocabulary. We also explore adding business values that support process design thinking, along with sketching. Our goal is to help unleash creativity in business designers and enrich the design process with values beyond drawing. This paper presents a design concept and an implemented prototype of such a system.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [User Interfaces]: User Interfaces—Interaction styles

1. Introduction

Business process modeling nowadays is not a novelty, but a necessity in business transformation ordeals. The need to bridge the gap between business requirements and IT is ever stronger in a weakening economy where one cannot afford expensive IT projects that do not meet business requirements. While there may not be a conclusive form of requirements that is an end-all be-all solution to fractured business requirements yet, a business process model is one of the forms that provide an explicit view of a required business operational structure. For example, at IBM, business consultants are trained to produce business process models as one of the necessary documents in the course of a business and/or IT engagement. These documents partially serve as business requirement agreements. The emerging of business modeling tools and business process management software proves that business process modeling is more than ever becoming a common practice with increasing number of users. Many business process modeling notations exist such as UML, BPEL, CogNIAM, IDEF0, XPDL, [Bus, COG, IDE, XPD] but one that is becoming a standard is BPMN [Obj]. However, whether it is one notation or another, the burden is on the users who must remember the textual or visual notations and the rules for component connections and the constraints that accompany those notations. The users must learn how to read and write the business notational language of choice. While rigorous notations provide effective and efficient way to thoroughly and clearly document business requirements, the rigor, on the contrary, is not conducive to early stage design thinking when creativity and problem solving are mostly the focus of the moment.

Sketching enables users to approach their design in a natural way. Many designers instinctively begin their process designs by sketching [SSR*07]. We believe sketching enables designers to approach their problem in a multidirectional and creative way. Literature showed that designers of any single notation do not naturally adhere to a strict notation [DH07]. They twist notations to express details that are not accounted for in the original notation. A study of business analysts found similar usage of notations, in which they regularly used freehand sketches and semi-structured diagrams to represent enterprise conceptual models [ATAF*09]. Though business process sketching happens in the domain that may not appear creative, we believe freeform sketch-based tools would allow business designers room to work around predefined notations. The freedom from notations and the ability

DOI: 10.2312/SBM/SBM10/143-150

to write down notes without any modality switching would add to the smooth flow of the design process.

In this paper, we described our efforts that embark on designing and prototyping the business process sketching tool *Inkus*, which delivers a mixed environment for both unstructured and semi-structured diagrams. Inkus does so by allowing the user to sketch their design freely, and after which they can annotate their diagrams using a common vocabulary of the desired business notations. From these annotations, a partial model is created, which is represented back to the user in the form of use cases. Our research is on devising a natural way for business users to freely model a business process while providing them with benefits that tightly couple with the incremental and iterative design process. We will discuss the rationale of the design and the implementation of Inkus in the design section.

2. Business Context

Our research goal is around increasing business values during the design phase and reducing the business-IT gap by producing IT solutions directly from business specifications, in this case the business model. Our colleagues at IBM research had developed a capability that can transform a business process model into an executable business engine that provides services strictly based on the business model specifications [KUM04, KN05].

In this approach, the intention is to have business users define a business process model using a methodology called Artifact-centric modeling [NC03], of which focus is on modeling the life cycles of key business artifacts that are the heart of a business. The methodology leads to a clean way of thinking about business operations without IT interference - by enforcing the thinking only around what happens to these key business artifacts. The use of artifacts as nouns and business tasks as the actions to the well selected nouns naturally shakes the model free of IT terms, as the latter tend to refer to non-business artifacts such as screens, database, etc. The focus on key artifacts also keeps the task level high and avoids tasks so small that may be particular to how a particular organization chose to achieve high-level key business actions. (Also, having too many key artifacts in a process is merely a symptom of lack of focus in the business or the modeling exercise does not yet have the right focus.) This approach of modeling has been successful in business design thinking as well as delivering IT solutions. The concept has been proven in several funded research projects that helped restructure business operations as well as delivered IT solutions. Business users felt natural to converse in the artifact-centric language since it aligned with how they see their business operated [BCK*07]. The artifact-centric modeling is a modeling paradigm with which Inkus slowly brings the users to conform.

As mentioned earlier, the resulting business models are used to create an executable engine that choreographs business logic at run-time according to the process model. A business state machine is derived from the task flow; the information model of business documents is used to create database definitions. Business services are created based on possible outcomes (outputs) of each task along with get and set services for business documents for particular users of each task. Related to this work is also a human interface prototyping tool [SSR*07] that uses a model-driven approach to create low fidelity UI prototypes as well as the final UI solutions.

The artifact-centric modeling concept appears rather sound at this point as it has been proven over the years that it works well, conceptually, with business users. Interestingly though, we have also seen evidence that the traditional tool we currently use for business modeling (IBM Websphere Business Modeler) is too cumbersome for the design process. It is often easier to discuss using a blackboard or draw a simplified version of the model using a less constrained (but not ideal) tool such as Microsoft PowerPoint. Also, though the business users actively participated in defining artifact-centric business models, they are more often than not reluctant to get involved in the modeling activities within the tool. The complexity of certain aspects of the modeling notations, as well as the learning curve required to use the business modeling tool, are often mentioned as the reasons.

Informally, we have interviewed and asked a small number of business analysts at IBM to sketch a business process for a given scenario on paper in any way they wish to express a business design. Surprisingly, they drew mostly rectangular boxes with lines connecting between boxes. We did not observe many variations (decision boxes, circles) though we observed that lines are consistently not drawn very neatly. Treatment for line drawing is very casual, for example, lines do not touch source and destination boxes in most cases, lines overlaps destination boxes in some cases. We also noticed this casualness to be consistent within each user. Boxes are drawn either with two, three or four strokes - again this is also consistent within each user. On one hand, this quick study showed that typical business users may not be visually expressive in coming up with creative visual icons as we anticipated. On the other hand, the exercise showed users' readiness in expressing a business design within their own drawing ability. Not surprisingly, this is our initial argument that business process tools need to free business designers from precision and strict notations, at least earlier on in the design process.

The research work described in this paper is our efforts to break away from the traditional business modeling, precision-driven, notation-heavy tools. We experimented with a sketch-based tool that, we believe based on reasons discussed earlier, is more conducive to business design. We also have embarked on embedding additional business values, use case generation from business models in this case, to help with analysis of a business model. This paper reports the beginning of our efforts.

3. Related Work

Several research projects [TAB99, DLC*99, SGP98] in the past applied electronic sketching for taking notes and making annotations. Not only does sketching provide a natural means for writing, these systems also provide in context interactions where notes and annotations were placed within the working context. Another set of research projects applied sketching in the early design process in various domain such as user interface design [LM95b,LNHL00,CV05,CSVV07], 3D design [ZHH06, IMT07] and PatchWork [vdKWB*98]. Similar to our work, these systems applied sketching to support smooth flow of thinking, reduce modal switching, and releasing users from having to deal with too much precision early in the design process.

Related work closer to our work in business process modeling are a few projects that have experimented with electronic sketching in the diagramming domain. [DW05, DHT00] attempted to create UML models using a sketch based input for rapid design [DW05, DHT00]. User evaluations of these approaches found that the gesture recognition within these tools made the products too difficult to use. Follow up studies in SUMLOW [CGH03] found it was better received by users when they allowed the users to first freely sketch. Users could first sketch their approach on a digital whiteboard, and then the system interpreted the sketch into a diagram. Other tools experimented with the similar approach in different domains and had similarly positive results [PA02, LM95a].

The nature of sketching maximizes several characteristics of the creative designer. Among the behaviors observed in creative designers, they tend to create models that are exploratory and ambiguous in nature [Cro06]. Sketching is a valuable medium for these creative designers because of its support for the imprecise models that they create, allowing ideas to be formalized incrementally [Goe95]. Within the creative design literature, there are several characteristics of creative designers that digital sketching could well support. Sketching is well positioned to support behaviors of shifting focus, rapid exploration, and ambiguous models. Designers shift focus often between different levels of detail and different alternatives [Goe95]. A designer may consider the breadth of a design by working with several low-detailed sketches [DH07], then shift their focus to narrow in on the detail of one specific object. Second, a designer needs the ability to explore multiple approaches with little cost to the designer. The tool should allow the designer to quickly express the vision they have in their mind, otherwise it may interrupt their flow of ideas. Lastly, designers use low detailed models that they can not only create quickly, but avoid committing to any decisions prematurely. In fact, low fidelity models actually encourage designers to experiment with alternative approaches [Goe95].

4. Inkus Design

This section describes the design involved in Inkus. In this section, the user is introduced to the sketching interface that drives the primary Inkus user experience (Section 4.1). Within the sketching interface, the user is shown how to build a formal artifact centric model from a sketch using annotations (Section 4.2). In Section 4.3, we review the notation used to support artifact centric BPMs. We then explain in Section 4.4 how the Eclipse-based interface involved in Inkus provides feed-back in the form of an outline and use cases.

4.1. Overview of Inkus

The overall Inkus interface is composed of two parts, (1) the *Drawing Space*, in which the user creates their design in a freehand manner, and (2) the *Project Management Space*, which manages several sketches and provides additional feedback to the user. Sketching sessions are created and stored in the Project Management Space, and a Drawing Space can be launched for each particular sketching session available in the Project Management Space. The Project Management Space is further explained in Section 4.5 and visible in Figure 3, however in the sections that immediately follow, we will focus on the interaction that happens in the Drawing Space.

The heart of the Inkus experience is a hybrid approach between informal notes and formalized elements. The Inkus environment allows the designer to begin their business process design by sketching a crude solution, and then gently transition the basic sketch to a formal, Artifact-centric BPM notation. Within the Inkus environment, users begin their business process design by simply sketching. Inkus avoids forcing the user to commit premature decisions by providing a freeform medium to experiment with partial ideas and write incomplete thoughts. In allowing the user to express unfinished thoughts, the user can use Inkus as a mental aid by not spending extra effort in memorizing concerns that cannot be expressed in a CAD tool's notation. Once an idea is sketched out, users can use a selection tool to assign a formal vocabulary to elements of a sketch. Once formal elements are defined in the sketch, Inkus provides feedback that gives the user insight into their process that would typically be difficult to discern in a low fidelity sketch. We do this by providing an alternative view of the sketch model and providing the user with use cases.

4.2. Creating a Model in Inkus

The Inkus environment allows the designer to begin their business process design by sketching a crude solution, and

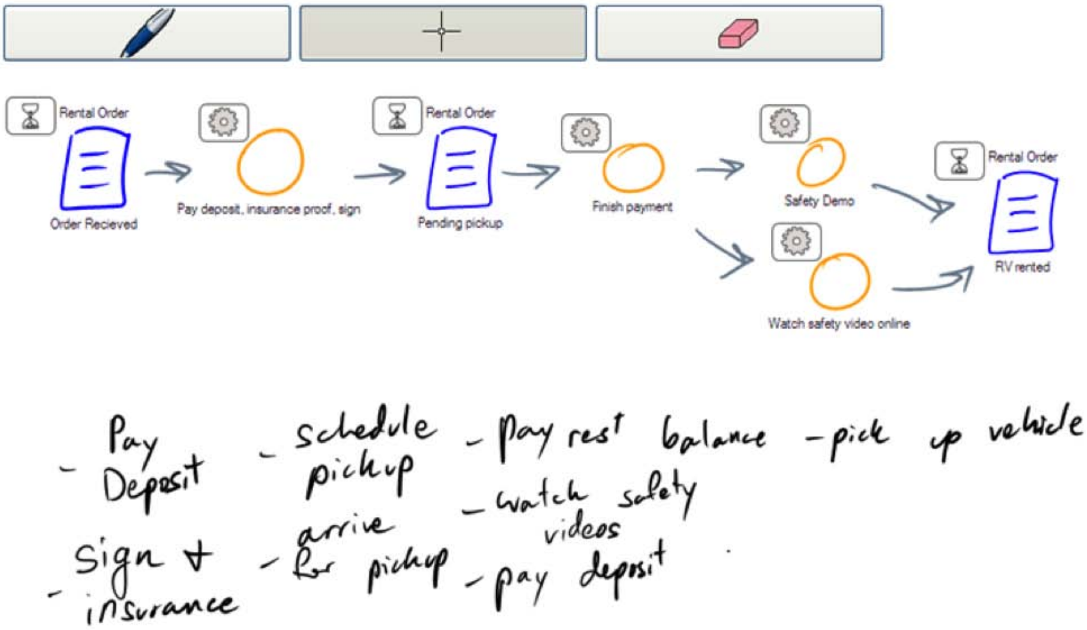


Figure 1: *An example of a sketched processed used to rent an RV annotated within the Inkus drawing space.*

then gently transition the basic sketch to a formal BPM notation. Within the Inkus environment, users begin their business process design by simply sketching. Inkus avoids forcing the user to commit to decisions prematurely by providing a freeform medium to experiment with partial ideas and write incomplete thoughts. In allowing the user to express unfinished thoughts, the user can use Inkus as a mental aid by not committing mental effort to memorizing concerns that cannot be expressed in a CAD tool's notation. Once an idea is sketched out, users can use a selection tool to assign a formal vocabulary to elements of a sketch. Once formal elements are defined in the sketch, Inkus provides feedback that gives the user insight into their process that would typically be difficult to discern in a low fidelity sketch. We do this by providing an alternative view of the sketch model and providing the user with use cases.

Inkus' sketch-first-assign-semantics-later approach introduces a shift in the design process involved in creating a BPM so that it is friendlier to business analysts. The business analysts mentioned in Section 2 reported that they do not use their existing BPM tools when interacting with customers because those tools are difficult to use. Current support tools for business process modeling, such as Websphere Business Modeler, are highly formalized and strict in what they can represent. These tools intentionally constraint the user by guiding them towards an optimized model. The constraints of a formalized tool is helpful for refining an existing model, but they become difficult environments to work in when the designer has not established a model they want to refine. The strict constraints of these tools is a large reason process designers first design on paper, then move to formalized process tools. This behavior has been observed in actual practice [SSR*07]. Our approach blurs the transition from freeform sketches to the formalized structure of specialized tools. In providing support for partial models, the user can simultaneously engage in exploratory behavior at both the abstract and detailed level that is only possible when combining the freeform nature of paper and semantic rich qualities of CAD tools.

Figure 1 illustrates a mock session for creating a business process model for an RV rental company. The lower half of the figure shows several informal notes the designer made to include in the rental process. The top half of the figure shows the RV rental process that was conceptualized from the notes made down below. Unlike sketch recognition tools that replace recognized figures with preexisting beautified shapes [PA02], Inkus permits the informal looking diagram to remain visible and instead annotates them with icons, text, and colors. Once the sketch elements are annotated with an icon, we consider them semi-structured, as they are now classified under the existing notation. Users can assign a notational meaning to each diagram they sketch. They can also declassify diagrams and permit elements to remain ambiguous.

The Drawing Space intentionally separates itself from the Eclipse environment to facilitate a minimalistic interface. The drawing space remains empty with the exception of the three buttons located at the top of Figure 1. The majority of the functionality of the drawing space is hidden to avoid overwhelming the drawing space with clutter. Instead, the user is presented with a large empty space to encourage the primary activity, to draw. The user may begin drawing anywhere on the canvas as soon as the drawing space is loaded. The three available buttons allow the user to toggle between drawing, selecting, and erasing. The drawing tool's functionality is straightforward and is the default setting. The selection tool, in contrast, allows the user to assign semantics to sketch elements in addition to typical selection behavior. Once a set of strokes is selected, a dashed border appears around the selected strokes as well as a box to toggle a pull down menu (a diagram with the pull down menu is visible in Figure 2. The user can move and resize the selected strokes using the dashed border. With the pull down menu, the user can assign an annotation to the selected strokes. Depending on the context of the node, different options will be available.

The Drawing Space exists in its own distinctive window to provide a comfortable drawing environment. The distinctive window of the drawing space allows the process designer to partition the feedback and technical interface of Inkus to a different monitor. When working out the details of a BPM, the focus of the activity is the drawing area. Presenting a simple drawing space allows the designer to directly engage their work without becoming distracted by surrounding interface. When the process designer wants to use the feedback provided by the project management space, the two distinctive windows allows the process designer the option of using a tablet with an external monitor. The tablet allows user to use the pen functionality to sketch in the drawing space, and the external monitor allows the user to view feedback as they annotate the active sketch. Among the feedback based on the annotations, the project management space displays a formal interpretation and use cases, which will be explainer further later in this paper.

4.3. Assigning Notations within Inkus

Once the user has built up enough of a sketch of their process in the drawing space, they may assign semantic meaning to their diagrams. The vocabulary of the available notation is based on an artifact centric approach to modeling business processes. A complete model traces the life cycle of an artifact from creation until it is consumed. In order to define this process, Inkus breaks down sketches into *status*, *task*, and *arrows*. The status component reflects the changing status of an implied artifact. The task component reflects that task taken on the inferred artifact. Lastly, the arrow component defines the flow of the artifact between status and task components. The full detail of this notation is explained in Section 4.4.

The interaction for assigning notations to sketched figures in Inkus is inspired by Mankoff's shape recognition system, Burlap, which presents a popup list of items which the diagram could be recognized as [MHA00]. However, unlike Burlap, Inkus does not attempt to recognize the elements and relies on the user to select an interpretation from a drop down list. This drop down list is visible in Figure 2, which shows a sketched figure after it has been selected using the lasso tool.

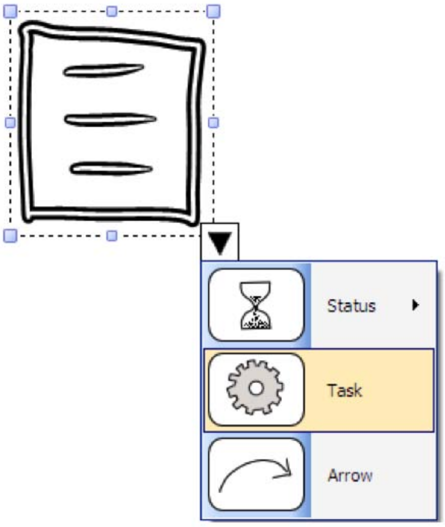


Figure 2: *A cluster of strokes can be assigned a semantic meaning through the pull down menu that appears after a cluster of strokes are selected.*

The lasso tool used to selected sketched elements is ready-at-hand at the top of the drawing space (visible as the second button at the top of Figure 1). In addition to the basic operations available to the lasso tool such as moving and resizing, a small arrow appears in the bottom right corner of all selected cluster of pen strokes. If the user deselects the cluster of strokes, the drop down arrow will disappear. However, if the user clicks the drop down arrow, a set of possible classifications appear. Once a cluster of strokes is categorized into one of the taxonomic values, it is now chunked as a group, and it will be added to the formal model (further explained in the Section 4.4). These classified groups can have textual attributes added to them that appear at the top or bottom of the cluster, depending on the attribute. In order to maintain the pen-based interaction, Inkus loads the Microsoft Tablet PC Input Panel for character recognition to input text.

We have chosen to make the drop down menu available only through the selection tool so that features would appear in context. Context becomes increasingly important for sketched based interfaces. Depending on the size of the screen, the amount of effort needed to move to one side of the screen to the other may become a factor. On the intended target platform, a tablet pc, cross the screen may not require great effort, but moving across the screen to access a feature becomes a distracting task on large touched based displays. Even with small screens, selecting a menu item remains an issue. Selecting an option from the menu bar may cause the user's hand to visually obstruct the screen, temporarily disrupting the user's concentration. To address this problem, a button to call the pull down menu appears to the periphery of selected components. A further motivation to place menus next to selected strokes is so that the contents of the menu change according to context. Going to great lengths to preserve our minimalistic look and feel, we made the sacrifice of making the majority of our functionality available through context menus. Since this risks making much of our functionality difficult to find, we structure our menus so that functionality is self-discoverable. The drawing space attempts to suggest functionality without being obtrusive. The button to load the context menu next appears next to selected items to suggest that additional functionality may be possible, but the button itself does not draw much attention.

4.4. Modeling Notation and Interaction

Depicted in Figure 1 is an example of process model in a hypothetical RV Rental Company. The model depicts the process that a customer must go through to rent an RV. For the sake of simplicity, the example is relatively short in comparison to the process model that an actual RV Rental company would use. In the mentioned figure, several segments of the sketch have been assigned taxonomic value. This is evidenced by the icon in the upper left hand corner of each figure. The icons with the hour glass represent *status*. The icons with the widget icon represent *task*. The symbol informs the user that the component is classified, and clicking the symbol brings up a menu that allows the user to add attributes to the respective *status* or *task* component. Each classified sketch segment was assigned using the lasso and drop down menu described in the previous section. Once sketch segments have been classified, they move as a unit, and have additional attributes that the user can assign.

A *status* component has visible attributes both at the top and bottom of the component. Each *status* component refers to a particular artifact, which is denoted at the top of the *status* component. In the RV Rental example above, each status component is affiliated with the *RV Order* Artifact, as can be seen in the text above the *status* component. Artifacts within Inkus are not explicitly initialized. Instead, they are implicitly declared when the user enters an artifact name into a *status* component. Future status components will recognize the artifact when they are labeled with the same artifact name. Also, if they refer to the same artifact, the diagram of every status component will be the same. This is reflected in Figure 2, each status component has the same paper image. We chose to make the diagrams consistent when the same artifact is used to help visualize the flow of the artifact as the status changes and *task* components perform operations on the artifact. Also, when creating a status component through the lasso pull down menu, the user can select among existing artifacts through a sub-menu. In automatically saving the initial diagram of an artifact and allowing the user to quickly assign that artifact through a submenu, we enable the user to assign several taxonomic values in rapid succession with minimal interruption.

The value at the bottom of a *status* component refers to its status. A component's status represents the progression of states that an artifact may go through in the business process. Within the Inkus model, the first and last components in a process must be a *status* components so that there is a clear start and ending point for a business process. In Figure 2, the initial status of the artifact is "RV Order Received", and the final status of the RV Order is "RV Rented by Customer".

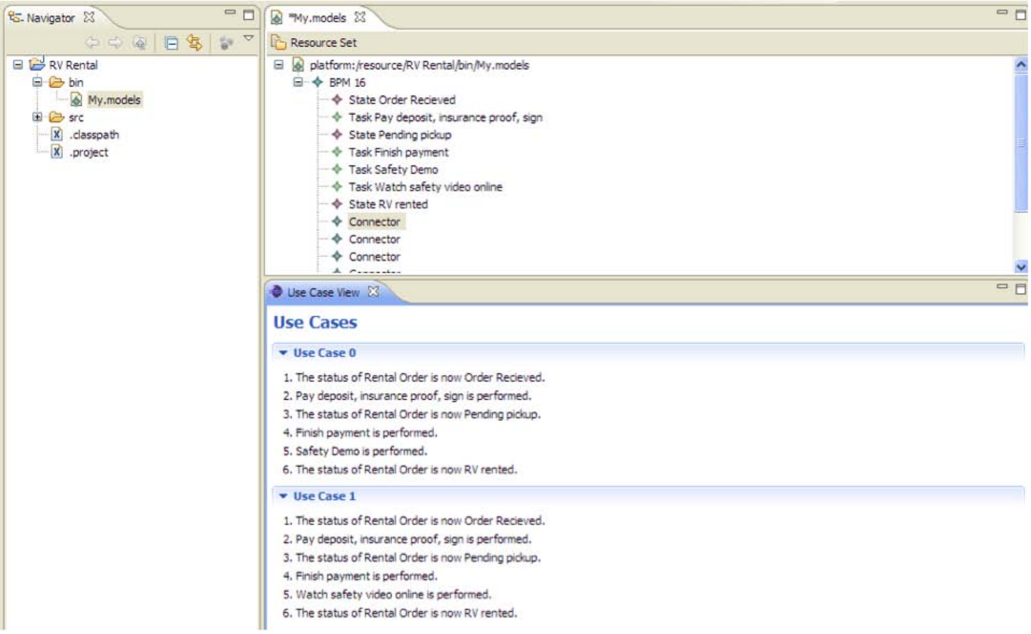
The *task* components act on the artifacts that are fed into them through *status* components. A task component has an activity that it performs on an incoming artifact, which is labeled at the top of the *task* component's figure. The agents that act on the artifact in a given task are denoted below the *task* component's figure.

4.5. Enhancing Awareness with Feedback

As the user assigns meaning to the stroke clusters, Inkus can begin giving feedback to the user through the project management space. The project management space is dependent on the taxonomic assignments that the user gives within the drawing space. Inkus builds an internal representation of the business process model from the components that the user creates. Since sketches can become messy and be difficult to manage over a long period of time, Inkus can use the internal representation for organization. We can now use the internal representation to present the user their model back to them in another perspective. Inkus also can perform heuristics on the internal representation and, in a future version of Inkus, present the user their sketch using a high fidelity BPM notation. The alternative perspectives and heuristics allow the user to comprehend the broader consequences that a change to their model may have.

Several tabs around the project management space move toward the goal of giving sketch feedback. Within the project manage space, there are tabs for a Model Viewer and Use Case View. The Model Viewer presents the outline of the presents an outline of the model as defined by the user in the drawing space. Below the Model Viewer is the tab containing the Use Case View. This view is the basis that gives the user a broader understanding of the consequences of their changes.

The Use Case View provides a literal step-by-step description of the process that an artifact may go through. A distinct use case is created for every path that an artifact may take. It will create a use case for every path that an artifact may follow. In Figure 1, there is a split with two branches in the latter half of the process model. The two branches result in two distinct use cases. Each element of the use case relies on the information entered for each component in the drawing



Use Cases

Use Case 0

1. The status of Rental Order is now Order Recieved.
2. Pay deposit, insurance proof, sign is performed.
3. The status of Rental Order is now Pending pickup.
4. Finish payment is performed.
5. Safety Demo is performed.
6. The status of Rental Order is now RV rented.

Use Case 1

1. The status of Rental Order is now Order Recieved.
2. Pay deposit, insurance proof, sign is performed.
3. The status of Rental Order is now Pending pickup.
4. Finish payment is performed.
5. Watch safety video online is performed.
6. The status of Rental Order is now RV rented.

Figure 3: *Project Management space.*

space. The text of step one of Figure 3 reflects the values of the first *status* component visible in Figure 1. The name of the artifact in the *status* component is "Rental Order", and the status of that component is "Order Received". Combined they form the sentence "The status of the Rental Order is now Order Received".

The Use Case View is available to aid the user in understanding large scale changes to a model. An unfortunate setback of sketches is that they can become messy and difficult manage as they grow in size. Formal structures suffer from the same tendency of becoming difficult to interpret as they grow in size, but sketches are likelier to become increasing difficult to read at a faster pace because of a smaller likelihood of follow gestalt principles. The Use Case View, in addition to the Model Viewer, provide another form of viewing the flow of an artifact within a process model. The Use Case View in particular aids the user in understanding changes done to a complex model. A model with many paths may have drastic changes when a connector is added or shifted from one path to another. Adding another entry point may have the unforeseen consequence of multiplying the possible paths. The Use Case View is updated dynamically with changes made to the model in the drawing space, thus giving the user a sense of cause and effect for their actions.

5. Conclusion

Creating business process using a sketch based medium is an approach that shows promise. It offers a quick way to express diagrams in comparison to CAD tools that process designers are actively using. Inkus allows process designers to express concerns they would normally not be able to in rigid mediums through sketches, easing the cognitive burden of mentally holding these concerns. The simple UI has a shallow learning curve, and is a tool that a business consultant could share with their client. The lightweight UI makes sharing process models easier to share with those not familiar with the domain. In this paper, we have introduced Inkus, a sketch-based application for designing business process models, in which users can gradually migrate from a basic sketch, to a formalized vocabulary.

Many opportunities remain to expand on the Inkus platform. Current features give the user some degree of feedback on their sketches once elements have been classified. We intend to explore other opportunities of providing the user feedback, such as comparing elements of sketches across one another, and applying heuristics. Currently the user can only move from the informal to formal by classifying components, but future versions aim to explore other avenues. We intend to explore importing existing architectures and allow process designers to sketch around them.

Our goal in creating Inkus was to create a fluid environment for rapid exploration. The interaction design of Inkus was created as a response to hassles that plague process of business process designers. Process designers within IBM make use of many tools to brainstorm, collaborate, share, and record business processes. The tools they use are unyielding in what they create, requiring multiple tools to accomplish all components of their tasks. In creating a flexible and rapid medium like Inkus, we hope to reduce the amount of redundancy that comes from using multiple tools, and increase overall efficiency.

6. Acknowledgments

We would like to acknowledge those that helped with the develop of Inkus and all of those that provided continual feedback: Anil Nigam, Avik Sinha, Stacy Hobson, Susan Spraranger, and Terry Heath.

References

[ATAF*09] Anaby-Tavor A., Amid D., Fisher A., Ossher H., Bellamy R., Callery M., Desmond M., Krasikov S., Roth T., Simmonds I., et al.: An Empirical Study of Enterprise Conceptual Modeling. *Conceptual Modeling-ER 2009* (2009), 55–69. 1

[BCK*07] Bhattacharya K., Caswell N., Kumaran S., Nigam A., Wu F.: Artifact-centered operational modeling: Lessons learned from customer engagements. *IBM Sy 46*, 4 (2007). 2

[Bus] Business process execution language for web services. http:http://www.ibm.com/developerworks/library/specification/ws-bpel/. 1

[CGH03] Chen Q., Grundy J., Hosking J.: An e-whiteboard application to support early design-stage sketching of UML diagrams. In *2003 IEEE Symposium on Human Centric Computing Languages and Environments, 2003. Proceedings* (2003), pp. 219–226. 3

[COG] Cogniam. http:/http://www.pna-consulting.nl/. 1

[Cro06] Cross N.: Designerly ways of knowing, 2006. 3

[CSVV07] Coyette A., Schimke S., V J., Vielhauer C.: Trainable sketch recognizer for graphical user interface design, 2007. 3

[CV05] Coyette A., V J.: A sketching tool for designing anyuser, anyplatform, anywhere user interfaces. In *In: Proc. of 10 th IFIP TC 13 Int. Conf. on Human-Computer Interaction INTERACT'2005* (2005), Springer-Verlag, pp. 12–16. 3

[DH07] Dekel U., Herbsleb J. D.: Notation and representation in collaborative object-oriented design: an observational study. In *OOPSLA '07: Proceedings of the 22nd annual ACM SIGPLAN conference on Object-oriented programming systems and applications* (2007), pp. 261–280. 1, 3

[DHT00] Damm C., Hansen K., Thomsen M.: Tool support for cooperative object-oriented design: gesture based modelling on an electronic whiteboard. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2000), ACM, pp. 518–525. 3

[DLC*99] Davis R., Landay J., Chen V., Huang J., Lee R., Li F., Lin J., Morrey C., Schleimer B., Price M., Schilit B.: Note-pals: Lightweight note sharing by the group, for the group. In *SIGCHI Conference on Human Factors in Computing Systems: the CHI Is the Limit* (Pittsburgh, 1999). 3

[DW05] Donaldson A., Williamson A.: Pen-based input of UML activity diagrams for business process modelling. In *Proc HCI 2005 Workshop on Improving and Assessing Pen-based Input Techniques, Edinburgh* (2005). 3

[Goe95] Goel V.: *Sketches of thought*. The MIT Press, 1995. 3

[IDE] Idef0: Function modeling method. http:http://www.idef.com/idef0.html. 1

[IMT07] Igarashi T., Matsuoka S., Tanaka H.: Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses* (New York, NY, USA, 2007), ACM, p. 21. 3

[KN05] Kumaran S., Nandi P.: Adaptive business objects: A new component model for business integration. In *7th ICEIS* (Miami, FL, 2005). 2

[KUM04] Kumaran S.: Model-driven enterprise. In *Global EAI (Enterprise Architecture Integration) Summit* (2004), pp. 166–180. 2

[LM95a] Landay J., Myers B.: Interactive sketching for the early stages of user interface design. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1995), ACM Press/Addison-Wesley Publishing Co., p. 50. 3

[LM95b] Landay J. A., Myers B. A.: Interactive sketching for the early stages of user interface design. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1995), ACM Press/Addison-Wesley Publishing Co., pp. 43–50. 3

[LNHL00] Lin J., Newman M. W., Hong J. I., Landay J. A.: Denim: finding a tighter fit between tools and practice for web site design. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2000), ACM, pp. 510–517. 3

[MHA00] Mankoff J., Hudson S., Abowd G.: Providing integrated toolkit-level support for ambiguity in recognition-based interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2000), ACM, p. 375. 5

[NC03] Nigam A., Caswell. N.: Business artifacts: An approach to operational specification. *IBM Systems Journal 42*, 3 (2003), 428–445. 2

[Obj] Object management group's information on business process modeling notations. http:http://www.bpmn.org. 1

[PA02] Plimmer B., Apperley M.: Computer-aided sketching to capture preliminary design. *Australian Computer Science Communications 24*, 4 (2002), 9–12. 3, 4

[SGP98] Schilit B., Golovchinsky G., Price M.: Beyond paper: supporting active reading with free form digital ink annotations. In *SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, 1998), ACM Press/Addison-Wesley Publishing Co., p. 249Ñ256. 3

[SSR*07] Sukaviriya N., Sinha V., Ramachandra T., Mani S., Stolze M.: User-Centered Design and Business Process Modeling: Cross Road in Rapid Prototyping Tools. *Human-Computer Interaction–INTERACT 2007* (2007), 165–178. 1, 2, 4

[TAB99] Troung K., Abowd G., Brotherton J.: Personalizing the capture of public experiences. In *User Interface Software and Technology Symposium* (1999), pp. 121–130. 3

[vdKWB*98] van de Kant M., Wilson S., Bekker M., Johnson H., Johnson P.: Patchwork: a software tool for early design. In *CHI '98: CHI 98 conference summary on Human factors in computing systems* (New York, NY, USA, 1998), ACM, pp. 221–222. 3

[XPD] Xpdl. http:http://www.wfmc.org/XPDL. 1

[ZHH06] Zeleznik R. C., Herndon K. P., Hughes J. F.: Sketch: an interface for sketching 3d scenes. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses* (New York, NY, USA, 2006), ACM, p. 9. 3