Exploring Usability and Learnability of Mode Inferencing in Pen/Tablet Interfaces

Matei Negulescu, Jaime Ruiz & Edward Lank
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
{mnegules, jruiz, lank}@cs.uwaterloo.com

Technical Report #CS-2009-29

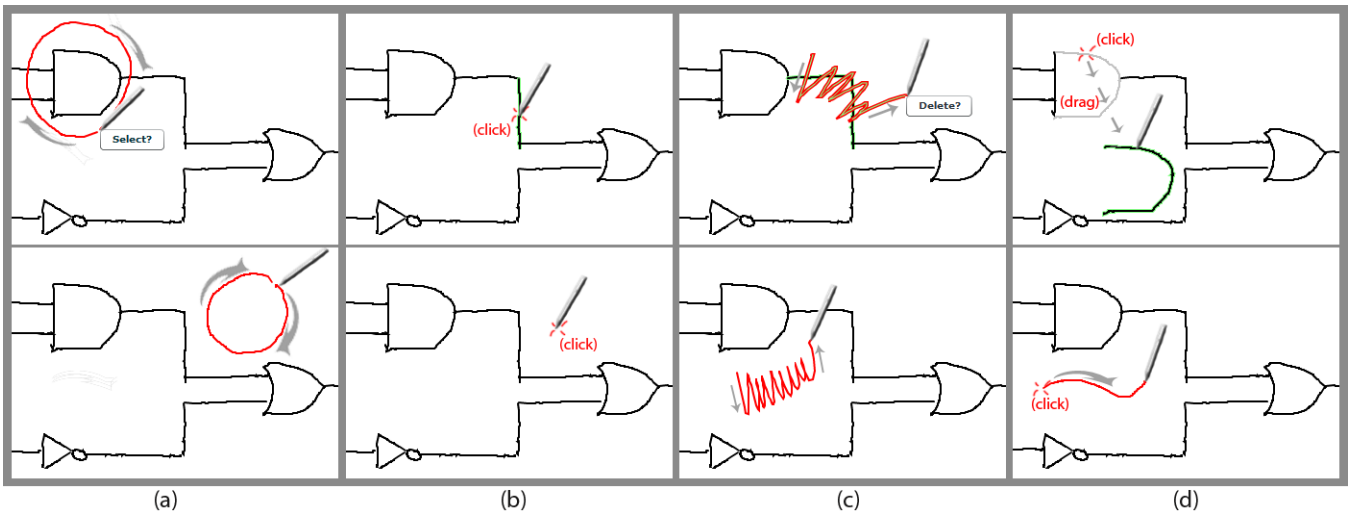

Figure 1: The inferred mode protocol. Panel a. shows smart circle select. When an object is circled, a mediator appears (top), but no mediator appears if the circle encloses nothing. Panel b. shows smart select click. Panel c. shows smart delete (top) and shading (bottom). Panel d. shows translation (top) vs smart drawing (bottom).

ABSTRACT

The inferred mode protocol uses contextual reasoning and local mediators to eliminate the need to access specific modes to perform draw, select, move and delete operations in a sketch interface. In this paper, we describe an observational experiment to understand the learnability – whether the features are discovered independently – and the usability – user preference and frequency of use – of mode inferencing within a tablet-based sketch application. The experiment showed that those participants instructed in the interface features liked the fluid transitions between draw and lasso selection, but did not like click-select and delete inferencing. As well, interaction techniques were not self-revealing: Participants who were not instructed in interaction techniques took longer to learn about inferred mode features and were more negative about the interaction techniques. Together these results inform the future design of pen/tablet interfaces that seek to make use of computational intelligence in support of interaction.

Author Keywords

inferred mode, sketch interface, pen, tablet, stylus

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: Miscellaneous

INTRODUCTION

Pen and paper have supported fluid interaction for brainstorming, problem solving, design, and other creative tasks for millennia. However, despite significant research into hardware and software, the current generation of pen-tablet computers do not support the fluidity of interaction delivered by pen and paper. While part of this failure may still be

Submitted for review to IUI 2010.

due to hardware limitations (screen resolution, the 'feel' of drawing), it is also true that our software systems, particularly our sketch interfaces, have yet to support the effective incorporation of computation into the sketching task.

To allow the stylus to accomplish multiple content-based tasks like inking, erasing and editing, sketch interfaces typically incorporate a set of interface states, or modes of interaction. Although various techniques seeking efficiency in mode switching have been examined [7, 12, 5] it has been shown that there is a measurable cost associated with modes [11]. As the number of interface modes increases, there is a growing need to develop intelligent mode switching techniques that provide low cost access to different interface operations. The inferred mode protocol [13] attempts to minimize mode cost by combining draw, select and delete operations in a single mode using contextual information and local mediator buttons.

Figure 1 depicts the inferred mode protocol's interaction paradigm. To partially eliminate the need to switch modes in the interface, the inferred mode protocol examines the gesture drawn by users and the context of that gesture. For example, as shown in Figure 1a, if a user draws a circle that contains an object, they may wish to draw a circle or to select the object. In this case, the interface supporting inferred mode inks a circle and also displays a local button mediator labeled with "Select?". A user can then select content using lasso selection by pressing the select button or they can leave the ink on the display by ignoring the button. As shown at the bottom of Figure 1a, if no content is inside the circle gesture, there is no ambiguity, and the circle is interpreted as ink.

The goal of this research is to evaluate the inferred mode protocol as a tool for improving interaction in sketch-based interfaces. There are three specific aspects of the inferred mode protocol we explore. First, if the inferred mode protocol is available in an interface, do users use the protocol? Second, do users need to be taught the features of the inferred mode protocol, or is it self-revealing? Finally, what are users subjective evaluations of the protocol and how might the protocol be improved?

We find that, after using the inferred mode protocol for multiple sessions, those users who have received training rate the protocol very highly. As well, users who have extensive experience with or knowledge of tablet computing can discover the features in the protocol and rate it highly. For those users without instruction and with less experience, understanding how the inferred mode protocol works is more challenging, and they find the behavior of the mediator frustrating. Together, these results may inform the design of tablet interfaces that make use of computational intelligence to improve interaction.

RELATED WORK

Research in mode switching on tablet computers can be broadly separated into two categories. First, researchers have actively studied different techniques for setting interface modes. Second, a set of research projects have explored possible techniques for eliminating software modes entirely from the interface.

In the area of mode switching techniques, Li et al. [7] studied five different mode switching techniques on tablet computers. These included using the non-preferred hand to set different interface modes, using the barrel button on the tablet stylus, using pressure, using the eraser end of the stylus, and using a press and hold technique. The goal of the research was to evaluate, from the perspective of speed, errors, and user satisfaction, each of these mode switching techniques to determine whether a "best" technique exists. Of the techniques they examined, they found that use of the non-preferred hand outperformed all other mode switching techniques when controlling two different interface modes, for example a draw mode and a select mode.

Extending this work, Ruiz et al. [11] examined the costs associated with larger sets of modes in an interface. They found that, as the number of interface modes increased, the cost of selecting from a set of n interface modes was described by the Hick-Hyman Law [3]. They also showed that, as expected, as the number of modes increased, so, too, did the number of mode errors made by users. They note that, because of the increasing cost in time and errors of sets of modes, there is benefit in exploring alternatives to mode switching in pen-tablet interfaces.

Various techniques have also explored alternatives to software modes in pen/tablet interfaces. Gesture-based interfaces are a common example of how modes can be partially or completely eliminated [1, 15]. A simple example of this is the scratch-out gesture in sketch programs like Windows Journal where a user can stroke back and forth over an object (i.e. 'scratch-out' an object) and the object is deleted, thus eliminating the need to switch to a dedicated delete mode to erase content. Other techniques have used variants of marking menus [10, 4], where different directional gestures map onto different interface commands.

Overview of the Inferred Mode Protocol

One alternative to modes in software interfaces is the inferred mode protocol, proposed by Saund and Lank [13]. The inferred mode protocol was introduced to solve the problem of switching operation modes through explicit actions. This protocol allows the user to perform draw, select, and delete operations by interpreting new gestures based on the context. We have already described the functioning of lasso selection in the introduction (Figure 1a). In Figure 1b, click selection is depicted. If a user clicks (inks a short stroke) on another stroke, the object is selected. If the user clicks in whitespace, then either an ink dot is placed on the screen, or, if selections exist, everything is de-selected and no dot is placed on the screen. Figure 1c shows delete versus shading. Finally, Figure 1d shows translation behavior. If a user performs pen-down on a selected object and drags, the object is translated. However, a pen down and drag anywhere else on the display results in de-selecting all objects and drawing the gesture.

Figure 2: The decision tree used to reason about user input with the inferred mode protocol.

The inferred mode protocol uses a decision tree to reason about user input. A truncated version of the decision tree is shown in Figure 2. This decision tree is focused specifically on smart circle select and smart click select features. The full set of features of the inferred mode makes use of a more complex decision tree, including reasoning about delete mediators and scribble gestures. In all cases, inferred mode decision tree reasoning begins by examining the state of the system, including whether a mediator is showing or whether selections exist. Depending on where the user clicked (on a selection or somewhere else on the display) and depending on the path the user drew (short or long, closed or open), various actions are performed to support interaction. The decision tree reasoning allows the user to perform inking, editing, and delete operations at any point in time without switching out of a single interface mode.

One characteristic of the inferred mode protocol is that it makes the assumption that all strokes should be classified as ink – preserving pen-and-paper behavior – unless the user explicitly invokes computational support (i.e. by selecting a button mediator) or unless object *state* indicates otherwise (e.g. the user is trying to drag a selected object). In this way, users are free to treat the tablet as a sheet of paper, and the pen-and-paper paradigm is preserved. However, if the users want augmented drawing behaviors such as editing and deletion, then they need to explicitly invoke computational support through the use of button mediators, as shown in Figure 1.

Evaluating Pen/Tablet Interface Techniques

All of the techniques proposed to either simplify mode switching or to eliminate explicit modes from sketch interfaces have been evaluated experimentally. However, many of the evaluations performed have used simplified tasks like pie-cutting [7] or line drawing [11], or have used discrete command invocation evaluation [4] where the user is told to perform a specific command – 'delete', 'cut', 'copy' – and the user performs the action that invokes the command. While these evaluations are useful in telling us about speed and error rate in controlled conditions, they tell us little about the usability of techniques in real-world drawing tasks.

While many of the evaluations have been laboratory studies on restricted interfaces, some researchers have performed studies with higher ecological validity in specific areas of sketch interface research. For example, in their evaluation of GestureBar, Bragdon et al. [1] had participants perform diagram transcription tasks and editing tasks that approximated real world diagram creation. However, the goal of their study was restricted to evaluating a training mechanism for gestures, not the evaluation of a gesture interface, itself. Users were forced to use gestures, and the study compared GestureBar to the use of a crib sheet for learning gestures.

In user interface work, one example of a study with much higher ecological validity is Kurtenbach and Buxton's work on marking menus [6], where they analyzed marking menus in a real-world graphical user interface over a period of time. The evaluation of marking menus was focused on a mouse-based interface for a conversation analysis/editor application, ConEd. Two users used the ConEd system with marking menus for approximately ten hours in total, with use spread over several days. This work, exploring real-world use of a small number of users in detail, was invaluable in validating many of the laboratory findings associated with the speed, accuracy, and learnability of marking menus.

METHODOLOGY

Our goal in this study was to evaluate the usability of one intelligent mode switching technique, the inferred mode protocol of Saund and Lank [13] during realistic sketching. Our study technique is inspired by Kurtenbach and Buxton's ecological evaluation of marking menus [6], though we have a larger number of users in our study. We describe an experiment where users were given pre-specified sketch entry and editing tasks to perform in an interface incorporating the inferred mode protocol, but were not required to use the inferred mode protocol to complete the sketching tasks.

We wished to measure user adoption of the inferred mode protocol, both from the perspective of learnability and user preference. Learnability measures how easy it is to acquire expertise with the interaction technique. User preference measures whether users actually make use of the interface. Over time, in an ecologically valid experiment, user preference can be measured by comparing the frequency of use of inferred mode features with the frequency of use of other options available in the interface. If users use either inferred mode or alternatives more frequently, we can claim that there is a preference for one or the other.

Task

The task was the entry and editing of a set of simple digital logic circuits. Participants were given an initial digital logic

	Instructed/Explicit		Instructed/Implicit		Not Instructed/Explicit		Not Instructed/Implicit	
	P1	P2	P3	P4	P5	P6	P7	P9
Number of Sessions	3	4	3	4	3	5	5	5
Smart Select Click	1.3	0.4	1.7	0.2	0.7	0.4	0.0	0.0
Smart Select Circle	11.7	4.6	11.7	8.5	3.7	0.2	0.0	0.0
Smart Delete	4.3	1.0	5.3	0.5	2.0	0.4	0.0	0.0
Smart Select Click Error	2.7	2.4	4.3	5.0	3.3	1.2	8.2	7.2
Smart Select Circle Error	0.3	1.0	1.3	0.2	0.0	0.0	0.0	0.0
Smart Delete Error	6.3	2.2	3.0	0.2	3.0	0.0	0.0	0.0
Ignored Select Circle	10.7	4.8	5.0	10.5	4.7	8.2	13.8	25.6
Ignored Delete	2.0	2.6	8.0	4.2	2.7	2.4	7.0	6.0
Button Select	0.0	3.0	0.0	0.0	2.0	6.0	9.4	8.6
Button Delete	8.0	5.8	0.7	7.8	4.7	7.2	7.4	9.0

Table 1: Average frequency of operations

circuit and asked to draw it in the interface. They were then asked to modify the digital logic circuit in specific ways, for example by inserting, deleting, or changing gates. While the "work" done by participants was not real, the users were free to perform the tasks in any way they wished within the sketch interface. We gave them no direction on how to perform the tasks, only what tasks they were to perform in the sketch interface.

Experimental Design

We designed a 2X2 observational study that looked at learnability and user preference the inferred mode protocol. To study learnability, we divided our participants into two groups, those who received instruction and those who did not. All participants received approximately 5 minutes of instruction in digital logic circuits. All but one participant had some knowledge of digital logic circuits or of formal logic. The participants in the Instructed group were also give a three minute overview of how the inferred mode protocol worked in the sketch interface they were using, while participants in the Not Instructed group were given no information on the inferred mode protocol. To limit bias, we were careful to show participants in the Instructed group both mechanisms for changing modes in the interface, and did not express any preference for one technique or the other. This design allowed us to determine how easy it was to master the inferred mode protocol. Essentially, was instruction necessary to master the interface technique, or was the technique self-revealing to users?

To study user preference, we wanted to see whether participants made use of the inferred mode protocol over time. To do this, we designed two interface variants. In the first interface variant, pictured here as Figure 3a, an interface was created that contained four modes: a draw mode, select mode, delete mode, and a smart mode. The modes were accessed via radio buttons positioned at the top of the screen. The draw mode performed inking in the interface. Select allowed content to be lassoed or clicked on for selection, and translation operations could be performed on selected content for editing. The delete button allowed users to delete entire strokes by drawing a gesture that intersected strokes that they wished to delete. Finally, the smart mode button

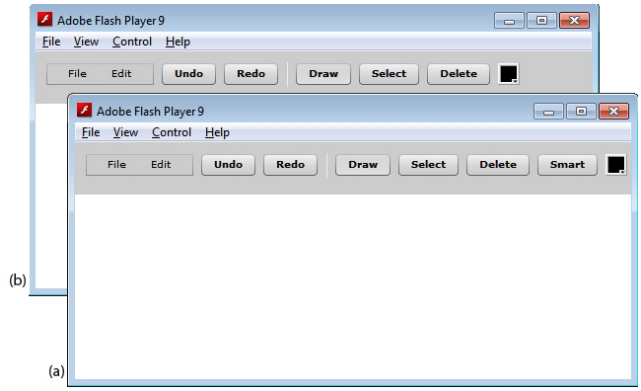

Figure 3: The explicit interface in the foreground (a) and the implicit interface in the background (b)

implemented the inferred mode protocol.

When designing the study, one concern we had is that participants using the four-mode interface might never make use of the "Smart Mode" and, therefore, might never see any of the interface techniques that comprise the inferred mode technique. Participants were free to perform the tasks however, they wished, and we wanted to ensure that at least some of the participants in our study saw the button mediators that invoke computational support. With this in mind, we designed a second interface.

The second interface variant (Figure 3b) had only three modes – draw, select and delete. Select and delete functioned identically to select and delete in the first interface. No computational intelligence was integrated into these modes. However, the Draw mode was designed to implement the inferred mode protocol, essentially mimicking the behavior of the "Smart" mode in interface 1.

In our study design, we label these interface conditions *explicit*, having an explicit smart mode, and *implicit* having the smart mode implicitly included in draw mode. As a result of the two instruction and two interface designs, we had four unique configurations for our study: Instructed/Explicit, Instructed/Implicit, Not Instructed/Explicit, and Not Instructed/

Implicit.

The study was designed as a between subjects, multi-session observational study. Each participant was assigned to one of the instruction/interface configurations, and remained with that instruction/interface configuration throughout their session (i.e. we did not use repeated measures). For each session, participants came to our lab and were given a set of drawing and editing tasks to perform, specifically a set of digital logic circuits to draw and then edit. Each drawing and editing session took approximately 45 minutes, and participants were paid $5 for each session they completed. Each participant completed between three and five sessions, as indicated in Table 1.

Observations

During each session, handwritten notes recorded strategies and behaviors of participants. As well, we used a screen capture application to record participants' actions. The video screen capture allowed us to verify the accuracy of handwritten notes and to quantify the number of times button modes, inferred modes, and mode errors occurred. After the last session, participants were interviewed on their impression of the sketch interface they had used. In a semi-structured interview format, we asked participants which features worked well, which worked poorly, and suggestions for improvements.

Participants

A total of eight participants completed our study, two participants per condition. One participant, P8, dropped out of our study after the first session, so we recruited a ninth participant as replacement. As a result, we label our participants P1 to P9, omitting P8 from further analysis.

RESULTS

The analysis of the results centers around interview and behavioural data gathered throughout the sessions. Table 1 summarizes our data. The first row indicates the number of sessions for each participant. Participants with higher levels of frustration used the application for more sessions (up to five sessions) than participants with lower frustration, as it was our desire to see what was wrong with the inferred mode protocol, and how to improve it. Therefore, participants who were least frustrated – P1, P2, P3, P4, and P5 – have fewer than five sessions. P6, P7, and P9 all have five sessions, an indication of their higher level of frustration.

The remainder of the table contains usage data on the inferred mode protocol and on other mode-switching techniques incorporated into the interface. The second grouping of data indicates the average number of times per session each participant used the inferred mode protocol interactions (labeled as *Smart Select Click*, *Smart Select Circle*, and *Smart Delete*). The third data grouping indicates the number of times participants tried to access inferred mode mediators and the mediators failed to appear or the inferred modes were activated in error. For example, a participant might draw a circular shape around an object that does not pass the threshold for recognition as a select circle operation. The mediator would, therefore, not appear, resulting in a *Smart Select Circle Error*. On the other hand, the participant might inadvertently click on an object on the screen, causing a selection action instead of a short pen stroke. This would results in a *Smart Select Click Error*. The fourth group of results, the *ignored* smart features, indicates those instances where a mediator appeared and users did not interact with the mediator. These are not errors in the inferred mode's behavior. The inferred mode always assumes inking, and if participants want augmented behavior, they must interact with a button mediator. To enable this interaction, if a closed loop is drawn, the button mediator will always appear if that closed loop contains any stroke or portion of a stroke. Finally, the last group, *Button Select* and *Button Delete*, are instances where participants used the explicit modes of operation.

In this section, we explore first the learnability of the inferred mode protocol by examining the differences between the Instructed and the No Instruction groups. Next we look at the usability of the individual features.

Learnability of the Inferred Mode Protocol

When we examine the learnability of the inferred mode, it is easy to see that the Not Instructed group were much less likely to use the inferred mode features. Figure 4 shows the differences between the Instructed and Not Instructed groups. For the Instructed participants, we see that these participants almost always used the inferred mode features, whereas the Not Instructed participants almost always use Button modes.

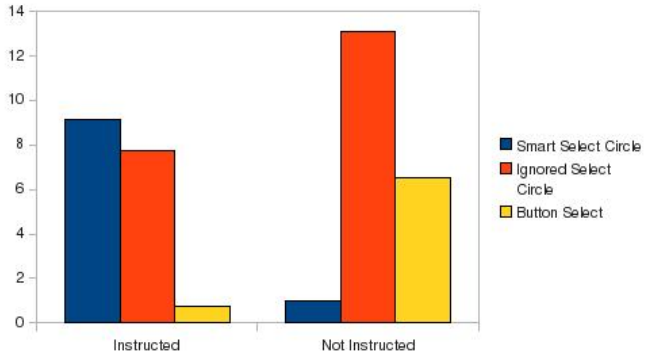

Figure 4: Average Frequency of Select Circle Operations

Within the Not Instructed group, we can further divide participants by interface. Two participants used the Explicit interface, where a special Smart mode implemented the inferred mode features, while two used the Implicit interface, where inferred mode features were always available.

Within the group of Not Instructed participants, we found, somewhat counterintuitively, that the participants with the *Explicit* interface were much more likely to learn and use the inferred mode features. The existence of the "Smart" button prompted the curiosity of the two users in this condition, P5 and P6. One participant, P5, analyzed the Smart mode very carefully during the first session and mastered all of the

smart mode techniques. While this participant did not use the smart features much during the first session, they made use of the features during the second and third sessions. Participant P6 also tried to master the "Smart" features during the first session but failed. Only during the fifth session did this participant begin to understand the features. As we limited our study to five sessions, we were unable to determine whether additional sessions would have increased P6's use of the Smart mode.

An analysis of the responses of these two users during the post-study interview suggests that success in discovery is a function of how knowledgeable a user is with pen/tablet interaction. P5 discovered the protocol in the first session and had a positive reaction to the features as evidenced by the usage data. When asked about the inspiration for the input gestures, he suggested YouTube videos on similar tablet software.

> *I have been watching some Youtube videos related to some HCI applications which says that if you circle it selects or it does something. [...] People always try to draw ellipses or circles to select a part of circuit or part of drawing. [...] Usually when you made a mistake on a paper you just do a scratch [P5]*

In contrast, P6 was not able to uncover the features of the "Smart" mode until the very last session despite several attempts. P6 was frustrated by the inability to master the protocol, and generally did not like the features. Whether P6's opinions would have changed with more extensive use is difficult to determine.

In hindsight, it may have been the case that the label – "Smart" – intrigued participants and encouraged them to master the features of the inferred mode. Smart has positive connotations, and participants may have responded to that. We can point, by analogy, to features such as accelerator keys in interfaces. The label assigned to these features, "accelerators", indicates subliminally that these are good features to master and use in an interface.

However, whether our use of the "Smart" label encouraged participants to explore the interface is somewhat irrelevant. The important observation here is that an explicit mode for extended behavior motivated participants to master the use of those extended behaviors. P5 appreciated the features, and commented favorably on them. P6 seemed less impressed, but was still free to ignore those features and use toggle button modes.

Participants' experience with the implicit interface was very different. The implicit interface seemed, again counterintuitively to us, to *hide* the existence of the inferred mode protocol and to *interfere* with learning its features. As a result, participants' viewed mediators that appear on the canvas while in the otherwise simple "Draw" mode as errors. This seemed to result in a poor understanding of the inferred mode, and participants, as a result, did not use these features. Both participants also had five sessions to master the protocol, and, as we see from Table 1, at no time did they make use of any of the smart features.

It should also be noted that participants in the Not Instructed/ Implicit group had ample opportunity to discover the inferred mode protocol. For example participant P7 saw the Select Circle mediator nearly 14 times per session, or 69 times over five sessions and P9 saw the same mediator 128 times, or almost twice as much. Despite the presence of these mediators, participants never explored their behaviors.

User Preference

Participant use of the protocol uncovered common themes relating to usability and usefulness of the features in the protocol. Select Circle was, by far, the most used of the inferred mode features. Smart Select Click and Smart Delete were less commonly used. We deal with each of the inferred mode features in turn in this section.

Select Circle

Participants P1, P2, P3, P4, and P5 used Smart Select Circle frequently. For all of these participants, it was easily the preferred mode for selecting content on the screen. Essentially, all participants instructed in the technique, and our participant P5 who mastered the technique during the first session, preferred this selection mode, and used it frequently.

Participants who made use of this feature saw it in the same light as a keyboard accelerator, with one participant even comparing it to a shortcut:

> *[Select Circle] is like a shortcut to me. I don't have to go to the menu and then back to the graph. [...] Very good feature to keep. [P4]*

Participants' knowledge of the inferred mode features had a significant effect on their impressions of the button mediators. Participants in the Not Instructed/Implicit condition ignored the mediators as an annoyance.

> *It was more annoying because I didn't know when the boxes came up and they got in the way when I was trying to do stuff afterwards. [...] I never bothered to do it because there were other ways to do it. [...] When it occured I know what it did but I didn't figure out how. [P9]*

In contrast, the participants who were in the Instructed conditions reported relatively little annoyance with button mediators. Participants typically ignored the mediators when they popped up unexpectedly.

> *I didn't even bother [dismissing the mediator] sometimes since I would just continue my work and it would just go away. [P1]*

Select Click

Select Click was one feature that offered some controversy with our participants. Most participants found its use limited, but opinions varied as to how frustrating it really was.

Similar to the reviews of Select Circle, participants in the Not Instructed conditions found this feature frustrating.

One problem with Select Click is inadvertant activations, common when dotting 'i's' or inserting punctuation. For example, P9, who did not use the inferred mode features, noted

> *I didn't know I could select and move things while I was drawing [...] I draw the '4', like the 'L' for the '4' and I'd try to draw the extra lines and I would [select] it. [P9]*

Participants in the Instructed conditions used this feature sparingly, either because they forgot it existed or because the circle select was sufficient for their particular task. One participant in the Instructed/Implicit condition used the feature solely to recover selections.

> *Click to select only works if you've cut something up. [...] Probably the only time I used it was to recover my selection when I accidentally unselected something. [P3]*

Overall, frustration levels for select click depended on whether participants have or gain knowledge of it early enough. Those that did or were instructed in its use found the inadvertent activations negligible and even forgot the feature existed. In contrast, participants who were not aware of the feature's existence found it annoying.

Delete

The Delete gesture in the inferred mode protocol proved to be the most difficult to judge from the experiment. As we can see from Table 1, most participants who wanted to delete used explicit modes. Even those who used the delete gesture generally used it rarely.

In our observations of participants, one thing that we observed is that a delete gesture is rarely the most efficient for participants. Participants, instead, had two different strategies for delete. First, if participants wished to delete a single stroke, most participants found the explicit delete mode both faster and more precise; scribbling has lower precision that a single gesture that intersects the desired stroke. Second, if participants wanted to delete several objects, participants would select the objects they wanted to delete and move them to an unused corner of the drawing. When they had collected all the "trash" in a single location, they would then delete all the strokes at that location using either the delete gesture or the explicit delete mode.

Design Enhancements

During our post-study interviews, many participants suggested enhancements to the system. The most common recommendation was the implementation of a Delete Selection option. It is frequently the case that participants wish to delete a specific region in the diagram. We saw participants create their own Delete Selection by selecting and moving objects from that region to an unused area on the canvas. The Delete Selection option would allow participants to eliminate the translation operation.

We are experimenting with options for Delete Selection. One that appears to hold promise and maintains our default pen-and-paper behavior is a "select-then-cross" operation where users first select an object (using smart circle select). Once the objects are selected, if they then draw a line through the selection, a button mediator prompts them to Delete. If they press the mediator, a delete occurs. Otherwise, because no mediator was pressed, the behavior defaults to pen-and-paper inking, and a line is drawn on the display and content is de-selected.

A second design suggestion involved options for eliminating click-select in the inferred mode. Participant P3 noted that the selection and cutting of curves is a common and often times tedious operation. Users first cut the curves. Then, if they deselect the objects, or if they drop the objects at another location and add to the end of the objects, it can become difficult to know where one stroke ends and the next begins. This participant felt that recovering selections would be simplified if there were a selection undo stack. Because much of the use of the select click feature is restricted to retrieving past selections, an undo stack would eliminate the need for select click.

Finally, there were several suggestions aimed at reasoning about content. For example, participants noted that if they drew a gesture that continued a previous stroke, it would be nice if the system connected the strokes into a single stroke. Participants also wanted to see this feature included in translation, where if digital gates were dropped onto a location, then small connectors attached to the gates could be connected to pre-existing wires that were located near the connectors. Techniques for reasoning specifically about content are outside the scope of this paper, but we mention these selections to highlight the importance of reasoning about content in sketch-based interfaces.

DISCUSSION

The inferred mode protocol is an example of Nielsen's non-command interaction paradigm [8]. The premise of the inferred mode is that the role of the computer in supporting interaction is to "interpret user actions and [to do] what it deems appropriate" [8]. Nielsen claimed that this form of interaction would dominate new user interface paradigms. However, adoption has been slow, and realistic studies of interaction provide evidence for why this is the case.

When evaluating noncommand interaction in pen/tablet interfaces, we see many of the same pitfalls associated with past generations of intelligent interfaces [14, 9]. For example, our users had difficulty developing mental models of how the inferred mode protocol worked. As we noted, the inferred mode protocol analyzes actions and context using a simple decision-tree model. Arguably, decision trees are the simplest form of computational intelligence, yet users still struggle to understand how the system works.

While it may seem that noncommand interfaces are difficult

to understand, it should also be noted that many of our users liked the inferred mode protocol. For example, participants noted that changing from "Select" mode to "Draw" mode is much simpler with the inferred mode, as a user can start drawing at any location on the canvas. As long as users are not performing a pen-down on a previously selected object, the interface inks their current gesture without an explicit mode switch, and selections are managed transparently. The challenge is in how best to communicate to participants the features that are available within intelligent interfaces like the inferred mode interface. With this in mind, we note two features of our experimental conditions that seemed to work well.

Our first observation arises from our Not Instructed group, the participants who were not trained on the inferred mode protocol. For this group, the most effective technique for incorporating intelligence into the interface was as an explicit interface mode. In the case of our explicit interface, even those participants who were not instructed on its functionality were able to determine exactly how the system worked through experimentation. The "Smart" mode gave participants a clue that there was a non-standard aspect to the interaction, and motivated them to understand exactly how the interface worked. In contrast, making computational intelligence a standard part of the interface by embedding it directly into the "Draw" mode caused significant problems for our participants in the Not Instructed group. From this, one of our primary observations is that computational intelligence within interfaces should be viewed as a technique for experienced users, in the same way that accelerator keys are a feature geared toward experienced users. Giving users a basic interface to accomplish work and allowing them to learn about interface enhancements at their own pace seemed to allow for higher user satisfaction and encouraged better learning.

Second, we note that allowing users to ignore certain features is not a solution to interaction design. It was possible for participants in the Not Instructed/Implicit condition to ignore the mediators. If the participants continued drawing, the mediators would vanish without disrupting the display. While these participants were able to accomplish their tasks using the explicit modes, the regular appearance of the mediator distracted participants and resulted in a negative impression of the inferred mode features. In contrast, participants who were in the Instructed/Implicit group did not exhibit the same difficulties with interaction. These participants had an identical interface with identical behavior, but because these participants were instructed about how and why the mediators appeared, they were able to ignore the mediators.

While tutorials or software instruction are both effective ways to communicate knowledge to users, users frequently have little interest in instruction. If there is no way to create an explicit on/off switch for smart interface features (and in some instances it is difficult to do this), it seems wise to make available context-sensitive help that describes for users the rationale behind why the system behaves the way it does. Recent work in the IUI community has explored communicating rationale to users, and noted some benefits to this communication [2]. In our study, we did not use context-sensitive help because we wished to create two very distinct groups of participants – those who knew about inferred features, and those who had to learn on their own. In a real-world implementation of our system, it is likely that all instruction would be through context-sensitive help, so a natural next step is to study how help on demand can be used to train users in smart sketch interface techniques like the inferred mode protocol.

Finally, and most positively, we found that, when well-designed, users valued computational intelligence in sketch interfaces, provided they understood how the computational intelligence worked. All users who were instructed in the inferred mode technique reported liking the technique. Table 1 also indicates that they were frequent users of the technique, using interactions perhaps a dozen times per session. As well, these participants typically avoided using the explicit 'Select' mode in the interface. Only one of the instructed participants, P2, used select mode, but this participant used inferred select more frequently than explicit select – 5 times per session on average versus 3 times per session on average. Delete gestures were less commonly used in the interface, but most participants noted that this was because the delete mode was more reliable and more precise than smart delete.

CONCLUSIONS

In order to better evaluate the inferred mode protocol, we devised an observational experiment that looked at the protocol's learnability without instruction, participants' usage and their recommendations for future improvements. The availability of smart circle selection was shown to be a particularly valuable aspect of the inferred mode protocol for our participants, with participants linking it to features like accelerator keys. Other features of the inferred mode were less frequently used, but were generally liked by participants who understood the features well.

Like all interfaces that include computational intelligence, learnability is a challenge for participants using the inferred mode. Many past studies of sketch interfaces give participants an overview of how the interface works, assuming that participants would step through a tutorial or learn about the interface in some way before use. In our study, we show that, if you make this assumption, with only a short overview of the inferred mode (about three minutes), participants liked the features, mastered the inferred mode easily, and used the inferred mode in sketching tasks spread over multiple sessions. However, in many real world applications, users take a 'walk-up-and-use' approach to software interfaces. With no instruction, we found that users had more difficulty mastering the inferred mode, but that, by making computational intelligence a feature that users specifically access by entering a 'Smart' mode, user satisfaction and learning improved. We recommend that computational intelligence, if at all possible, be a feature that users turn on themselves. This result echoes recent work in techniques like Adaptive User Interfaces [2], where descriptions of rationale and user control

over intelligent actions has been shown to increase the acceptance of adaptation.

Finally, while many techniques to improve mode switching in sketch interfaces for pen/tablet systems have been proposed, these interaction techniques have typically been evaluated in the laboratory. These laboratory studies are useful in comparing techniques, and in highlighting those techniques that have promise in deployed applications. Our study, a multi-session between subjects study of mode inferencing, is designed to extend our understanding of the role, benefits, and liabilities associated with incorporating computational intelligence into sketch interfaces by adopting a more ecological approach to the evaluation of Saund and Lank's inferred mode protocol.

ACKNOWLEDGEMENTS

We would like to thank the participants in our study and colleagues in the Waterloo HCI lab for comments and suggestions. The research described in this paper was funded by the Natural Science and Engineering Research Council of Canada (NSERC) and the Palo Alto Research Center (PARC, Inc.).

REFERENCES

1. A. Bragdon, R. Zeleznik, B. Williamson, T. Miller, and J. J. LaViola, Jr. Gesturebar: improving the approachability of gesture-based interfaces. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 2269–2278. ACM, 2009.

2. A. Bunt, C. Conati, and J. McGrenere. Supporting interface customization using a mixed-initiative approach. In *IUI '07: Proceedings of the 12th international conference on Intelligent user interfaces*, pages 92–101. ACM, 2007.

3. W. Hick. On the rate of gain of information. *Journal of Experimental Psychology*, 4:11 – 36, 1952.

4. K. Hinckley, P. Baudisch, G. Ramos, and F. Guimbretiere. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 451–460, New York, NY, USA, 2005. ACM.

5. K. Hinckley, F. Guimbretiere, P. Baudisch, R. Sarin, M. Agrawala, and E. Cutrell. The springboard: multiple modes in one spring-loaded control. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI 2006*, pages 181–190, 2006.

6. G. Kurtenbach and W. Buxton. User learning and performance with marking menus. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 258–264. ACM, 1994.

7. Y. Li, K. Hinckley, Z. Guan, and J. Landay. Experimental analysis of mode switching techniques in pen-base user interfaces. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI 2005*, pages 461 – 470, 2005.

8. J. Nielsen. Noncommand user interfaces. *Commun. ACM*, 36(4):83–99, 1993.

9. D. A. Norman. *The Design of Everyday Things*. Basic Books, September 2002.

10. I. Oakley and J. Park. Motion marking menus: An eyes-free approach to motion input for handheld devices. *Int. J. Hum.-Comput. Stud.*, 67(6):515–532, 2009.

11. J. Ruiz, A. Bunt, and E. Lank. A model of non-preferred hand mode switching. In *GI '08: Proceedings of graphics interface 2008*, pages 49–56, 2008.

12. J. Ruiz and E. Lank. A study of the scalability of non-preferred hand mode switching. In *Proceedings of International Conference On Multimodal Interfaces, ICMI 2007*, 2007.

13. E. Saund and E. Lank. Stylus input and editing without prior selection of mode. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 213–216. ACM, 2003.

14. L. Tessler. The smalltalk environment. *Byte*, pages 90–147, 1981.

15. R. Zeleznik and T. Miller. Fluid inking: augmenting the medium of free-form inking with gestures. In *GI '06: Proceedings of Graphics Interface 2006*, pages 155–162, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society.