The Usability and Learnability of Pen/Tablet Mode Inferencing

by

Matei Negulescu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The inferred mode protocol uses contextual reasoning and local mediators to eliminate the need to access specic modes to perform draw, select, move and delete operations in a sketch interface. This thesis describe an observational experiment to understand the learnability, user preference and frequency of use of mode inferencing in a sketch appli- cation. Novel methodology is presented to study both quantitative and long term qualitative facets of mode inferencing. The experiment demonstrated that participants instructed in the interface features enjoyed fluid transitions between modes. 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. Over multiple sketching sessions, as users develop expertise with the system, they combine inferred mode techniques to speed interaction, and frequently make use of scratch space on the display to retrain themselves and to tune their behaviors. Lastly, post-task interviews outline impediments to discoverability and how performance is affected by negative perceptions around computational intelligence. The results of this work inform the design of sketch interface techniques that incorporate noncommand features.

Acknowledgements

I would like to thank the HCI lab for helping make this happen. Thank you Ben, Adam, Mike, Fil. And a special thank you to the best supervisor, Prof. Lank, and mentor, Jaime.

Dedication

This is dedicated to my parents, my brother, and love of my life, Helena.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Pen and paper have supported fluid interaction for brainstorming, problem solving, design, and other creative tasks for a large part of history. Moreover, sketches themselves form the basis of creative problem solving by allowing people to generate new and alternative ideas. The benefits of sketching on pen and paper can be attributed to the inherent speed of production, and the ability to maintain ambiguity and flexibility of interpretation [16, 35]. Indeed, even the act of drawing a simple sketch is highly valuable in both problem solving and expression as it forces the individual to synthesize relevant information while ignoring irrelevant data [48].

Considering these benefits, it is no surprise that past research has looked into replicating the advantages in the digital domain. For instance, the Electronic Coktail Napkin provides similar support for ambiguity and flexibility [17], while SILK allows designers to focus on the creative process [29]. 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 due to hardware limitations (screen resolution, the 'feel' of drawing), it is also true that software systems, particularly our sketch interfaces, have yet to support the effective incorporation of computation in the sketching task.

The problem of interaction is clear when considering the modern WIMP paradigm. The standard widgets (e.g. buttons, lists, menus) were built with keyboards and mice in mind

and are ill suited for sketching. As a result, much of the research in Human Computer Interaction in the sketch domain has focused on new methods of interacting with strokes on digital canvases built for tablet PCs, surfaces and mobile devices. Specifically, a significant direction has been attempting to switch operation modes (e.g. drawing, selecting, deleting) with a stylus while moving away from WIMP's toolbars and menus (see the Related Work chapter).

This work focuses on the Inferred Mode Protocol – an interaction technique developed by Saund and Lank [46] that uses context to distinguish between drawing, editing and deleting. Though the inferred mode is a reasonably well known sketch-based interaction technique, it has undergone little scrutiny with regards to real-world usability and learnability. This thesis describes work evaluating the inferred mode protocol using a novel experimental design. This thesis first describes the inferred mode protocol in detail and then go through the research problems, the contributions and outline before delving into the study itself.

1.1 Inferred Mode Protocol

The focus of this research is on the inferred mode protocol [46] which attempts to minimize mode cost by combining draw, select and delete operations in a single mode using contextual information and local mediator buttons. Figure 1.1 depicts the inferred mode protocol's interaction paradigm.

The inferred mode protocol uses a decision tree to reason about user input. A truncated version of the decision tree is shown in Figure 1.2. This decision tree is focused specically 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

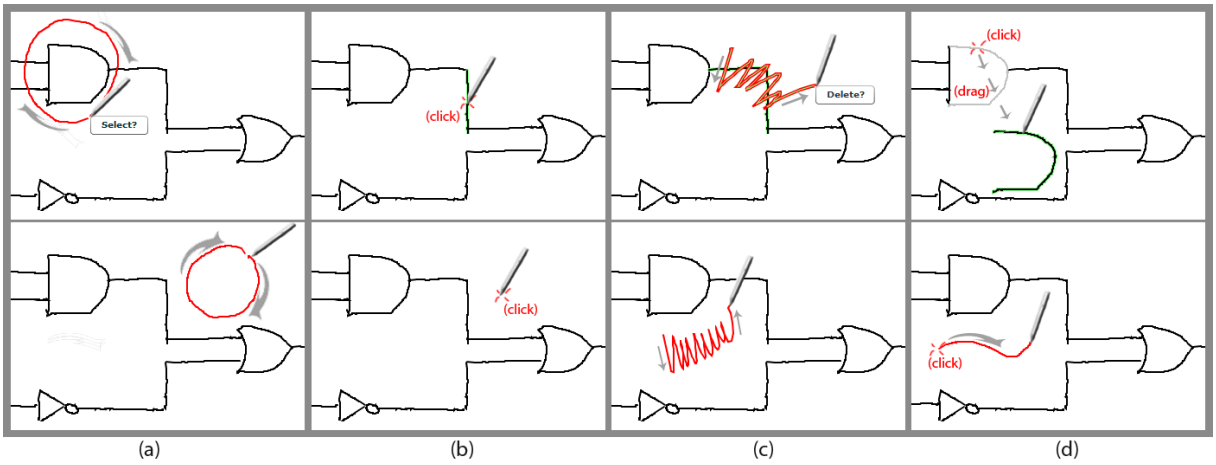

Figure 1.1: The inferred mode protocol. Panel a. shows smart circle select. When an object is circled, a mediator appears (top), but no mediator appears if the circle encloses nothing. Panel b. shows smart select click. Panel c. shows smart delete (top) and shading (bottom). Panel d. shows translation (top) vs smart drawing (bottom).

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 classifed 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 1.1.

For example, as shown in Figure 1.1a, 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 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 1.1a,

Figure 1.2: The reasoning behind the inferred mode protocol's Select Circle.

if no content is inside the circle gesture, there is no ambiguity, and the circle is interpreted as ink. Similarly, Figure 1.1b depicts click selection. 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 deselected and no dot is placed on the screen. Figure 1.1c shows delete versus shading. Finally, Figure 1.1d 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 deselecting all objects and drawing the gesture.

1.2 Research Problems

In order to frame the motivation of this work, this section presents the two primary sets of research problems.

The first 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 explored. 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?

The second goal of this research is to use the evaluation of the inferred mode as a platform for the study of similar interaction techniques that use computational intelligence to assist the user in their task. Though past researchers have built significant methodology for evaluating usability and learnability of interaction techniques and interfaces (see [37] and [19], respectively, for surveys), computational intelligence adds further complexity to their study. This work hopes to provide useful lessons about how computational intelligence in interfaces should be evaluated as a result of designing a long term study of noncommand interaction.

Certainly, it is common practice to report metrics such as recognition rates, type I and II errors, etc. However, special consideration must also be given to the prior perception of users towards the currently non-standard nature of intelligence in interfaces as well as the

effects it has on users over a prolonged period. For instance, adding recognition technology implies that there is a need to study how failures affect users, how fatigue influences results (if it applies), how feedback of complex internal algorithms is best displayed and how it may modify perceptions. Additionally, what are proper methods of correction and frustration costs of failures? As a result, this work explores methodology to study the long term usability and learnability of the inferred mode protocol with the hope that the lessons learned evaluating it will apply to other implementations of computational intelligence in interaction.

1.3 Contributions

Providing valid answers to the above research questions requires careful experimental design and requires longitudinal study [39] or user behavior. To address this challenge, a 2X2 observational study was designed. We created two interface variants of the inferred mode, one in which the inferred mode was present by default and one in which the inferred mode had to be specifically invoked using a special "smart" mode; and two instruction conditions, one where participants were give a short overview of the inferred mode and one where no instruction was given. Researchers analyzed screen videos of eight participants, two in each group, performing sketching tasks over multiple sessions each (over 30 one-hour sessions were analyzed). The researchers also conducted interviews of participants to capture impressions.

The analysis examines issues of learnability and usability of the inferred mode technique. First, without instruction it is difficult to develop an accurate mental model of the interaction technique, and that participants rapidly become frustrated and ignore interface behaviors that they do not understand. Moreover, the results outline impediments to discoverability such as a failure to attract attention to the features and a disconnect between what users expect and what the system produces as a result of their actions.

However, if participants either understand (through instruction) or develop understanding (through exploration) of the technique, they use the technique liberally during their interaction with the sketch. Given an understanding of the technique, they also spend

significant time exploring ways that the technique can be used to optimize their behavior within interfaces. Participants optimize on a variety of factors, involving speed, perceived operational cost, and use of whitespace. For example, participants use scratch-space on the display to explore tolerances and improve their ability to invoke gestures; participants combined operations such as select and delete to perform group deletion in unused screen space; and participants treated mediators in different ways depending on the perceived cost of the operation (e.g. delete is more costly than select, even with an undo operation present).

Finally, this work highlights some of the design implications of this research. Specifically, it touches on the need for training on interaction techniques, the benefits of scratch space, and enhancements to inferred mode operations. Together, these results inform the design of new inferred mode techniques to support fluid inking and editing in interfaces.

1.4 Outline

The remainder of this thesis is structured as follows. First is a description of related work in the area of mode switching in interfaces. The section delves into the definition of the mode problem and how it relates to tablet interfaces. In focus are the costs of modes in an interface: cognitive demand affecting performance times, and a larger number of errors stemming from being in the wrong mode. A number of alternative mode switching techniques are presented in order to understand the state of the art in interface modes.

Next, the related work section describes past work looking at methodology for evaluation computational intelligence in interfaces. Past evaluations of interfaces are considered based on an ecological validity spectrum which maps well controlled laboratory studies and long term, flexible studies.

The main chapter of thesis presents an observational study that looks at the learnability and usability of the Saund and Lank Inferred Mode Protocol in a more realistic setting. The results involve looking at the use and perceptions of individual features, as well as an in depth look at both initial and extended learnability of the protocol. Qualitative and quantitative results present the importance of explicit instruction over perceptions and

performance. Observations during the study uncovers how discoverability occurs and how behaviour evolves over time to optimize interaction.

Finally the thesis closes with a discussion that looks into how perceptions affect performance with computational intelligence, and lessons learned in terms of how to evaluate noncommand interaction methods. A number of future work possibilities are considered to better understand the inferred mode and take the lessons learned about feedback and recognition accuracy into non-sketching domains.

Chapter 2

Related Work

In order to properly situate this research, this chapter contains a summary of previous research into mode switching and evaluations of interaction techniques that use computational intelligence.

2.1 Cost of Modes in Tablet Interfaces

Despite Larry Tesler's plea to not "mode me in" [47], modern interface designers still struggle with overloading actions onto input devices. For instance, in a table PC drawing application a stylus must alternatively generate new ink strokes, select existing strokes, or delete content. There are two problems with modes. The first is that they may introduce errors into user interaction. The second is that there is a cognitive cost to modes that increases in proportion to the number of modes available in an interface [45]. This cognitive cost translates directly to increased time performing tasks in the interface.

This section describes the two penalties for modes in interfaces – time and errors – as well as important related work that has studied the two. Finally, the rest of this section is a brief look at two classes of techniques that improve efficiency in mode selection. The first class consists of techniques which provide improved access to modes by efficiently positioning options or providing a separate command space for activating them. The

second class looks at noncommand methods of interaction where the user fluidly gestures and the system uses computational intelligence to interpret which strokes are commands and which are inking strokes.

Mode Errors

Mode errors result from the user either forgetting to select a mode prior to inputting strokes or simply forgetting which operational mode the system is in. For instance, Yang et al looked at fluid sketch-based interaction techniques that naturally switch between gestures and inking [30]. To switch between gestures and inking, Yang et al considered a number of popular techniques: holding down a barrel button, using pressure to distinguish mode, holding the stylus motionless on the canvas, and setting the mode with the non-preferred hand. As a percentage of total strokes, switching modes by holding a barrel button and by using pressure both produced error rates above 3%; holding a button motionless to switch modes led to error rates above 5%; and using the non-preferred hand to switch modes produced only 1%. Moreover, when erring in setting the mode, the user finds themselves activating miscellaneous commands or drawing spurious strokes on the canvas. As a user regularly performs a large number of strokes on the canvas in a session, even a seemingly small error rate of 3-5% produces numerous occurences of mode errors. Moreover, the price of such mode errors in sketch-based interaction can be high as the user must not only get into the right mode, but also likely spend additional time repairing the results of the erroneous input.

Increased Cognitive Cost

In addition to the cost of repairing a mode error, there is also a cognitive cost that increases with the number of operational states in an interface. Past work by Ruiz et al built a temporal model for switching modes with the non-preferred hand [45], the relatively error free and efficient mode switching technique. Ruiz et al formulate the task requiring prior selection of mode with the non-preferred hand as a decision problem broken down into three time intervals: the time until mode switch occurs (i.e. perception and planning

time), the time interval between the activation of the mode and the first activation of the pen, and finally the time spent gesturing with the pen. As the latter intervals (the interval between mode switch and gesture start, and the gesture time) are independent of the number of modes in the interface, only the cognitive load time measuring perception and planning are considered. The authors show empirically that the cognitive load associated with finding and selecting a mode from a list of options is dependent on the number of available choices (i.e. modes) and is modeled well by the Hick-Hyman Law [23, 7]. The Hick-Hyman Law describes participants' response to a set of choices by modeling the time as a logarithmic function of the number of equally probable choices. As a result, Ruiz et al show that the cognitive load associated with selecting a mode with the non-preferred hand grows according to a logarithmic function of the number of modes available in the interface, everything else being equal. Considering Yang et al's previous study which found mode switching through the non-preferred hand as the least error prone and most efficient technique when compared to techniques that use hardware buttons alternative command spaces (e.g. pressure space)[30], it is likely that this function of cognitive load may be a lower bound for similar mode selection techniques.

2.1.1 Alternative Mode Switching Techniques

Among the first efforts to improve mode selection from a list of items was research that considers item positioning. Placing the modes as pie slices centered around the cursor's current position was shown to be more efficient than a traditional list-based menu [6].

Marking menus further iterated on the concept of localized pie slices by allowing the user to either use radial menus, or perform a quick gesture mark in the direction of the desired item. The user can perform a shortcut by simply stroking in the direction of the desired option (i.e. a stroke whose angle is within the option's pie slice) without waiting for the radial menu to appear [42]. Hierarchical menu structures could be supported by allowing the user to change the direction of their strokes for selecting sub-menu options.

Lastly, Hinkley et al's Scriboli allowed the user to draw one stroke to specify both the target object and the operation through a marking menu [25]. Scriboli supported encircling

items and then appending a pigtail-like gesture ending in a direction in accordance to the appropriate operation mode in the radial marking menu.

As tablet interfaces became more popular, researchers sought to overload the stylus to perform various actions.

One direction to solve the ambiguity in interpretation has been to use a different command space to switch operational modes. For instance, Mynatt et al's Flatland [33], Guimbretière's PostBrainstorm [21], and Landay's SILK distinguish between command input and content using a button press on the stylus[29]. However, these techniques are overly-reliant on specific hardware to interact with the system. Less hardware reliant command spaces proposed to switch modes include speech [8], and stylus hover space [20].

Recent work has evaluated some of the above methods of switching between ink and gesture modes. For instance, Li et al have demonstrated that pressing a button with the non-preferred hand to switch modes is much more efficient than pressing a stylus button or using alternative command spaces (e.g. stylus pressure) [30]. Systems like Springboard [24] and Alias SketchBook [15] use the non-preferred hand to enable a quasimode – a transient mode enabled as long as the non-preferred hand activates the button. Using the non-preferred hand has proven to be a reliable and highly effective method of mode switching.

The above techniques work to improve efficiency in selecting an interpretation of user input. However, these techniques require the selection of mode as a precursor for operations in the system. As a result, though the methods provide efficient access to modes in an interface, the mode problem still plagues such systems. The user must keep track of the current system state or face misinterpreted commands and spurious inputs that are costly to repair. Additionally, regardless of how efficient the access to modes is, there is significant cognitive cost that only grows with the number of mode options available in the interface.

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.

2.1.2 Noncommand Interaction

Nielsen's noncommand interaction paradigm allows the user to perform fluid and "natural" input while the system uses computational intelligence to interpret the user's intent and support them in their task. In the context of mode switching, the user provides sketch input and the system interprets strokes as either inking or command gestures depending on local characteristics. Local characteristics include the spatial context of a stroke ("Where is it drawn?") with respect to other input strokes, its temporal context ("When is it drawn? Before/After other strokes?"), its domain context ("Can we interpret input based on knowledge of the operating domain?"), and geometrical characteristics ("How is it drawn?") that naturally define the input strokes without prior selection of mode. In addition to the Inferred Mode Protocol, which has been previously outlined, therer are other systems of note that allow for fluid sketch input where the intepretation is left to a semi-intelligent system.

Teddy is a system that allows the user to create 3D shapes by interpreting their 2D strokes [26]. The system uses contextual cues (where and when the stroke is created) to distinguish between a number of modeling operations. For instance, a closed stroke in an empty area is used to define an initial spherical 3D shape that can be further refined with modeling operations. In contrast, a closed stroke on an already created 3D object takes the system to an Extrude mode and select the 3D subset of the object defined by the stroke. Subsequent strokes define the direction and shape the selection should be extruded. In contrast, a non closed stroke is just a painting stroke. A scratchout stroke can either be interpeted as a delete operation if over a painting stroke or a smoothing operation if over a pre-selected extrude region. Teddy restricts the user's input to one domain – the creation and definition of a 3D model – to restrict the interpretation of strokes in the system as one of extruding, cutting, or painting operations. It uses this domain as well as a few predetermined command gestures towards allowing the user relatively free-form input whitout explicitly setting the mode through traditional means.

SketchREAD is a sketching system in which the sketch is automatically recognized based on little information regarding the domain of the input [3]. SketchREAD allows the user to input a sketch whose syntactical components are recognized automatically. The

system uses Bayesian Networks to reason about the user's input based on understanding the given domain. The system allows the user to input basic shapes (e.g. triangles, circles, etc) and specify relations between them to create compound elements. Bayesian networks are used to maintain running hypotheses of the interpretation of stroke input. For instance, the authors use circuit diagrams as an example input. A diode can be sketched as a triangle with a line drawn such that it touches one end of the triangle and is parallel with the opposite edge. A Not-gate can be represented as a triangle with a circle at one end. There is no need to switch modes to assist the user in understanding whether an input triangle is to be interpreted as a Not-gate, a diode, or perhaps just a random shape. The system keeps partial hypotheses for understanding the shapes as the user draws them fluidly. The user simply sketches out their diagram and the system interprets input according to relational and geometrical characteristics without any further input from the user.

Flow Selection allows users to sketch and repair curves without prior selection of mode [27]. The main feature of the interaction technique involves selecting and repairing existing strokes on the canvas. The user can hold the stylus relatively near a region that is to be edited. A single point alongside the stroke nearest the epicenter will be selected. As the user maintains the stylus still, the selection expands outwards to points on the stroke around the current selection. The user can then drag the stylus to manipulate the selected points. Points closest on the stroke to the epicenter of the selection will move faster, while those farther away will be translated less for every pixel of stylus movement. Effectively, the user can drag part of the stroke to repair or smooth them similar to manipulating control points. Users efficiently alternate between drawing strokes, selecting them, or manipulating them without explicitly changing modes.

Fluid Inking uses noncommand interaction to sketch and manipulate strokes without prior selection of mode [49] in a manner similar to the Inferred Mode Protocol. The system contains many of the command gestures of the inferred mode: a scratchout gesture for delete, and encircling to delete. Moreover, the authors allow users to copy, cut and paste with a gesture resembling the letter L. The system recognizes potential gestures in the input strokes, but does not act right away. The user can resolve the ambiguity by tapping on the canvas. Alternatively, holding the stylus on the canvas brings up a more traditional radial menu. While the inferred mode uses a mediator button to settle the

ambiguity between interpretations, Fluid Inking uses taps at the end of the gesture to activate a gesture.

2.2 Evaluating Computational Intelligence in Sketch Interfaces

The study of usability and learnability of interaction techniques has been a focus of HCI research since the beginning of the field (see e.g. [37] and [19] for surveys of usability and learnability methodology). Following Nielsen's formalization of usability [40], traditional evaluations are concerned with the learnability of the system – how easy can users accomplish basic tasks without prior experience –, efficiency – how quickly can experienced users perform a task –, memorability – the ease with which users become proficient again after discontinuing use –, errors – the number of errors users make as well as the difficulty of error recovery –, and satisfaction. Though Nielsen looked at learnability in the context of first-time use, Grossman's survey of learnability expands the definition to include extended learnability [19]. The extended learnability of a system corresponds to the change in performance over time.

In order to study the usability and learnability of systems, there have been many methodologies introduced and borrowed from other scientific fields. As a general classification, Nielsen categorizes usability evaluations as those that are tasked with learning about problems with a system (*formative* evaluation), and those that assess how well the application achieves its stated goals, often in comparison with a similar system (*summative* evaluation).

Examples of formative evaluations relevant to this work include Ericsson and Simon's use of the think-aloud protocol in which participants are asked to verbalize their intents, strategies and overall thinking process as they use a system [13]. Rieman asked participants to fill out a diary of their experience with a system in order to identify learning strategies and difficulties [43]. Such a diary study is done in a natural setting which provides some confidence that the findings can be generalized to realistic scenarios. Lastly, the highly popular heuristic evaluation sits on the other end of the spectrum by evaluating an interface

by considering its performance on a set of common heuristics in a relatively controlled setting[38].

In terms of relevant summative evaluation methodologies, past work has considered controlled lab studies aimed at extracting subjective metrics of the system overall, or compared with competing systems. For instance, Elliott et al. asked participants to perform a series of tasks in a lab setting and complete a survey judging the learnability and usability of the system [12]. In general, summative evaluations are lab-based controlled studies which attempt to judge effectiveness, error rates, and learnability through quantitative means.

When it comes to the specific problem of mode switching, all of the techniques proposed to either simplify or to eliminate explicit modes from sketch interfaces have been evaluated experimentally. However, many of the evaluations performed have been quantitative summative evaluations using simplified tasks like pie-cutting [30] or line drawing [45]. Others have used discrete command invocation evaluation [25, 18] where the user is told to perform a specific command – 'delete', 'cut', 'copy', 'select' – and the user performs the action that invokes the command. While laboratory 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 [39].

Some researchers, recognizing the shortcomings of pure laboratory experiments on controlled tasks, have performed studies with higher ecological validity in specific areas of sketch interface research. One example is Bragdon et al.'s [4] GestureBar evaluation, where participants performed diagram transcription and editing tasks that approximate real world diagram creation. Two gesture learning techniques, GestureBar and a crib sheet, were contrasted. However, as the goal of their study was restricted to evaluating a training mechanism for gestures, the participants were given a strict script/tutorial that described the tasks they must do in order to learn a particular set of gestures. The script was realistic, but was still carefully controlled and fully specified. Users had less agency than they would in self-directed tasks.

In user interface work, one example of a study with high ecological validity is Kurtenbach and Buxton's [28] analysis of marking menus in a real-world graphical user interface

over a period of time. Two users used an application with marking menus for approximately ten hours in total, with use spread over several days. Although the evaluation had only two participants, the researchers were able to extract very personal continuous feedback that looked at both short-term usability and effectiveness and long term perceptions of the application through multiple extended sessions. 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.

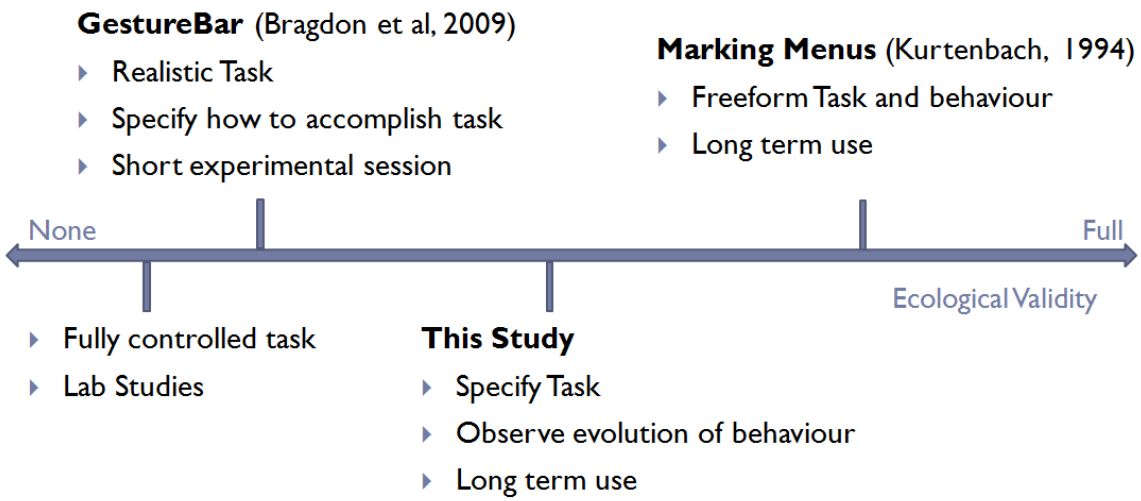

Figure 2.1: Situating the presented study among related evaluations on the ecological validity spectrum.

Evaluating computational intelligence tools like the inferred mode requires special care. For instance, evaluating the types of errors that users make must be extended to account for the precision of the system itself (e.g. the accuracy of a shape recognizer). Moreover, computational intelligence in tablet interfaces often deal with multiple interpretations of input (e.g. overloading a stroke to represent a delete operation, or a draw operation depending on context). This means that an evaluation must take into consideration both how the ambiguity is presented, and the effectiveness of the resolution mechanism provided in the interface. Finally, intelligent user interfaces can provide opportunities for unforseen evolution of user behaviour over extended use. Altogether, these requirements suggest

the need for methodology that measures not only quantitative performance metrics in a relatively controlled environment, but also one that gives sufficient time for users to develop their understanding in a realistic scenario.

To better situate the work among some of the past work mentioned, consider Figure 2.1 which shows evaluation on an ecological validity spectrum. Bragdon et al.'s study of GestureBar represents a class of evaluations that represents a prototypical well controlled laboratory study. In contrast, Kurtenbach's work on Marking Menus falls on the other side of the spectrum, providing the opportunity for a long term, detailed look at the usability and learnability of the technique. The study presented in this thesis hopes to situate itself between these two extremes, providing a specific task but allowing for long term use in order to uncover evolving behaviour.

Chapter 3

Observational Study

The goal in this study was to evaluate the usability of the inferred mode protocol of Saund and Lank [46] during realistic sketching. This study technique is inspired by Kurtenbach and Buxton's ecological evaluation of marking menus [28] and Bragdon et al.'s [4] evaluation of GestureBar. This work describes an experiment where, as in Bragdon's evaluation, participants were given pre-specified sketch entry and editing tasks to perform in an interface incorporating the inferred mode protocol, but, as with Kurtenbach and Buxton's evaluation, participants were not required to use the inferred mode protocol to complete the sketching tasks.

A primary goal is to measure user adoption of the inferred mode protocol, both from the perspective of learnability – how easy it is to acquire expertise with the technique – and user preference – whether users actually make us of the interface. Over time, 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 participants use either inferred mode or alternatives more frequently, one can claim that there is a preference for one or the other. How participants make use of the technique provides us with details on how expertise is acquired.

To accomplish a thorough study of the inferred mode protocol, an observational experiment was designed in which 24 participants reproduced circuit diagrams and then edited

them over three to five sessions. This chapter outlines the experimental design of the study and the data analysis techniques.

3.1 Experiment Design

This section describes the observational experiment, from experimental set-up, tasks, procedure and participant demographics.

3.1.1 Experimental Set-up

In this study, participants were asked to sketch out circuit diagrams and then edit them accordingly. All tasks were done on a simple sketching application written in Adobe Flash (see Figure 3.1 for a screenshot of the application). The application was maximized at a resolution of 1400x1050 on a Toshiba M200 tablet PC running at 1.8GHz. Participants used the stylus to perform all tasks in the experiment.

3.1.2 Experimental Tasks

In order to get a good understanding of the inferred mode protocol, there were two requirements in the choice of experimental tasks. Firstly, there was a need to provide a longer term study that could help understand the inferred mode protocol under both the short and long term scenarios of use. The second requirement was to balance the valuable the internal validity of a controlled laboratory study with the external validity of a more realistic study.

To fulfill these requirements, a task was designed in which participants are required to reproduce a given digital circuit diagram using the tablet PC. Figure 3.2 shows the given circuit to reproduce, as well as a participant's rendition. Upon completion, participants were asked to modify the digital logic circuit in specific ways, for example by inserting, deleting, or changing gates.

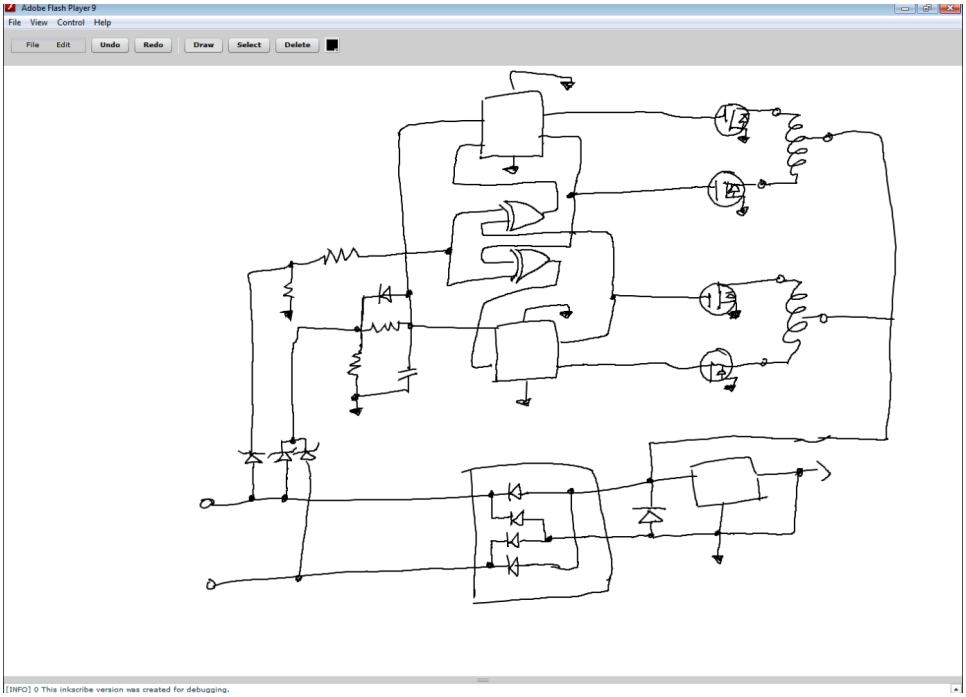

Figure 3.1: A screenshot of the sketch Flash application maximized on a Tablet PC.

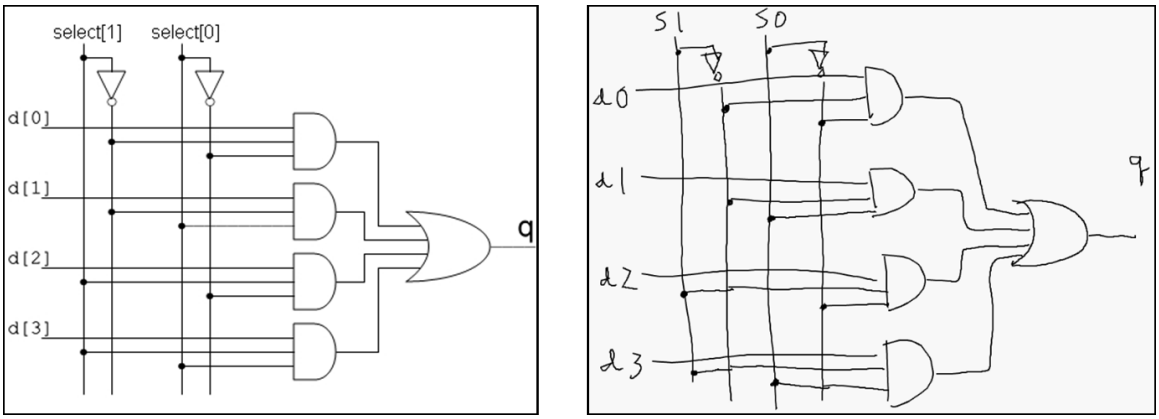

Figure 3.2: The first task given to the users; on the left, the given drawing and on the right is a participants' rendition.

As there was a desire to provide the opportunity for long term use, participants were

asked to come in for three to five sessions composed of sketch-and-edit sequences. Samples of the given sketches and edits are given in Figure 3.2. The selected diagrams roughly increased in complexity. Circuit diagrams were chosen as a terget for this experiment since they provide a realistic scenario for sketching that Computer Science students already perform throughout their undergraduage studies.

While the "work" done by participants was not real, they were free to perform the tasks in any way they wished within the sketch interface. There was no direction on how to perform the tasks; only what tasks they were to perform in the sketch interface. The combination of a long term study as well as the choice of tools helps to maintain some degree of ecological validity

3.1.3 Experimental Procedure

The designed evaluation was a multi-session 2X2 observational study that looked at learnability and user preference of the inferred mode protocol. To study learnability, participants were divided into two groups, those who received instruction and those who did not. All participants received approximately 5 minutes of instruction of digital logic circuits. The participants in the Instructed group were also give a three minute overview of how the inferred mode protocol worked in the sketching interface they were using, while participants in the Not Instructed group were given no information on the inferred mode protocol. To limit bias, researchers were careful to show participants in the Instructed group both the inferred mode protocol and mechanisms for changing modes in the interface without using the inferred mode protocol, and did not express any preference for one technique over the other. This design allowed the researchers to determine how easy it was to master the inferred mode protocol. Was instruction necessary to master the interface technique, or was the technique self-revealing to users? What strategies did users in different conditions take to master the technique?

To study user preference, the resaerchers wanted to see whether participants made use of the inferred mode protocol over time. To do this, two interface variants were designed. The first interface variant, pictured in Figure 3.3a, contained four modes: draw, select, delete and smart. 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 implemented the inferred mode protocol.

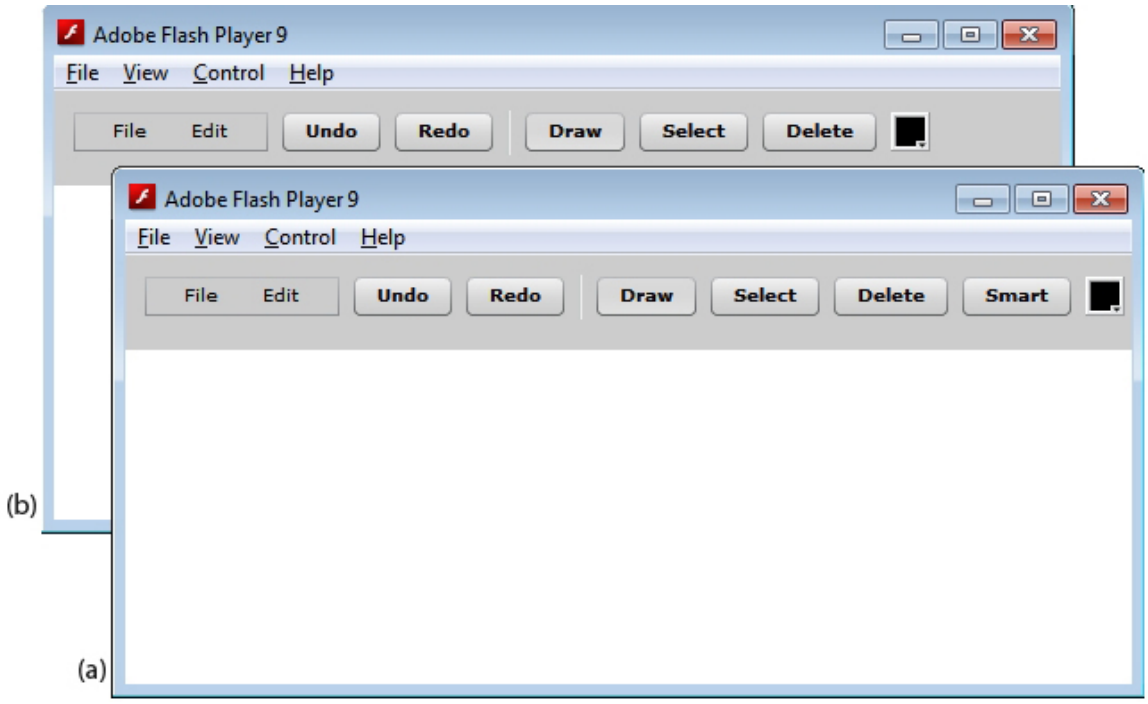

Figure 3.3: the explicit interface in the foreground (a) and the implicit interface in the background (b). note the extra "smart" button in the explicit interface.

When designing the study, one concern was 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 there was a need to ensure that at least some of the participants in the study saw the button mediators that invoke computational support. With this in mind, a second interface was designed.

The second interface variant (Figure 3.3b) 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 the first interface.

In the study design, these interface conditions were labeled *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, there were four unique configurations for the 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. repeated measures was not used).

For each session, participants came to the researchers' 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. There were a total of five sessions consisting of digital logic circuit drawing and editing sequences. Though there were five potential sessions, the researchers stopped the experiment once behaviour stabilized after a minimum of three sessions. Each drawing and editing session took approximately 45 minutes, and participants were paid \$5 for each session they completed. At the end of the experiment, a semi-structured interview was administered in order to understand participants' qualitative views of the tasks, their tools, struggles and successes.

3.1.4 Recruiting and Participants

A total of 24 participants completed the study (eight females), allowing for six participants per condition, assigned randomly. Participants ranged in age from 22 to 31 ($mean = 24.9$, $s.d. = 2.0$). Participants were undergraduate and graduate students generally selected from the Faculties of Mathematics and Engineering at the University of Waterloo. Of the two students who were not studying Mathematics or Engineering, one was part of the Faculty of Arts and another in Science.

All participants were comfortable with computers and general graphics applications such as Microsoft Paint and Adobe Photoshop, but had almost no experience with tablet PC interaction (only P5 had passive knowledge from YouTube videos on tablet PC interaction). In addition, all participants had at least passing experience with digital logic circuits.

The selection process was focused on participants to represent early adopters experienced with computing and graphics applications, though not necessarily expert tablet users. This sample population allows the study of relatively new technically savvy users who have some experience with drawing and editing logic circuits.

3.2 Data and Analysis

This section briefly discusses how qualitative and quantitative data was captured and analyzed.

3.2.1 Data Capture

In designing this experimen , the researchers were interested in not only quantitative data that examines patterns of use, but also qualitative data representing participants' feedback and qualitative changes in behaviour over time. In satisfying the first requirement, all sessions were recorded using software running on the tablet PC. This video data allowed the experimenters to not only examine instances of tool usage, but also identify novel strategies and behaviours of participants. Moreover, this video data was supported by field notes consisting of observations of behaviour and any feedback given throughout the session.

Lastly, the last source of data were transcripts from the exit interviews. These transcripts outlined participants' impressions and suggestions for improvements.

3.2.2 Analysis of Data

In order to understand how participants made use of the application and why, researchers analyzed the data both quantitatively and qualitatively. First, researchers performed a closed coding of the videos consisting of approximately twenty-two hours of video recordings. One researcher analyzed the patterns of use and counted the instances of tool activations, erroneous output, inadvertent input, and ignored mediators when they appeared. Additionally, researchers performed an open coding of the video recordings, the observational field notes, and the diagram artifacts that participants produced. This allowed the experimenters to understand participants' learning behaviour and emergent expert behaviour as they navigated through the tasks. Lastly, an open coding of the exit interview transcripts was performed in order to uncover user perceptions, usability issues, and potential avenues to improve the inferred mode protocol.

3.2.3 Hypotheses

Prior to carrying out the observational study, a series of hypotheses were developed that express the experimenters' expectations for participants' patterns of use and the learnability of the inferred mode protocol.

H1. Instruction Effect Hypothesis: Instruction plays a large role in participants' subjective evaluation of the protocol. Instructed participants will be more positive towards the inferred mode protocol than those uninstructed. The expectation is that appropriate instruction is the primary predictor of future use and subjective evaluation.

H2. Preference of Use Hypothesis: Based on the small scale prior study of the inferred mode protocol, the expectation is that those users who receive instruction will choose to use the inferred mode over the standard toolbar modes, regardless of whether they are using the implicit or explicit interface.

H3. Error Hypothesis: User preference of the inferred mode protocol is not linked with the number of mode errors or inadvertent activations of the protocol. Moreover, the expectation is that instructed participants' positivity is not linked to the number of inadvertent activations of the protocol (i.e. there is no significant difference in the number

of inadvertent activations between instructed and uninstructed participants). In other words, the experimenters expect that instructed participants are more lenient towards erroneous activations than their uninstructed counterparts.

H4: Consistent Use Hypothesis: Participants will make use of their preferred tools regardless of explicitness of the interface. In other words the expectation is that instructed participants to use the inferred mode protocol extensively while staying away from toolbar options (as per H2) while uninstructed participants will prefer the toolbar options, regardless of the presented interface variant (implicit or explicit).

H5: Implicit Discovery Hypothesis: Not Instructed/Implicit users will discover the features of the inferred mode more readily than Not Instructed/Explicit users. The expectation is that being passively exposed to the features of the inferred mode protocol will lead to better results in discovering the features rather than simply avoiding them through the explicit interface.

H6: Early Discoverability Hypothesis: Exploration of the features is mainly done in the first session. Behaviour of use stabilizes quickly after the first session.

These hypotheses help frame experimenter predictions of the usability and short term learnability of the inferred mode protocol.

Chapter 4

Results

This chapter explores use and learnability of the inferred mode protocol. First, an analysis of coded video data is presented. This coded video data captures use of inferred mode protocol. Both the overall use of the intelligent selection features and the use of the specific interaction techniques is examined to determine what factors influence behaviours. Next, an analysis of the learnability of the inferred mode protocol is presented, also based on data from coded video. This work uses Grossman et al's classification of learnability [19] to distinguish initial and extended learnability. The following sections explore how participants learn to use the basic features of the protocol, examine training and retraining behaviours, and describe expert behaviours that evolved during the study.

4.1 Use of the Inferred Mode Protocol

As noted in Chapter 3, three different sources of data were captured when studying the inferred mode protocol: video-taped sessions of use, observational notes of the researcher, and transcripts of post-study exit interviews. In this section an analysis of frequency of use of the different features of the inferred mode protocol is presented based on screen capture of participants sessions. Table 4.1 shows a summary of use, split by condition.

The first row indicates the number of sessions for each participant. Recall that each

	Instructed/Explicit						Instructed/Implicit					
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
Number of Sessions	3	4	5	3	5	4	3	4	4	5	3	4
Smart Select Click	1.3	0.4	0.0	0.0	0.0	0.0	1.7	0.2	20.2	0.0	0.0	0.8
Smart Select Circle	11.7	4.6	0.2	13.7	1.0	0.0	11.7	8.5	5.5	0.0	3.7	12.0
Smart Delete	4.3	1.0	0.8	2.7	13.8	0.2	5.3	0.5	0.0	1.8	3.7	7.8
Error Select Click	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0
Error Select Circle	0.3	1.0	0.0	0.7	0.0	0.0	1.3	0.2	0.5	0.0	0.0	0.8
Error Delete Error	6.3	2.2	0.6	3.3	6.6	1.0	3.0	0.2	0.0	1.0	3.0	4.2
Ignored Select Circle	10.7	4.8	1.2	7.0	11.4	0.0	5.0	10.5	14.8	15.2	5.0	19.0
Ignored Delete	2.0	2.6	0.6	2.0	5.8	0.0	8.0	4.2	5.2	8.4	2.0	2.5
Button Select	0.0	3.0	7.6	2.0	3.0	2.5	0.0	0.0	16.8	5.8	0.3	4.2
Button Delete	8.0	5.8	8.4	4.7	14.2	8.0	0.7	7.8	22.8	13.4	2.7	9.5

	Not Instructed/Explicit						Not Instructed/Implicit					
	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24
Number of Sessions	3	5	5	5	5	5	5	5	5	5	5	5
Smart Select Click	0.7	0.4	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
Smart Select Circle	3.7	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0
Smart Delete	2.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Error Select Click	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Error Select Circle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0
Error Delete Error	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Ignored Select Circle	4.7	8.2	0.0	0.0	0.0	0.0	13.8	25.6	9.0	11.8	18.8	9.0
Ignored Delete	2.7	2.4	0.0	0.0	0.0	0.0	7.0	6.0	9.0	8.2	6.8	3.8
Button Select	2.0	6.0	10.2	0.0	10.0	12.8	9.4	8.6	3.5	11.4	5.6	9.8
Button Delete	4.7	7.2	20.0	18.0	21.5	11.2	7.4	9.0	7.0	6.8	11.6	6.5

Table 4.1: Average frequency (number of uses per session) of operations, rounded to one decimal place. P1-24 represents the labeling of participants.

participant performed multiple sessions using the inferred mode. One factor to note here is the variable number of sessions for different participants. The reason for this variable number of sessions is that participants who seemed more frustrated with the inferred mode, or who had more difficulty with aspects of the techniques, participated in more sessions than did participants who quickly mastered the techniques. During each session, as participants completed their task, observations of the participants indicated how frustrated, difficult, or awkward the interface seemed to be to use. As it was this research's objective to evaluate the inferred mode protocol's learnability and adoption, participants who were least frustrated – the majority of instructed participants and P13 – have fewer than five sessions (an average of 3.9 sessions). In contrast, all but one of the uninstructed participants 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*. On the other hand, the participant might inadvertently click on an object, causing a selection action instead of a short pen stroke (a *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 it. 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. Finally, the last group, *Button Select* and *Button Delete*, are instances where participants used the explicit modes of operation.

Participants' use patterns were analyzed by performing a between-subjects analyis of variance with Instruction and Explicitness as factors. The quantitative analysis that follows is framed in terms of the hypotheses *H1-H4*.

4.1.1 H1: The Instruction Effect Hypothesis

As can be seen in Table 4.1, whether or not participants received instruction played a large role in their choice of tools. Instructed participants generally made use of the inferred mode extensively, while uninstructed users had trouble mastering the features – in the case of Not Instructed/Implicit users – or bypassing the features altogether.

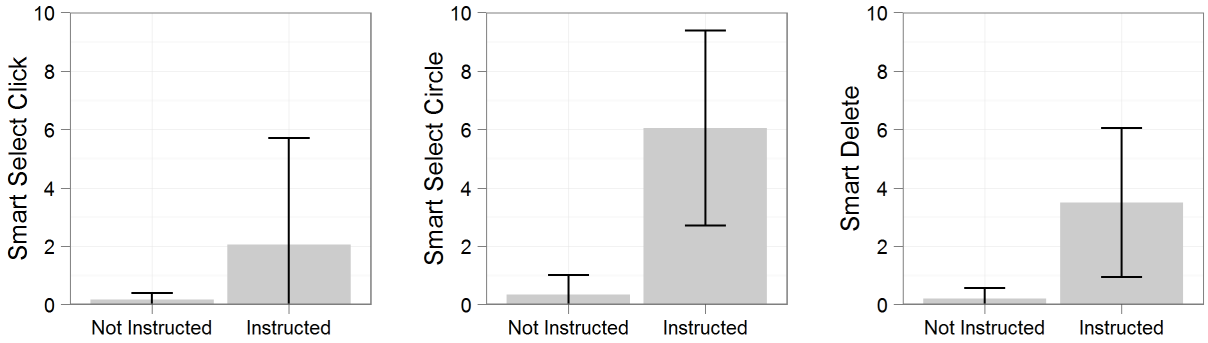

Figure 4.1: Average instances of inferred mode features, by instruction.

Overall, participants behaved as predicted by the *Instruction Effect Hypothesis*. As seen in Figure 4.1, those who received instruction in the use of the inferred mode used Smart Select Circle (mean = 2.05, S.D. = 5.74) and Smart Delete (mean = 3.49, S.D. = 4.02) more than those who did not (mean = 0.18, S.D. = 0.34 and mean = 0.20, S.D. = 0.58), respectively for those uninstructed). An analysis of variance showed a significant effect of Instruction on instances of the inferred mode's select circle ($F_{1,22} = 13.57$, $p < 0.005$) and on delete ($F_{1,22} = 7.89$, $p < 0.05$). Of note is that there was no significant effect of Instruction on instances of select click – a sign that this feature is less important when given good alternatives for selecting content.

As per the hypothesis, whether or not participants received instruction is an accurate predictor for whether participants make use of the inferred mode features. When breaking the instructed participants based on whether they used the implicit or explicit interface, there is no significant difference in the all participants' patterns of use. Though the sample size becomes too small to generalize in this case, it is interesting to see that the inferred

mode protocol is used even when participants are not forcefully exposed to the protocol. Participants appear to make the conscious decision to engage the features even when given the choice to completely bypass any contact with the protocol through traditional alternatives in the interface.

4.1.2 H2: The Preference of Use Hypothesis

Due to instructed participants' continued use of the inferred mode to interact with content, we would expect there to be significantly fewer number of explicit button operations (Button Select and Delete) when compared to participants in the Not Instructed conditions. However, as Figure 4.2 shows, this was not the case. Instructed participants used the Button Select and Button Delete (mean = 3.77, S.D. = 4.76 and mean = 8.83, S.D. = 5.86 respectively) at similar rates as to that of uninstructed participants (mean = 7.44, S.D. = 3.99 for Button Select and mean = 10.91, S.D. = 5.76 for Button Delete).

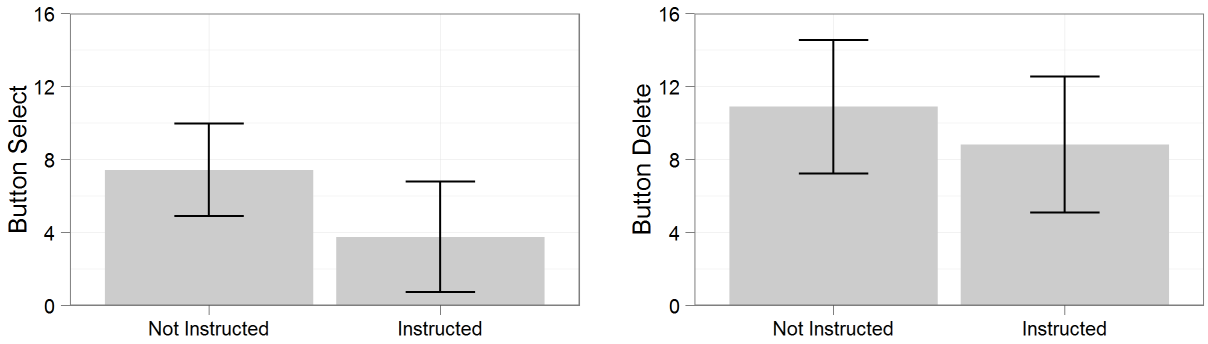

Figure 4.2: Average instances of button activations, by instruction.

An analysis of variance showed no significant effect of Instruction on instances of Button Delete ($F_{1,22} = 0.77$, ns), and merely a weak trend for Instruction and Button Select ($F_{1,22} = 4.20$, $p = 0.053$). Counter to the stated Preference of Use Hypothesis, instructed participants did not use the inferred mode features exclusively, though they did so extensively. However, the inferred mode protocol does not appear to be a replacement for traditional

mode switching. As participants gained expertise with the interface, they attempted to optimize their behaviour and varied their selection of tools to achieve the current task. The section discussing Extended Learnability will touch on specifically which factors participants optimize with experience.

4.1.3 H3: The Error Hypothesis

Based on the theoretical efficiency of the inferred mode protocol, the expectation was that it would be used extensively if participants are given instruction. Moreover, the Error Hypothesis formalized the prediction that users will be more lenient with regards to errors or inadevertent activations of features they find useful. The validated Instruction Effect Hypothesis states that instructed participants use the inferred mode significantly over uninstructed users, though not exclusively. The extensive use and positive qualitative feedback of instructed users falls in contrast to the number of the high number of errors encountered while activating the protocol (false negatives) and a high rate of inadvertent ativations (false positives). In other words, the expectation is that participants recognize the inferred mode's usefulness even in the face of errors.

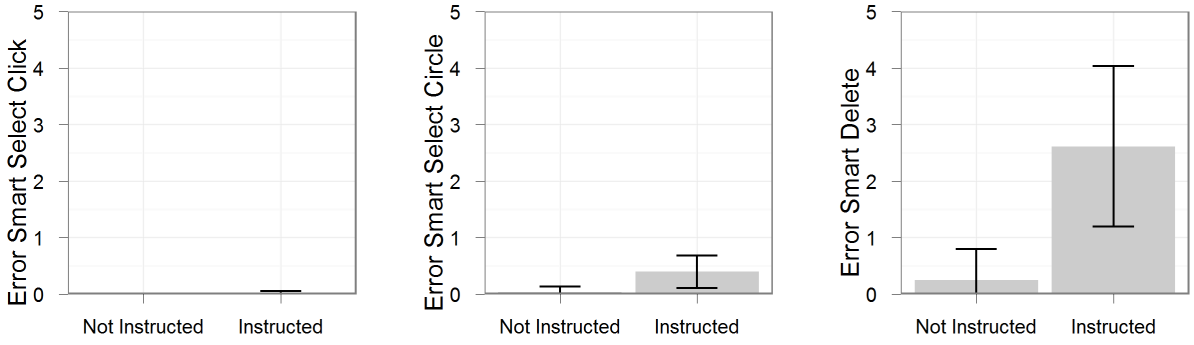

Figure 4.3: Average instances of false positives and false negatives by Instruction.

As expected, with more use of the inferred mode protocol, instructed participants encountered significantly more instances of errors for deleting and selecting content than uninstructed participants ($F_{1,22} = 11.716$, $p < 0.01$ for Error Smart Delete and $F_{1,22} = 6.755$,

$p < 0.05$ for Error Smart Select Circle). As Figure 4.3 shows, instructed participants encountered an average of 0.4 errors for encircling text compared to uninstructed participants' 0.04 instances. The difference is much more obvious when looking at instances of Smart Delete errors: instructed participants encountered 2.6 errors on average, compared to uninstructed participants' 0.25 instances. Of course, this is evidence that uninstructed participants simply did not use the protocol and therefore encountered fewer errors. However, it is important to note that despite the fact that such computational intelligence is more error prone than traditional interaction, users are more lenient if they see value in its use.

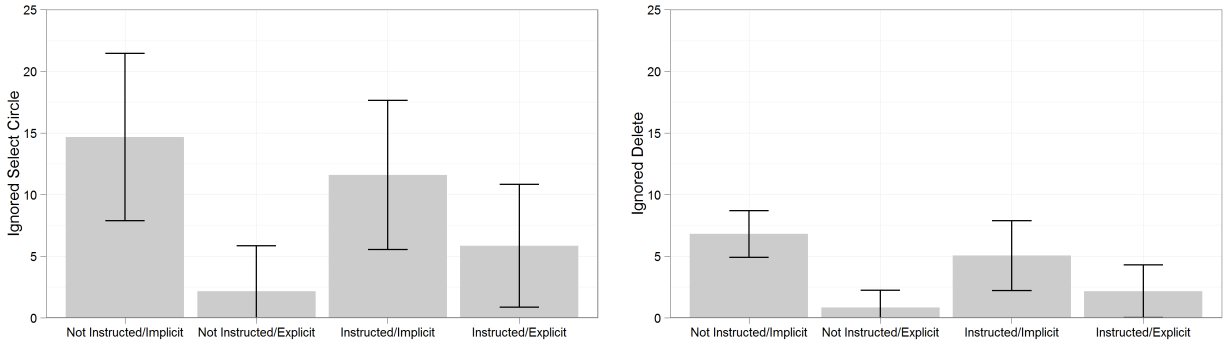

Figure 4.4: Average instances of ignored features by condition.

In addition to erroneous activations of the inferred mode protocol, it is important to note the false positives that arise when enabling such computational intelligence. It was previously noted that qualitative feedback was much more positive for participants who were instructed in the use of the protocol than those who were not. One of the major issues that led to frustration was the frequency of inadvertent activations of the protocol. As Figure 4.4 shows, when breaking participants into their respective conditions, there is no significant difference in number of false activations between users who used the same interface, regardless of instruction. All participants who used the Implicit interface encountered a large number of inadvertent "Select?" and "Delete?" mediators. Despite this similarity, participants who did not receive instruction were highly negative and frustrated about these unexplained occurences. This pattern in the number of instances of inadvertent

activations of inferred features further confirms the Error Hypothesis: participants who received instruction are more lenient towards the inferred mode protocol than uninstructed users. This pattern holds true even as instructed participants encounter significantly *more* false negatives (i.e. mode errors), and a similar number of false positives as uninstructed participants.

4.1.4 H4: The Consistent Use Hypothesis

The expectation was that participants who are instructed in the use of the protocol will not only use it extensively, but also use it continuously and consistently. *H4* formalizes the belief that participants will use the inferred mode protocol regardless of which interface they are presented with.

Overall, the analysis of use supports this consistent use hypothesis. Though it was noted that instructed participants do not solely rely on the inferred mode protocol to perform their task, there is virtually no effect of Explicitness (whether the interface included an explicit Smart mode for the inferred mode, or if it was hidden in the Draw mode) in participants' patterns of use. There was no significant effect of Explicitness in instructed participants' use of the inferred mode features (Figure 4.5). Moreover, though instructed participants still judiciously use the Button Select and Button Delete in the toolbar, their frequency of use is not significantly different when split by Explicitness.

In fact, there was only one effect of Explicitness on any of metric regardless of breakdown: when only considering Not Instructed participants ($n = 12$), there was a significant effect of Explicitness on instances of Ignored Select Circle ($F_{1,10} = 17.32$, $p < 0.005$) and Ignored Delete ($F_{1,10} = 42.15$, $p < 0.001$). This pattern was unsurprising since Not Instructed/Implicit participants had the inferred mode enabled, while the Not Instructed/Explicit users could – and, in fact, did – bypass it by seldom activating the Smart mode. Figure 4.5 shows a summary of features used by participants split by whether they used the Explicit or Implicit interface

The validity of the Instructed Effect hypothesis demonstrated that instructed participants use the inferred mode protocol at a higher rate, though not moving away from

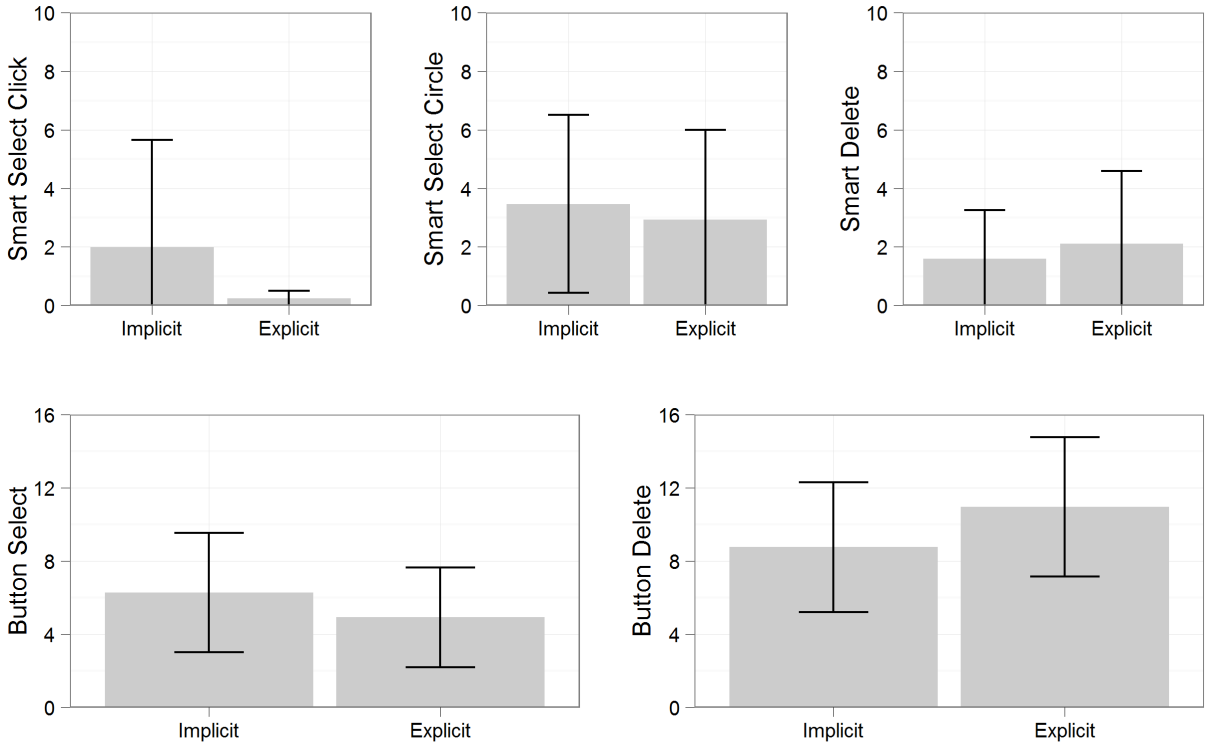

Figure 4.5: Average instances of features used, by explicitness.

traditional button modes (as per the partial invalidity of The Preference of Use hypothesis). Given this, the Consistent Use Hypothesis aimed to predict behaviours of use that are consistent regardless of explicitness in the interfaces. The analysis of use confirms that the way the interface is laid out (either presenting the inferred mode protocol explicitly through a button or having it enabled by default) is not a direct factor in participants' choice of tools. However, though Instruction is a primary predictor of inferred mode us, Explicitness has an important effect on learnability which will be discussed in later sections.

4.2 Usability of the Inferred Mode Protocol

Participant use of the protocol uncovered common themes relating to usability and usefulness of the features in the protocol. These include participants' perceptions of mediators and of the different inferred mode techniques. The qualitative feedback from the users is in line with the quantitative evidence regarding hypotheses H1-H4: participants that receive instruction are more positive than those that do not and make consistent use of the inferred mode features to perform their task, ignoring the large number of false positives and negatives in their experience. This section first presents feedback with regards to the individual features of the inferred mode, and continues with perceptions of the general computational intelligence from the post-experiment interviews.

4.2.1 Individual Features

Having looked at the overall patterns of use in the last section, this subsection now delves deeper into participants' feedback regarding the mediators and the individual features of the inferred mode protocol.

One characteristic of the inferred mode is the frequent presence of mediators on the display. Mediators were perceived very differently by participants in the Instructed and Not Instructed conditions. Instructed participants typically ignored the mediators when they did not want to select or delete content, as exemplified by P1.

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

In contrast, Not Instructed/Implicit did not form a mental model that linked their pen input with mediators appearing on the screen; these participants found the mediators annoying, and found that they interfered with interaction on the display.

> *I have to tap out of the screen. So that was annoying. [..] It gives me this minipanic, for a microsecond; [..] it wasn't something I wanted and I was afraid it would affect my work. [P21]*

The majority of the instructed participants used Smart Select Circle frequently, as shown in Table 4.1. For these participants, it was easily the preferred mode for selecting content on the screen. 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. [P12]*

Select Click was one feature that was controversial with participants. Most participants found its use limited, but opinions varied as to how frustrating it really was. Similar to Select Circle, participants in the Not Instructed conditions found this feature frustrating. The problem with Select Click is inadvertent activations, common when dotting 'i's' or inserting punctuation. For example, P19, who did not use the inferred mode features, noted that it frequently selected things while they were drawing.

Participants in the Instructed conditions used Select Click sparingly, either because they forgot it existed or because the circle select was sufficient for their particular task. However, instructed participants did not consider inadvertent activations very problematic when sketching in the interface. As a result, when asked if they would keep this functionality in the inferred mode, most participants considered it a useful alternative. This attitude is exemplified by P9, who used Select Click extensively (20 instances per session).

> *I guess it is useful having a secondary backup. Sometimes if I'm just selecting a line I can just tap on it; I like that function (Select Click); [...] When it fails I like having the Select button. [P9]*

Lastly, participants were split in their use of the delete gesture. Overall, instructed participants averaged 3.5 instances of the scratchout gesture (as per Table 4.1). However, six of the users (P1, P4, P5, P7, P11, P12) used the gesture extensively at a rate of 6.3 per session, while the remaining six instructed users averaged 0.7 instances. Though participants were reasonably positive about the feature in concept, there was concern regarding its usefulness and the quality of the implementation. Users noted that the delete gesture is rarely the most efficient method for deleting content due to a lack of accuracy.

Sometimes it's hard to control how big of an area to scratch out. [..] If the diagram gets really crowded, you don't have the room to do a scratchout. [P11].

Additionally, participants reported a significant number of false negatives when they attempted to use the delete gesture. This observation is supported by the high number of Smart Delete Errors recorded from the video data. The system misrecognized 2.6 intended scratch out gestures per session for instructed participants. Most alarming is that this value underestimates the true error rate since these failures turned participants away from the feature altogether.

For delete, I feel using [explicit] buttons is more convenient because last time I saw a problem and delete didn't show up. Especially when I want to delete something very small. [P10]

This failure in recognition is caused by two factors: first, the implementation is too restrictive in distinguishing the scratch out gesture from regular draw strokes; second, the scratch out gesture itself may not be appropriate when attempting to delete relatively small strokes, surrounded by little whitespace. Facing these problems, even participants who made use of the inferred mode's delete noted that the explicit delete operation is more precise, making it the preferred method for deleting strokes in the drawing. Delete gestures are not sufficiently precise to avoid surrounding content.

4.2.2 Perceptions of Computational Intelligence

Instructed participants generally liked the inferred mode protocol as a method for adding and manipulating strokes in a sketch interface. The technique was perceived as being more efficient than the explicit buttons by the majority of users.

If you have to select the [explicit] button, then it's a little bit more work because you're kind of moving away from your picture; whereas if you're drawing in the flow of things, you can just click on [the mediator]. [P9]

Uninstructed participants were generally less positive; implicit users were very negative as a result of the false activations that plagued their sessions. However, participants who did not prefer the inferred mode protocol (both uninstructed and the few instructed participants) had some important observations and biases towards the protocol and computational intelligence in general.

To summarize participants' perceptions towards computational intelligence, there is a distinction between purposeful and multi-purpose interaction. The protocol tries to infer the user's intent and interpret the input appropriately. Embedded in this dialogue is the possibility of a misunderstanding between user and system. In contrast, selecting from explicit modes in a toolbar allows the user to be very specific in their intent. To quote P16, the explicit interaction paradigm is "purposeful," while the inferred mode – and similar computational intelligence – represents a multi-purposed paradigm.

> *I want it so that it's purposeful. I guess what you've done here is create shortcuts. [P16]*

The ambiguity that comes with multi-purpose interaction can be daunting and steer users away from such systems. The observation that the inferred mode protocol is not "purposeful" may be a common thread to some of the participants' perception of the inferred mode:

- The ambiguity inherent in the inferred mode leads to distrust and a sense of "error proneness"
- The protocol is less suitable for a particular task when compared to a specific tool
- The protocol is less accurate than explicit modes

Firstly, the relative complexity and inherent ambiguity in the interpretation of the input is perceived – at times, rightly so – to lead to more errors in interaction. P5 used the inferred mode's delete gesture but stayed away from the selection operations.

[Smart] Select is not very often used. I am just scared of making mistake and I don't want to confuse between the delete and select [functionality]. [P5]

P5 chose to bypass the inferred mode's select both because they did not perform many select operations, and because of a fear of errors. Intelligent systems like the inferred mode can be seen with distrust by their users who see the computational ambiguity as a potential for errors.

Secondly, when comparing the inferred mode to the purposeful interaction of explicit mode switching, participants perceive the multi-purpose nature of the former to be inferior for a particular task. For instance, P3 was a Instructed/Explicit user who mainly stayed away from the inferred mode. When experimenting after their final session, the user was surprised to learn that the inferred mode's Select Circle worked better than the explicit Select button mode in a particular case.

Kind of surprising the simple one didn't work, but the smart one did. [...] I tend to believe the [explicit modes] work better; more reliable and have more control. [P3]

Of note is the language used by P3; the explicit modes are referred to as the "simple" modes, recognizing the inherent complexity of the inferred mode. Additionally, participants perceived single purpose tools like the Select button mode as more accurate for the task they are built for than a multi-purpose tool like the inferred mode. This bias is one of specialization versus generalization of functionality. P12, for example, perceived the inferred mode's Select Circle to be less accurate than the Select mode, even while the internal source code is the same for both methods.

I thought [Smart Circle] Select wouldn't work as well than the button. I guess I never actually compared them. [P12]

P12 is one of the participants with the highest usage metrics of the inferred mode. Though they used and liked the features, their perception was that the inferred mode is

less accurate because it provides multiple functions as opposed to specializing functionality. Moreover, they realized that this was an unfounded perception as they had not explicitly tested their hypothesis. The inferred aspect of the protocol is seen as inferior in some cases and more error prone simply because it is not a separate tool bred for a singular purpose.

Though the inferred mode protocol is a relatively simple instance of computational intelligence, it is still partly plagued by the perception of inferiority. Most participants recognized its usefulness for quickly and efficiently editing diagrams. However, the fact that it uses a hidden computational model for determining output is combined with its multi-purpose nature to inspire distrust in the form of perceived inferiority and error-pronedness.

4.3 Learnability of the Inferred Mode Protocol

A major goal of this work was to understand the inferred mode protocol's learnability, both in the short and long term. In the context of this study, learnability is affected by a number of factors in noncommand interaction. First, users must be able to notice the feedback provided by the system and deduce the existence of a computational black box that behaves in some unknown way. Second, users must be able to determine how the system output is mapped to their input. Only by forming this mental model can the full system be understood to a usable degree. Finally, with enough experience, users must deduce the inner workings of the black box (e.g. how recognition works) and use this knowledge to come up with novel workflows.

As a result of the experiment setup, the discussion can focus on initial learnability using uninstructed participant groups to determine whether they notice the feedback and form a mental model of input to output. Moreover, the study enables the discussion of training and retraining behaviours that lead to discovery and proficiency with the system. Lastly, this work consider extended learnability by providing examples of advanced behaviours and an evolved understanding of the computational black box as a result of mastery of the system.

4.3.1 Initial Learnability

Using Grossman et al.'s definition of learnability [19], this work considers initial learnability to be the initial performance with the system.

When examining video data and interview data, there were three themes surrounding initial learning behaviours. First, for users instructed in the inferred mode, this section explores training and retraining behavior during interaction. Next, for participants in the Not Instructed condition, this work explores active experimentation versus passive experiences of these participants as they worked with the interface. Finally, this section considers reported impediments to discoverability and difficulties in learning.

Implicit Discovery and Early Discoverability

Participants in the Not Instructed condition (P13-24) were required to learn to use the inferred mode features without any explicit training, and no embedded help nor "What's This" widget was provided to clarify behaviors of the inferred mode, by design. Participants using an interface with embedded help would exhibit either the behavior of the Instructed participants – i.e. they would use the help feature to master the inferred mode techniques and then behave as Instructed participants – or the behavior of the Not Instructed participants. The goal with the Not Instructed users was to explore initial encounters with the inferred mode to determine the strategies employed to learn to effectively use the technique. These learnability attempts are analyzed in response to the Implicit Discovery and Early Discoverability hypotheses.

First, the Implicit Discovery Hypotheses outlined the prediction that the always-on nature of the Implicit interface – i.e. where the inferred mode protocol was on by default – would encourage participants to master the inferred mode faster, so that mediators were useful rather than a distraction, whereas participants using the Explicit interface – i.e. a separate "Smart" mode – would have little impetus to explore the behavior of inferred mode. In fact, the opposite was true for P13, P14 and P16; the presence of a "Smart" mode seemed to encourage experimentation for these participants. Though only half of the participants in the Not Instructed/Explicit condition experimented, there was no similar

catalyst for experimentation for the Not Instructed/Implicit users.

Early behavior of participants in the Not Instructed condition was similar when interacting with the inferred mode initially (either by default or through the "Smart" mode). Participants learned that clicking away from the mediator would cause it to vanish. However, participants with the Implicit interface would then continue to passively encounter and dismiss the mediator, and made no apparent effort to master the inferred mode techniques. As noted by P20

> *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. [P20]*

All Not Instructed/Explicit participants had five sessions to master the protocol. As seen in Table 4.1, except for P21 who experimented and eventually discovered the features, the remaining Not Instructed/Implicit participants did not make use of any of the smart features. This is in spite of ample opportunity to discover the inferred mode protocol. As Figure 4.6 demonstrates, there is no significant difference between the number of ignored/inadvertent activations of the inferred features between the Instructed/Implicit and Not Instructed/Implicit conditions. For instance, P14 ignored the Select mediator a total of 59 times over 5 sessions, with P20 seeing 128 instances. Despite the presence of these mediators, uninstructed participants never explored their behaviors.

Of note is that there is an exception to this rule: P21 continuously explored the features throughout the sessions and finally discovered them all. This user was interested in the interaction paradigm and took significant time to actively explore all facets of the interface. By the end, their use of the inferred mode appeared to converge with Instructed participants.

Confirming the Early Discoverability hypotheses, the analysis uncovered that users mainly explore the interface in the early stages, if at all. In the Not Instructed/Explicit condition, the existence of the "Smart" button prompted P13, P14 and P15 to explore the interface. P13 analyzed the Smart mode very carefully during the first session and

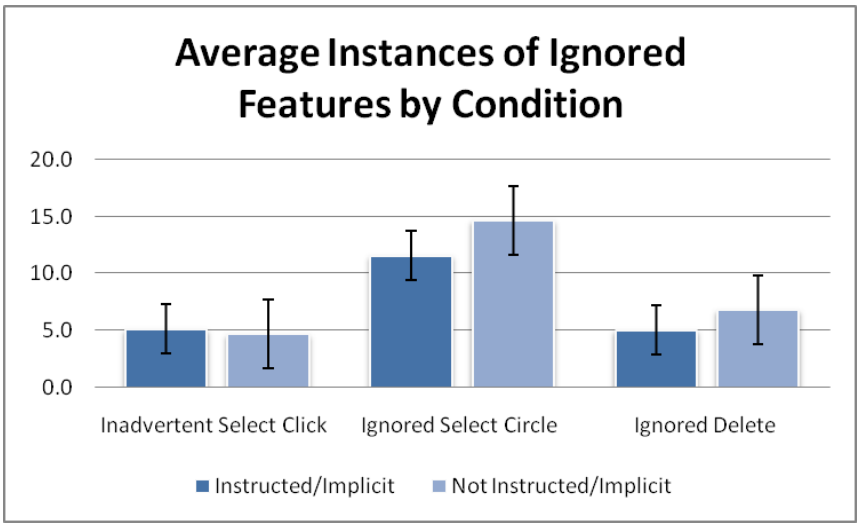

Figure 4.6: Average instances of ignored features by condition.

mastered all of the smart mode techniques, then converged on the behavior of the instructed participants for later sessions. P14 and P15 also tried to master the "Smart" features during the first session but failed initially. However, during the last sessions (sessions four and five, respectively) these participant began to understand and use the features. As the study was limited to five sessions, it was not possible to determine whether additional sessions would have increased these users' use of the Smart mode, and whether their behaviors would have converged on the behavior of instructed participants.

Overall, counter to experimenter predictions, having an explicit mode to enable the inferred mode serves as a catalyst for active exploration. This active exploration is critical towards the Not Instructed/Explicit participants' understanding of the features of the inferred mode protocol and mainly occurs in the early sessions outside of task-driven interaction. In contrast, the passive understanding gained by the Not Instructed/Implicit users was not enough to help them uncover the features in the five-session experiment.

Training and Retraining

While the Instructed participants did not need to actively explore the interface, to understand the basics of the inferred mode, they actively trained and retrained throughout the sessions. During the initial session, instructed participants would try each of the features carefully to ensure that they could successfully activate the inferred mode. On subsequent sessions, participants would begin to perform their tasks. As they required features of the inferred mode, they moved to white space on the display and practiced the inferred mode techniques prior to making use of the technique in their drawing. Figure 4.7 shows an example of one participant retraining on the delete gesture.

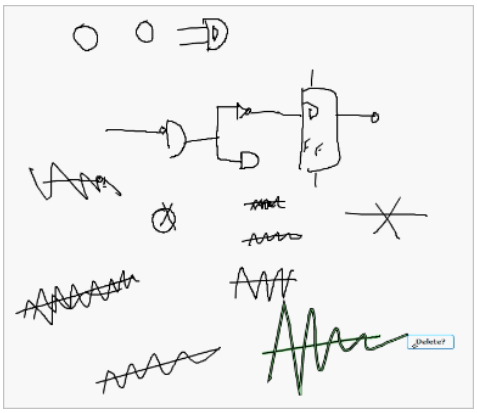

Figure 4.7: A participant using part of the canvas as a designated "training area" to experiment with the delete gesture

This training and retraining behavior demonstrates the value of "scratch space" within sketch interfaces. In GestureBar [4] researchers noted the need for multiple interactions of a gesture to train muscle memory so the gesture could be replicated successfully on the display. As a result of this need for training or practice, Bragdon et al. created a separate panel activated from the toolbar for users to experiment on with gestures. In this study's interface, there was also a need for scratch space where users can explore the behaviors of the interaction techniques and the tolerances associated with mediator activation. While GestureBar accomplishes this using separate panels, this study's participants seemed to prefer interaction with content on the actual drawing canvas in an unused area and cleaning

that area of the canvas up using a group delete operation after interaction. This distinction between participants in this study and participants in GestureBar's formative studies may be because of the distinction between a recognition interface like GestureBar, where each gesture must be recognized by a gesture recognizer, and sketch interface techniques like the inferred mode protocol, where the user needs unrecognized ink content to interact with, i.e. to select or to delete, in order to explore gesture behavior.

Impediments to Discoverability

An analysis of post-task interviews uncovers some important reasons why Not Instructed participants did not discover the features of the inferred mode. These reported impediments to discoverability fall into five themes:

- Perception of erroneous systems
- Lack of interest
- Confusion of expectation
- Lack of guidance
- Comfort or the perception of usefulness

A significant failure in Not Instructed/Implicit participants' discovery of the inferred mode features was previously noted. Though a lack of guidance certainly plays a large role, participants did not form a good mental model that links their input on the canvas with the mediators that appear. The result is that participants perceived the mediators as errors in the system, rather than useful functionality.

> *It's sometimes select, sometimes delete; I thought it's not a rule; I thought it's an error. I didn't notice it is related to the way I was selecting or something. [P22]*

This perception of an erroneous system is partially caused by the instances in which the mediators appeared. The given task asked participants to draw a variety of circuit diagrams with many small parts. Patterns in gestures (e.g. drawing a circle around other strokes brings up a Select mediator) are very difficult to distinguish when drawing many small elements. As a result, any passive learning regarding the functionality is limited by the inability to form these implicit patterns from the input stream. Moreover, this perception of an erroneous system further limits any efforts to actively experiment.

Secondly, though the "Smart" mode appears to encourage exploration, a lack of interest prevented other participants' discovery. This lack of interest is found in most of Not Instructed/Implicit participants as well. They were simply focused on the given task and were not interested in trying to actively experiment with the interface.

> *Oh, no I didn't notice [the Smart button]. What's the function of this button? [P15]*

Thirdly, some participants reported a mismatch between their own expectations and that of the system's. For instance, P16 expected the interface to react in some way when selecting the Smart mode. Furthermore, simply selecting the Smart mode takes them out of Draw mode. The user's expectation is that this mode's functionality lies outside of drawing strokes.

I clicked on the Smart button but I didn't try drawing. [..] When you click on it, it takes you out of draw mode. [..] I tapped on it and it just seemed irrelevant; it didn't appear to have functionality. [P16]

This is an example of mismatched expectations between designer and user. Though the researchers expected the user to experiment by adding strokes to the canvas, some participants exited the Smart mode when no immediate effect was discovered.

Moreover, the disinterest in discovery caused by mismatched expectations was exacerbated by a lack of guidance in this experiment's system. In addition to the minimal training the Not Instructed participants received, experimenters did not provide any embedded help options by design. The goal was to simulate a common scenario: a user wants

to perform a task (e.g. draw circuit diagrams) but does not take the time to go through tutorial videos or help menus. However, though few avenues for explicit or embedded help were provided, some users leave the onus on the designers to demonstrate features that are best suited for a particular task. In post-task interview, P17 mentioned that they simply did not explore the interface because the focus was on the task and not on discovery. There was no explicit requirement to use a particular tool and therefore little impetus to find it in the interface.

> *If I was told to carve a line, I would look to which button to carve the line with. [P17]*

Though the experimenters intentionally gave little guidance, it was surprising to find that participants would like designers to steer the user towards the most efficient tools for performing a task. This observation may be validation for the need for better tutorial tasks and GestureBar-like practice sessions.

Lastly, an alternative explanation for Not Instructed/Implicit participants' lack of exploration was simply habitual use of established tools such as explicit toolbar items (i.e. Button Select and Delete).

> *For the first 5 times I thought [the Select mediator] was an error; but by the 6th to 10th time I realized it may be another way of doing things; but by that time [..] you've already learned not to use that function. You're just used to go to the Select button. [..] My mentality is 'if I were to learn how to use this, it wouldn't be faster than what i'm doing right now' [P23]*

Habits form as a result of continued use with the system. Though P23 realized the mediators may specify additional functionality, there was no obvious impetus for exploration. There is no perceived benefit to take the time to experiment when the explicit tools are performing well.

Though discoverability is difficult to judge in an interface that uses computational intelligence, it is important to consider factors that impede learnability in the interface itself, in the given task, and in the perceptions of the participants.

4.3.2 Extended Learnability

Extended use over multiple sessions leads to a familiarity with both the benefits and limitations of available tools. Participants who understood the functionality of the inferred mode (P1-14) quickly began to combine operations to more effectively make use of the interaction techniques. This evolved understanding and user driven optimization exemplifies the study of extended learnability. This section presents learning behaviour past the initial stage and the related expert patterns of use that emerge in experienced users.

Evolved Understanding

As participants gained experience with the inferred mode protocol and the system as a whole, they demonstrated an advanced level of understanding.

Firstly, there were some differences between participants' perception of delete and select operations. The delete operation removes content from the display, whereas the select content simply changes the state of on-screen content. The delete operation, even in the presence of "Undo", was treated more carefully by the participants. Of note is that some participants were very careful to dismiss the delete mediator to avoid accidental activation, while they were more comfortable simply ignoring the select mediator.

Secondly, participants demonstrated a deep level of understanding by abstracting functionality. For instance, encircling content not only selects it, but the lasso also permanently cuts strokes. With extensive use, Instructed participants understood that the select functionality can effectively be treated as a "trim" tool. Smart Select Circle was, then, not only relegated for selecting and moving behaviours, but also became a tool to cut extraneous strokes and delete them by scratching them out. In the same way, Select Click was also absracted as a tool to recover selections by selecting the previously cut strokes or to verify stroke boundaries for further editing. The ability to treat the features as high level Trim and Recover functionality seems trivial to a relatively knowledgeable user, but beginners do not show evidence of such abstraction. Uninstructed users showed evidence of abstracting Select to Trim after a session, but in large did not use clicking to select as a recover mechanism in the same way Instructed users did.

Lastly, though most of the instances of evolved understanding presented were given by Instructed participants, Not Instructed participants also developed a passive understanding of the features. Participants tended to learn quickly to dismiss the mediators when they appeared, regardless of their understanding of functionality. Moreover, while Instructed participants implicitly dismissed appearing mediators by continuing to draw, some of the more careful uninstructed participants modified their workflow around them. When a mediator appeared, these participants either explicitly dismissed it or changed targets and started drawing content away from the affected areas. Such care around mediators was, for some users, a passive response to the ever changing system behaviour, and for others an active avoidance of features they did not understand or did not desire. Though Not Instructed participants suffered from a missing mental model between input and output, the output – or the effect of activating the mediator – was often understood. In this way, avoiding the mediator was a similar reaction to Instructed participants' specific care around content altering buttons (e.g. the Delete mediator).

User-driven Optimization

Observing participants' progress throughout the sessions, there were a few advanced behaviours which amount to user-driven optimizations of the given interaction tools. As they gained experience, participants combined their own pre-determined workflows with behaviours that optimized whitespace, speed, and number of errors. The result was unepexpected behaviours which uncovered valuable feedback regarding the usefulness of the provided tools and the evolution of behaviour.

The primary factor optimized by participants was, unsurprisingly, time of completion. Though it was previously reported that Instructed participants make extensive use of the inferred mode features, not all of the participants used it right away. P5, for instance, initially used the explicit modes out of habit. However, after experimenting with the features, they slowly transitioned away from the explicit modes. P5's initial session saw 15 instances of Button Delete and no instances of the inferred mode. By the last session, the participant almost exclusively used the inferred mode with 30 activations of the inferred mode's scratchout gesture and zero explicit deletes. This user's evolution is indicative of

a pattern in some instructed participants: an initial reluctance to work with the inferred mode is overcome after experimentation and significant use.

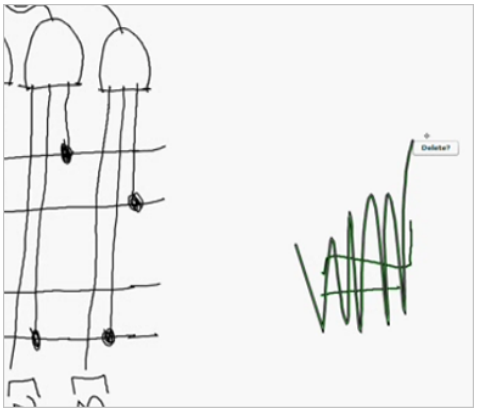

Figure 4.8: A participant scratching out accumulated strokes in their assigned "trash" region of the canvas.

Additionally, there were differences in how participants use the tools to perform pre-defined workflows. For instance, P23 and P4 had previous experience with digital circuits. A common workflow for these users involves laying out the logic gates first and then linking them appropriately. However, P23 was not instructed and did not discover the features of the inferred mode. In the latter sessions, P23 minimized the number of editing operations they performed by first laying out the gates in their final locations (Figure 4.9a). In contrast, P4 assigned an area as their virtual workbench which gathered all the logic gates they would ever need (Figure 4.9b). The speed of editing given by the inferred mode protocol allowed them to quicky select and drop gates wherever needed. Though it is impossible to generalize, it is interesting to note participants tweaking the same pre-defined workflow to take advantage of their current toolset's advantages.

Secondly, participants work towards optimizing the precision and use of whitespace with significant experience. In the inferred mode's delete, a scribble gesture must be made over content to be deleted. However, when content is densely arranged on the display, a scribble lacks precision to delete without affecting surrounding content. As a result, participants would move small content into white space and perform deletion away from surrounding

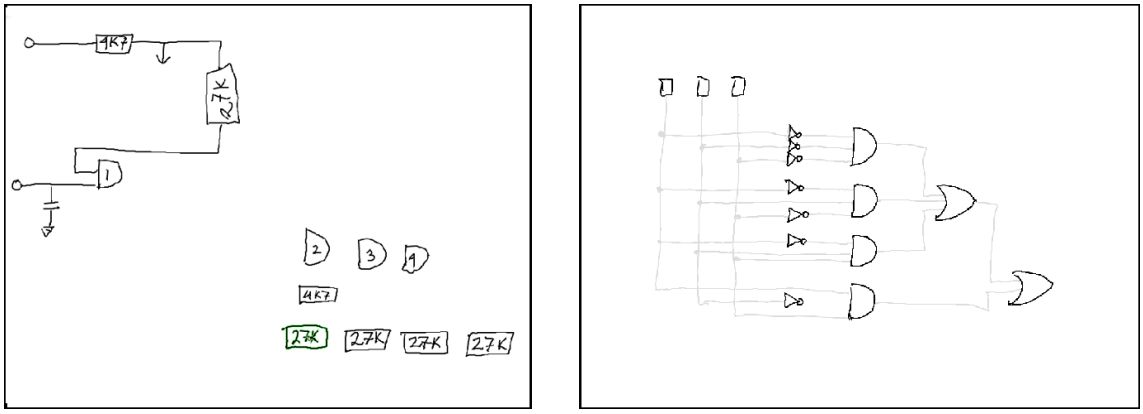

Figure 4.9: Experience influencing pre-defined workflows. Panel a. shows a virtual workbench from which gates are moved into place using the inferred mode. Panel b. shows a similar workflow that minimizes editing with the explicit buttons by laying out the logic gates in their final locations.

content. Participants also used circle select in the inferred mode to accumulate strokes for later deletion. They would move the strokes to a "trash" location on the display, as shown in Figure 4.8, and then apply the delete operation to all of the unwanted strokes at once.

Thirdly, experimenters consistently found that participants understand which features to use to maximize their precision. In addition to the above behaviour that bypasses the scratchout gesture's inherent imprecision, select click is often used when participants need very accurate selection. Though the select lasso provides substantial control it also requires significant care. Clicking to select is a much quicker alternative that does not change the state of the strokes.

Lastly, instructed participants gained a quick understanding for the reasoning behind false activations of the inferred mode. As a result, Not Instructed/Explicit participants found ways to minimize these erroneous activations by selectively enabling the Smart mode. P2 noted that the initial content creation stage of their session does not require many editing operations. This participant chose to use the default Draw mode with sparse instances of explicit editing operations (i.e. Button Select and Delete). As the second stage of the session required many editing operations, P2 used the Smart mode to quickly

switch between draw, select and delete operations using the inferred mode. This selective use of the inferred mode takes advantage of the protocol's relative speed, while minimizing erroneous activations in cases where editing is not common.

Chapter 5

Discussion

5.1 Usability of the Inferred Mode Protocol

The analysis of use and post-task qualitative data has uncovered valuable information regarding the usability of the inferred mode protocol. First, determining the toolset in even a simple drawing application is a difficult problem that cannot be solved by simply providing the most efficient tools without overlap. Second, the implementation and choice of gestures have an impact on how likely users are to return to a single-mode paradigm like the inferred mode. Lastly, though instruction has a big impact on users' overall perception of the system, participants come with pre-defined biases regarding noncommand user interfaces.

If given instruction, participants used and were generally highly positive of the inferred mode's circle to select and scratchout gesture. However, though instructed participants used the inferred mode extensively, it was not a replacement for traditional button modes. Feature bloat – the suggestion that an application contains an overly complex set of unnecessary features – has been a much maligned issue in the software world (e.g. [1, 2]). Though there is sufficient overlap between some of the features (e.g. Button Select, Smart Select Click and Smart Select Circle), not all participants used the same set. While some users had no need for Smart Select Click, others found it a useful alternative. Moreover,

the study suggests that, given experience, participants tend to optimize their workflows by using whichever tool works best for a particular situation – be it an explicit button mode, or an inferred mode feature. This experiment's findings mirror that of McGrenere and Moore [31]: users have different subjective experiences of "bloat" and benefit from a degree of flexibility and choice in their toolset. Over time, this experiments' participants developed a deep understanding of the tools available and enjoyed the ability to mix and match explicit and inferred editing tools as the situation dictated.

Moreover, users used the inferred mode protocol extensively even when given the choice of a traditional mode toolbar. The participants valued the protocol as an efficient means of interacting with content. Even when a feature like the protocol's scratchout gesture did not function according to the users' expectations, they were still positive of the gestural interaction paradigm. The fact that users consistently used the protocol throughout all of their sessions, regardless of the given interface, is highly promising.

However, participants' positivity regarding the protocol even in the face of erroneous output was not a carte blanche for failure. For instance, users had difficulty activating the scratchout gesture for deleting content in closed spaces. Such errors in implementations can drive users away from the inferred mode altogether. When a user failed to perform a gesture, they turned to other tools for attaining their goals. Though not a large problem in this experiment, re-capturing users who have turned away from a special input mode like the inferred mode as a result of implementation failures can be very difficult. Some of the users temporarily switched away from the inferred mode for the session when they encountered erroenous input, only to return later when its' use was "worth it." Additionally, designers of similar gesture systems should consider not only implementation-specific constraints such as recognition thresholds, but also the design of the gesture itself. For instance, a scratchout gesture is inherently more prone to errors when the target content is not surrounded by whitespace. An alternative gesture (e.g. circle to select and then simply cross once for deletion) can provide much better results and maintain a user's interest to the inferred mode protocol.

Additionally, subjective experience is highly affected by the level of instruction offered to the participants. The Not Instructed/Implicit participants had the inferred mode permanently enabled, which led to high levels of frustration as they struggled with a system

that behaved unpredictably. As a result, the Not Instructed/Implicit participants were much less lenient towards false positives in the system than Instructed participants. Users' perception of the failings of an interface that uses computational intelligence is highly affected by their developed mental model. This issue further exacerbates the need for proper learning opportunities and help systems that slowly build knowledge of the system.

Along the same lines, participants had various predetermined biases and misconceptions about the inferred mode protocol. The inferred mode protocol is an example of Nielsen's noncommand interaction paradigm [36] which tries to predict what the user is doing and interpret the input appropriately. The purpose of such computational intelligence is to meet the user halfway and help them perform their task. However, participants come with predetermined opinions of intelligence in interfaces. For instance, some of the users had a negative perception towards the inferred mode due to its lack of "purposefulness." This finding mirrors those found in adaptive systems like Eager [9]. Cypher reported that, although the Eager system provided efficient interaction, users were "uncomfortable with relinquishing control" [9]. In the same way, the reported lack of purposefulness is indicative of the paradigm of noncommand user interfaces: users no longer control the system with a strict object/verb interpretation; instead, input is inherently flexible, multipurpose and merely give hints of the user's intent with the expectation that the system meet the user halfway. Stemming from this observation, some of the users perceived the inferred mode as being somehow less accurate and less suitable for a task than explicit methods. Though the majority of the instructed participants made use of the inferred mode extensively, the experiment uncovered a distinct set of perceptions regarding the accuracy and suitability of noncommand user interfaces that affected user perceptions and behaviours. A more detailed qualitative experiment focused on user's pre-task perceptions regarding noncommand interfaces may be useful in uncovering intial hurdles of use.

Fortunately, though these reported pre-conceived misgivings regarding the inferred mode protocol as an interaction paradigm were generally negative, there is hope. Users who were not forced to use the inferred mode (i.e. using the Explicit interface) and stayed away from it (e.g. P13, P16, P17) mentioned that they may go beyond their initial negative opinion of the noncommand paradigm. With time, P21 even mentioned that, like the Microsoft Ribbon interface, they may grow to appreciate its usefulness. Similar to the

Microsoft Ribbon, the biggest challenge of the inferred mode is to get users accustomed to this interface paradigm [11]. Disimilarly, however, noncommand user interfaces suffer from negative perceptions related to their embedded ambiguity and are only now becoming more popular in user systems. It is to be expected, then, that getting accustomed to such a different interaction paradigm will be more difficult than acceptance of the new Ribbon user interface.

5.2 Learnability

A major goal of this research was to study the learnability of interaction techniques that use computational intelligence to assist the user. This observational experiment looked at the learnability of the inferred mode protocol in a more realistic scenario. This section discusses the findings in terms of lessons learned about the study of learnability itself and about the inferred mode protocol in specific.

5.2.1 Learnability of the Inferred Mode Protocol

In terms of the study of learnability of the inferred mode protocol, there were vastly different behaviours and understanding form between the user groups. Instructed participants alotted time during the first sessions to get a better understanding of the gestures. These users understood that the purpose of the experiment was to evaluate a novel gestural interaction technique and tried their best to understand it well. The act of introducing and explaining a novel interaction technique draws the user's attention and encourages experimentation.

In contrast, uninstructed participants were presented with a task and a drawing application, with no particular hints as to the functionality of objects in the interface. These users focused on the task and how to complete it as fast as possible. As a result, any mediators that appeared were perceived as errors in the system. The participants did not link the feedback provided by the Saund and Lank Inferred Mode Protocol to their input. Adding to the list of impediments of discoverability was a confusion as to what should

happen when selecting the "Smart" mode in the interface or even if the extra effort to experiment was even worth it. The study design forced an all too common situation where the user wishes to perform a task through a "walk up and use" approach. The experimental setup unveiled many of the impediments to such an undirected approach to noncommand interactions. Though placing the inferred mode functionality in a separate "Smart" mode encourages experimentation by nature of its labeling, this study's results suggest that enabling it by default does not produce a reliable model for how such interaction paradigms function. In the absence of tutorial videos (e.g. GestureBar [4]) or helpful tooltips, the user is left with an interface that is familiar, yet mysteriously difficult to predict.

This disconnect between interaction designers and users is a function of their perceptions and expectations of one another. Designers may expect users to search for help and seek instruction or otherwise discover the features on their own. Some users expect the system to present them with novel functionality up front and do little outside of using common tools to perform their task. In essence, users are focused on the task at hand, and will use tools they are familiar with to best perform their task. A lack of interest – or rather goal-orientedness – is a large problem in initial discoverability of novel features. Moreover, the study results suggest that participants also do not necessarily find new features useful or "worth it" to learn, even if they discover them. In terms of the inferred mode, this disconnection meant that participants who were not instructed had issues with discoverability and extended learnability as their skills stayed mainly at the baseline.

In contrast, instructed participants were given an idea about how system output maps generally to their input. Given a hint of the existence of the computational black box that sits at the core of the inferred mode protocol, participants formed a strong mental model and, through active experimentation, deduced the inner workings of the black box. The resulting understanding allowed these instructed participants to invent new workflows that take advantage of all available tools in interface and otherwise use the canvas in ways the designers did not imagine. An advanced understanding of the inferred mode protocol stemmed from active experimentation induced either by explicit instruction providing a baseline framework, or by an intriguingly labeled mode that houses the computational intelligence.

5.2.2 Study of Learnability

In designing the methodology and executing the study, there was much learned about the nature of learnability when it comes to computational intelligence in interfaces. First, the task that participants are asked to perform and its domain have large repercussions on user behaviour. Second, care must be taken to consider the prior experience of the chosen user sample in studying a relatively uncommon example of noncommand interaction. Lastly, learnability happens at various stages and selecting an appropriate timeframe in order to capture the transition from novice to expert is highly important for interaction techniques that use computational intelligence.

Firstly, when designing the study, the task choosen can play a large role in the findings relating to learnability. This study simulates a real task of drawing and editing circuit diagrams. This allowed for a reasonable measure of ecological validity. However, the circuits chosen for the task ended up affecting the results in unanticipated ways. For instance, circuit diagrams often have small connector circles to indicate that a wire "is-connected-to" another wire. These connector circles are helpful in disambiguating whether two wires are connected, or one simply goes over top of another for aesthetic reasons. However, in the context of the inferred mode protocol, the connectors provide interesting side effects. Certainly, when forced to use the inferred mode protocol, participants found many more false positives while drawing the small connectors than if they were given a different task. However, for uninstructed participants, they were also a source of confusion which led to disatisfaction and frustration. The connectors are often drawn as spirals producing small shaded-in circles. In such close proximity, the system can interpret the spirals as either a selection, or a scratchout gesture prompting a delete mediator. The ambiguity inherent in drawing the small objects did not allow uninstructed participants to form a mental model between their input and the system's output. Though it is common knowledge to take care of the task researchers choose for studying interaction techniques, there are subtle effects on the results that can be missed without a close inspection of the particulars of the interation with the task.

A second noteworthy observation is the effects of the user sample and their experience on the results relating to learnability. A common definition of users when dealing with

the study of learnability is to consider new users – those without any formal training – and experienced users. However, the inferred mode protocol represents a novel and uncommon interaction paradigm. In order to properly study learnability, experimenters must additionally consider users who have experience with drawing systems as presented in this study, but not necessarily with gestural interaction. This additional parameter is described by Davis et al as "subsequent learning" [10]. Considering this additional type of user in the analysis unveils another dimension to the study of learnability. All of this experiment's users had significant experience with drawing programs such as Microsoft Paint, or Photoshop. However, none of them used tablet PCs and, most significantly, were not experienced with similar types of noncommand interaction found in the inferred mode protocol. For instructed participants, the demonstration of a novel interaction technique encouraged them to experiment more than they would have otherwise in a more traditional application. In contrast, uninstructed participants were presented with what looked like a simple drawing application. The apparent familiarity to an application like Microsoft Paint may have led to users focusing immediately on the task at hand. Though it was surprising how few uninstructed participants were curious enough to click on the mediators that appeared, the lack of experimentation was detrimental to the discoverability of the inferred mode.

Thirdly, of note is the importance of separating initial from extended learnability and to study the transition from one stage to the next. Starting from a baseline which separates the intial conditions for participants (i.e. instructed vs uninstructed) produced an interesting flow of skills and experiences. Active experimentation and retraining can lead to very similar behavioural patterns even when participants start under different instrction. In contrast, a lack of experimentation leads to a vastly different subjective experience. Passivity is not rewarded well in interfaces that use computational intelligence, perhaps as a result of the increased complexity and inherent ambiguity. Forming a mental model of the interaction is difficult without some exploration external to a results-driven scenario of use. Finally, being able to track user progress over a long period of time helped uncover expert behaviours that often go unreported in traditional evaluations of interaction techniques. Finding a balance between a long term observational study and a qualitiatively rich lab-controlled study is difficult but potentially rewarding. The attention to extended

learnability – one that happens as a result of extensive use with a system – in this study was a source of great qualitative findings that enriched the study of the inferred mode protocol.

Finally, evaluating noncommand user interfaces has unique challenges due to the often ambiguous understanding of input. In addition to challenges regarding learnability of a, as of yet, rarely used interaction paradigm, such computational intelligence can lead to difficulties in studying the formation of mental models. For instance, P14 and P15 actively explored the interface but failed due to the uncertainty of the recognizer. When not providing instruction, the ability for users to form a mental model is dependent on their linking input to output. Providing even a simple recognizer used in the inferred mode protocol provides difficulties in performance and hinders the study of learnability of the technique. Due to imprecise tuning of the parameters, the exploratory attempts of P14 and P15 were disrupted as a result of false positives and negatives. As similar looking inputs produced different outputs, participants' confidence fell and their interest to uncover the features waned. This fall in confidence as a result of recognizer performance mirrors the experimenters' findings in another example of recognition-based interaction [34]. During the evaluation of motion gestures for mobile interfaces, experimenters discovered that learnability is hurt by performance dips as a result of an uncertain input stream. Figure 5.1 shows an example of two users' performance, measured in accuracy to perform the given motion gestures over time.

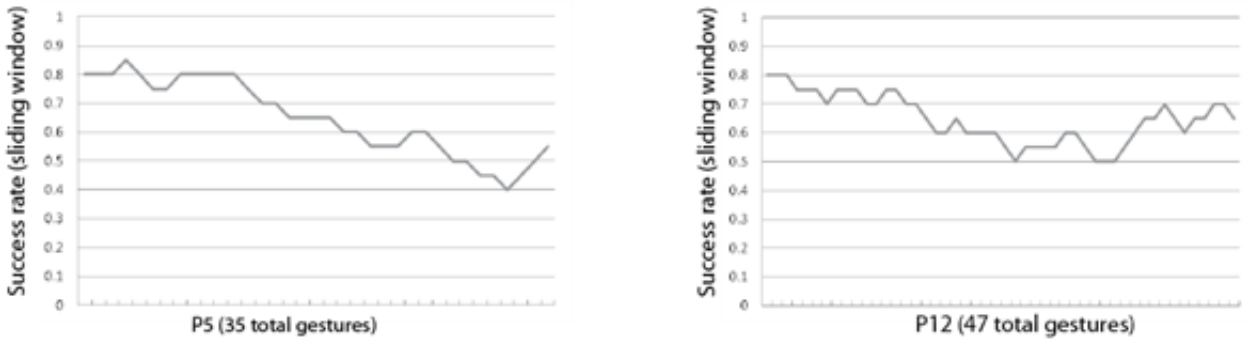

Figure 5.1: Two participants' performance over a five input rolling-window in a study evaluating mobile motion gestures.

During this study, participants' performance dipped to extremely low levels as they attempted to perform the gestures. As recognition performance fell significantly, users tried to diagnose why those recognition failures were occurring. In the case of motion gestures, participants had no way to characterize why gestures failed. As a result, they would try to vary the intensity, the timing, the direction, the device angle, etc. In essence, users tried to explore the space of recognizer inputs to determine whether some other set of parameters of movement would enhance accuracy. This pattern of recognition performance affected confidence and created a negative feedback loop in this study of the inferred mode as well. However, this problem was exacerbated for users P14 and P15 who received no specific instruction on the features. Though recognition parameters can be tweaked, it is difficult to tune it to all users, especially if given no instruction of the input domain. The difficulty in providing feedback in recognition-based noncommand interaction affects the study of learnability significantly as active exploration is easily deterred.

Chapter 6

Conclusions and Future Work

The inferred mode protocol is an example of Nielsen's noncommand interaction paradigm [36]. 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" [36]. 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, there are many of the same pitfalls associated with past generations of intelligent interfaces [47, 41]. For example, users had difficulty developing mental models of how the inferred mode protocol worked. As 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 Instructed participants preferred the inferred mode protocol and used it extensively. 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. 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, creating an explicit mode for noncommand interaction seemed to work well. For half of the Not Instructed/Explicit participants, the "Smart" mode gave them a clue that there was a non-standard aspect to the interaction, and motivated them to understand how the interface worked. P13, P14 and P16 actively explored the Noncommand features of the inferred mode, with the first mastering the technique early on and behaving like an instructed participant. The latter two developed an understanding of the smart mode over five sessions of sketching. In contrast, making computational intelligence standard in the interface by embedding it directly into the "Draw" mode caused significant problems for participants in the Not Instructed group.

This study explored the learnability and use of features of the inferred mode protocol using a multi-session observational study. Results show that, with instruction, participants value intelligent interface techniques, and make liberal use of them during drawing. Furthermore, this work highlights lessons learned for incorporating noncommand behaviors into sketch interfaces in realistic settings.

This chapter ends with an outline of potential future work uncovered through and closing thoughts.

6.1 Future Work

As a result of the study, there are two branches that warrant further exploration. First, there are a number of design enhancements that can be made to the inferred mode protocol. Finally, there is considerable room for improvement in terms of feedback and recognition.

6.1.1 Design Enhancements

During 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. 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.

Current versions of the inferred mode are currently experimenting with options for Delete Selection. One that appears to hold promise and maintains the default pen-and-paper behavior is a "select-then-cross" operation where users first select an object (using smart circle select) then draw a line through the selection to prompt to Delete. If they press the mediator, a delete occurs. Otherwise, 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. P3 noted that selection and cutting of curves is a common and often 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 selections, an undo stack would eliminate the need for select click.

Thirdly, sketching with the inferred mode protocol uses an inked stroke as a primitive. Given this, the inferred mode protocol can expand its editing toolset by allowing strokes to be merged and cut at will while still maintaining its independence from a specific domain (e.g. digital circuit drawing). For example, two strokes can be automatically merged if they are relatively close and have similar spline properties at their end points (e.g. c_0 continuity). Feedback about strokes' endpoints and local properties can be overlaid on the drawing without significantly distracting the user. Providing a simplified mechanism to group and combine strokes has the potential to improve editing tasks.

6.1.2 Feedback and Recognition

Noncommand interaction like the inferred mode protocol provides significant challenges in communicating the inner model to the user and aiding them to perform better. As demonstrated, without an impetus for active explorations, users have difficulty creating

a mental model of behaviour. This problem is exacerbated by performance variability created by recognizers.

One idea currently being explored to address both the issue of feedback and recognizer variability is a multiple threshold recognition system that can be used to help users who have difficulty understanding and performing commands. When discriminating commands such as *circle to select* and *scratchout to delete* from regular stroke input, designers often create a threshold, i.e. a criterion value, that best trades off between false positives (accidental activations) and false negatives (failed attempts to perform the command gesture). If the criterion value is too permissive, many false positives will occur. However, if the criterion value is too restrictive, it may become very difficult for the system to reliably identify intentional user gestures from its input stream.

A solution to this problem is to simply provide multiple thresholds for recognizing command gestures. Consider Figure 6.1 illustrating the allowable input space for strokes using the inferred mode protocol.

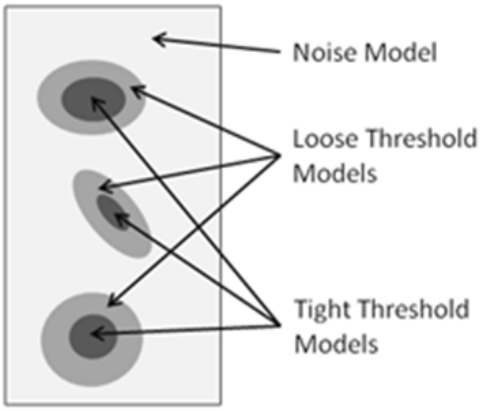

Figure 6.1: An illustration of a two-level thresholding recognizer for stroke command gestures.

The proposed two-level thresholding technique provides two recognition thresholds to improve performance and is summarized as follows: if a user-performed stroke does not meet a strict threshold for recognition as one of the command gestures, we then consider the gesture using a relaxed threshold a more permissive criterion value and wait to see

if a similar input stroke follows it. The system recognizes a gesture either if the end-user performs a tightly thresholded stroke gesture (i.e. success in the first instance), or if the user performs two relaxed thresholded gestures within a short period of time. Such a recognizer with two thresholds can assist users who have difficulty performing commands such as select circle or delete. Moreover, recognizing two loosely thresholded input strokes can be a sign that a user needs feedback to determine how to converge onto the template gesture. If the recognition process can be broken down into independent features that need to lie within a threshold for activation, a similar bi-level thresholding approach for each feature can outline specific problems with the input through its loose model and how to fix it. Such a recognizer that uses multiple thresholds of recognition can be used to effectively improve true positives while enabling the detection of specific problems in the user input and suggest feedback mechanisms.

6.2 Conclusion

Supporting computational intelligence in the sketching task has been a much studied problem. Traditional user interface elements such as buttons, lists, and menus are unwieldly when applied to the inherently fluid and ambiguity-laden task of sketching.

A large issue in translating traditional WIMP-style widgets to the sketching domain is the mode problem which deals with how to assist the user in interpreting stylus input. Inking, selecting, deleting and translating content are just some of the potential interpretations for user input that are often broken up into operational modes. However, there is significant cognitive load associated with modes that rises logarithmically with respect to their number.

The inferred mode protocol is a non-command method of interaction that uses contextual cues (how and where a stroke is drawn) to interpret the input. As the user draws, the system attempts to infer intent by recognizing that encircled content may imply selection, or that scratching out content may mean that the user wants to perform a delete operation. The user can resolve ambiguity in input by selecting a mediator that appears local to the input stroke.

This work deals with four major research problems:

- How to evaluate computational intelligence in interfaces?
- What can this study say about the usability of the inferred mode protocol as a means to infer user intent? Do participants make use of the protocol if given more traditional choices?
- What are important aspects dealing with its learnability, both in the short and long term?
- How does behaviour change over time? What can be inferred from users' continued evolution of behaviour?

In order to answer these questions, this work presented a long term observational experiment that looks at the usability and learnability of the inferred mode protocol. This study design serves as a platform for evaluating long term use of similar interfaces that use compuational intelligence. The results of this experiment suggest that users who are instructed in the use of the inferred mode protocol are positive towards the inferred mode protocol and make use of it even if given the choice of a traditional interface paradigm. These users see use in the protocol and are more lenient in their judgement than their uninstructed peers, even while expeiencing similar rates of false positives and significantly higher rate of false negatives. In terms of learnability, allowing the user to explicitly enable the inferred mde through the interface helps discoverability by encouraging active exploration. This active exploration is critical towards forming a mental model that links input to output. In contrast, enabling the inferred mode by default leads to a passive understanding that does not result in discovery.

The results of the study suggest significant impediments to discovering features of the inferred mode protocol including a lack of interest and confusion in between what the user and the system expect. Moreover, when encountering such computational intelligence in interfaces, users tend to have negative intial biases that must be overcome for successful integration of new features.

Finally, this work explored various methodology in developing the study. This experiment combines quantitative and qualitative analysis of use over a long period of time. Such a long term analysis helps uncover difficulties in studying noncommand interaction like the inferred mode protocol as a result of inherent ambiguity.

References

[1] The bloatware debate. *Computer World*, August 8, 1998.

[2] Do computers have to be hard to use? complex, volatile, frustrating; there must be a simpler way. *New York Times*, May 28, 1998.

[3] Christine Alvarado and Randall Davis. Sketchread: a multi-domain sketch recognition engine. In *ACM SIGGRAPH 2007 courses*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[4] Andrew Bragdon, Robert Zeleznik, Brian Williamson, Timothy Miller, and Joseph 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.

[5] Andrea Bunt, Cristina Conati, and Joanna 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.

[6] J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. An empirical comparison of pie vs. linear menus. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '88, pages 95–100, New York, NY, USA, 1988. ACM.

[7] Stuart K. Card, Allen Newell, and Thomas P. Moran. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 2000.

[8] Philip R. Cohen, Michael Johnston, David McGee, Sharon Oviatt, Jay Pittman, Ira Smith, Liang Chen, and Josh Clow. Quickset: multimodal interaction for simulation set-up and control. In *Proceedings of the fifth conference on Applied natural language processing*, ANLC '97, pages 20–24, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics.

[9] Allen Cypher. Eager: programming repetitive tasks by example. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, CHI '91, pages 33–39, New York, NY, USA, 1991. ACM.

[10] S Davis and S. Wiedenbeck. The effect of interaction style and training method on end user learning of software packages. *Interacting with Computers*, 11(2):147–172, 1998.

[11] Martin Dostál. User acceptance of the microsoft ribbon user interface. In *Proceedings of the 9th WSEAS international conference on Data networks, communications, computers*, DNCOCO'10, pages 143–149, Stevens Point, Wisconsin, USA, 2010. World Scientific and Engineering Academy and Society (WSEAS).

[12] G.J Elliott, E Jones, and P Barker. A grounded theory approach to modelling learnability of hypermedia authoring tools. *Interacting with Computers*, 14(5):547 – 574, 2002.

[13] K. A. Ericsson and H. A. Simon. Verbal reports as data. *Psychological Review*, 87:215–251, 1980.

[14] G Fischer. *Shared knowledge in cooperative problem-solving systems–integrating adaptive and adaptable components*. Elsevier Science Publishers, 1994.

[15] George Fitzmaurice, Azam Khan, Robert Pieké, Bill Buxton, and Gordon Kurtenbach. Tracking menus. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, UIST '03, pages 71–79, New York, NY, USA, 2003. ACM.

[16] Vinod Goel. *Sketches of Thought*. MIT Press, Cambridge, MA, 1995.

[17] Mark D. Gross. The electronic cocktail napkin–a computational environment for working with design diagrams. *Design Studies*, 17(1):53 – 69, 1996.

[18] Tovi Grossman, Patrick Baudisch, and Ken Hinckley. Handle flags: efficient and flexible selections for inking applications. In *GI '09: Proceedings of Graphics Interface 2009*, pages 167–174, Toronto, Ont., Canada, Canada, 2009. Canadian Information Processing Society.

[19] Tovi Grossman, George Fitzmaurice, and Ramtin Attar. A survey of software learnability: metrics, methodologies and guidelines. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pages 649–658, New York, NY, USA, 2009. ACM.

[20] Tovi Grossman, Ken Hinckley, Patrick Baudisch, Maneesh Agrawala, and Ravin Balakrishnan. Hover widgets: using the tracking state to extend the capabilities of pen-operated devices. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 861–870, New York, NY, USA, 2006. ACM.

[21] François Guimbretière, Maureen Stone, and Terry Winograd. Fluid interaction with high-resolution wall-size displays. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, UIST '01, pages 21–30, New York, NY, USA, 2001. ACM.

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

[23] W. E. Hick. On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, 4(1):11–26, 1952.

[24] 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.

[25] Ken Hinckley, Patrick Baudisch, Gonzalo Ramos, and Francois 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.

[26] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3d freeform design. In *ACM SIGGRAPH 2007 courses*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[27] Gabe Johnson, Mark D Gross, and Ellen Yi-Luen Do. Flow selection: a time-based selection and operation technique for sketching tools. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '06, pages 83–86, New York, NY, USA, 2006. ACM.

[28] Gordon Kurtenbach and William 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.

[29] James A. Landay. Silk: sketching interfaces like krazy. In *Conference companion on Human factors in computing systems: common ground*, CHI '96, pages 398–399, New York, NY, USA, 1996. ACM.

[30] 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.

[31] J. McGrenere and G. Moore. Are we all in the same "bloat"? In *GI '00: Proceedings of Graphics Interface 2000*, pages 187–196, 2000.

[32] Joanna McGrenere, Ronald M. Baecker, and Kellogg S. Booth. A field evaluation of an adaptable two-interface design for feature-rich software. *ACM Trans. Comput.-Hum. Interact.*, 14, May 2007.

[33] Elizabeth D. Mynatt, Takeo Igarashi, W. Keith Edwards, and Anthony LaMarca. Flatland: new dimensions in office whiteboards. In *Proceedings of the SIGCHI con-*

ference on Human factors in computing systems: the CHI is the limit, CHI '99, pages 346–353, New York, NY, USA, 1999. ACM.

[34] Matei Negulescu, Jaime Ruiz, Yang Li, and Edward Lank. Tap, swipe, or move: attentional demands for distracted smartphone input. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 173–180. ACM, 2012.

[35] Mark W. Newman and James A. Landay. Sitemaps, storyboards, and specifications: a sketch of web site design practice. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, DIS '00, pages 263–274, New York, NY, USA, 2000. ACM.

[36] Jakob Nielsen. Noncommand user interfaces. *Commun. ACM*, 36(4):83–99, 1993.

[37] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995.

[38] Jakob Nielsen. Usability inspection methods. In *Conference companion on Human factors in computing systems*, CHI '95, pages 377–378, New York, NY, USA, 1995. ACM.

[39] Jakob Nielsen. Novice vs. expert users, February 2000. http://www.useit.com/alertbox/20000206.html.

[40] Jakob Nielsen. Usability 101: Introduction to usability, 2012. http://www.useit.com/alertbox/20030825.html.

[41] Donald A. Norman. *The Design of Everyday Things*. Basic Books, September 2002.

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

[43] John Rieman. A field study of exploratory learning strategies. *ACM Trans. Comput.-Hum. Interact.*, 3(3):189–218, September 1996.

[44] 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.

[45] Jaime Ruiz, Andrea Bunt, and Edward Lank. A model of non-preferred hand mode switching. In *GI '08: Proceedings of graphics interface 2008*, pages 49–56, 2008.

[46] Eric Saund and Edward 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.

[47] L. Tesler. The smalltalk environment. *Byte*, pages 90–147, 1981.

[48] Barbara Tversky. What do sketches say about thinking? In T. Stahovic, J. Landay, and R. Davis, editors, *AAAI Spring Symposium on Sketch Understanding*, Menlo Park, CA, 2002. AAAI Press.

[49] Robert Zeleznik and Timothy 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.