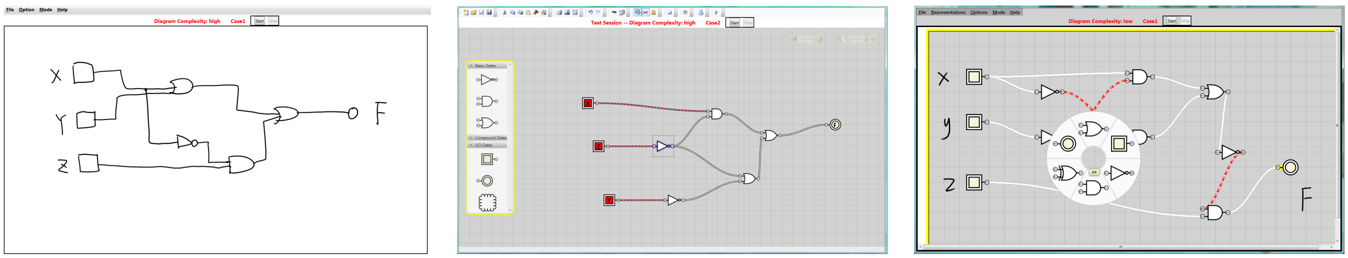
User Perceptions of Drawing Logic Diagrams with Pen-Centric User Interfaces

Bo Kang*
Department of EECS
University of Central Florida
4000 Central Florida Blvd.
Orlando, FL, 32816

Jared N. Bott†
Department of EECS
University of Central Florida
4000 Central Florida Blvd.
Orlando, FL, 32816

Joseph J. LaViola Jr.‡
Department of EECS
University of Central Florida
4000 Central Florida Blvd.
Orlando, FL, 32816



(a) Sketch Interface (b) Drag-and-Drop Interface (c) Hybrid Interface

Figure 1: Three interfaces for creating logic diagrams in our study. (a) An ideal natural 100% recognition accuracy sketch interface as if using pen and paper. (b) A drag-and-drop interface with a static logic gate toolbar on the left side to drag and drop gates. (c) A hybrid interface, combining features from sketch and drag-and-drop, with a radial menu for creating gates.

Abstract

Researchers hypothesize pen-based interfaces are the input method of choice for structured 2D languages, as they are natural for users. In our research we asked whether naturalness, similarity to pen and paper, is more important than speed of entry and ease of use by performing a study comparing interfaces for creating logic diagrams. We compared a Wizard of Oz based sketch interface with 100% accuracy, a drag-and-drop interface, and a hybrid interface combining features from sketch and drag-and-drop. Eighteen college students with logic gate diagram backgrounds participated in the study. We found that participants finished fastest with the hybrid interface, but ten out of eighteen participants felt that the sketch interface was fastest. Ten participants ranked the sketch interface easiest to use, while the hybrid interface was rated highly on ease of use metrics. Participants showed significant inclination towards the sketch interface as being natural. While the hybrid and sketch interfaces were ranked best for overall preference, neither was ranked more than the other. Even though the hybrid interface was empirically faster, user preferences for the interfaces varied, with many participants favoring the sketch interface. Finally, we tested for correlations between overall ranking for interfaces and other rankings on the interfaces and found the strongest correlation to be with ease of use. Based on our results, we believe that combining sketching with other interface paradigms could lead to better interfaces for structured 2D languages.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Interaction Styles, Evaluation/Methodology;

*e-mail: bkang@cs.ucf.edu
†e-mail: jbott@cs.ucf.edu
‡e-mail: jjl@eecs.ucf.edu

1 Introduction

Pen-based user interfaces have been extensively researched for several decades. One important area of pen-based user interface research is sketching, where users draw as if using pen and paper. Due to their similarity to pen and paper, sketch-based interfaces are commonly believed to be the best method for structured 2D languages, such as mathematics, chemical bond diagrams, music notation, and circuit diagrams [8]. Researchers commonly investigate new pen-based interfaces and methods for new domains [11], and work is performed on making recognizers better [7, 9, 22]. However, little thought is given to whether they are ideal, and what makes them ideal. While naturalism (similarity to pen and paper) is important, is it more essential than other factors like speed and ease of use? Should we continue our research into pen-based interfaces for structured 2D languages, and if so, what areas should we focus on for improvement?

Our goal in this research was to determine if people would prefer a sketch interface with 100% recognition accuracy over other computer interfaces that were faster and more traditional. To that end, we performed a study where we asked participants to use and compare three interfaces for creating logic diagrams. Figure 1 shows the three interfaces we used, a sketch interface, a drag-and-drop interface, and a hybrid interface that combined handwritten wires and labels from the sketch interface with the spatial UI from the drag-and-drop interface and a pie menu for creating gates.

2 Related Work

In the pen-based user interface community, researchers have been trying to examine the natural interactions users have with pens. One area of pen-based interfaces that is especially interesting is sketching, where users leverage their existing notions of pens and drawing to create complex, natural 2D drawings. Structured 2D languages are well-suited for sketching as they are often complex; sketch-based interfaces have been created for many kinds of structured 2D languages [2, 7, 15, 23].

Some researchers have created new pen-based interaction techniques exploring the natural interactions users have with pens. Crossing is a technique in which strokes replace mouse movements and clicks [1]. CrossY uses the crossing technique to create a system where single strokes can be used to perform complex UI interactions [3]. Other researchers have tried to fuse different interaction techniques to create more task-oriented pen-based tools, such as DENIM [16]. DENIM is an informal pen-based sketch system created to rapidly design websites by sketching pages and interacting with them using gestures and visible controls. LogicPad is a tool that can create logic diagrams using a hybrid style of sketch and drag-and-drop interaction [12]. While LogicPad presents a usability evaluation of its hybrid interface, we are examining hybrid interfaces in comparison to other pen-based interfaces.

Research into input methods has also been performed. MacKenzie et al. compared mouse, trackball, and pen input for pointing and dragging tasks and found the highest performance was for the pen during pointing and for the mouse during dragging [17]. Forsberg et. al. investigated the applicability of pen computing in desktop environments with diagramming task using mouse-and-keyboard and pen-based techniques [6]. This work is closely related to ours, but we are solely investigating user perceptions of pen-based interfaces without examining mouse and keyboard interfaces.

Other work has been performed on evaluating sketch-based interfaces and user perceptions of sketch-based interfaces. Wais et al. evaluated recognition feedback in the context of circuit diagram sketch recognition [21]. Vatavu et al. performed studies to examine user perceptions of making gestures with a stylus [20]. They found that gesture difficulty was most closely correlated with completion time. Bott et al. performed a Wizard of Oz study investigating user perceptions of mathematics handwriting recognition with respect to accuracy, recognition mode, and input complexity [4]. As recognition accuracy affects users' perceptions and efficiency in finishing the task, we apply the same Wizard of Oz approach to our logic diagram sketch interface to give users the perception that there is 100% recognition accuracy. In order to let the sketch interface be as natural as possible, we simulate batch recognition (recognition after the completion of writing, invoked by performing an action) and delay feedback to users until they invoke recognition.

Negulescu et al. describe the inferred mode protocol, which enables switching modes efficiently in a sketch interface, and they evaluated the learnability and usability of mode inference, by checking if the interaction mode assisted users in drawing logic diagrams more efficiently [19]. We had similar goals in designing the hybrid interface used in our study, but are exploring user perceptions of pen-based interfaces for structured 2D input in order to guide development of pen-based interfaces and techniques.

3 User Interfaces

We used a Wizard of Oz approach for all three user interface, where we programmed each interface with the Boolean equation to output for each diagram according to the participant's ordering (see Section 4.2 for details on how we balanced the equations). In other words, each interface returned a pre-specified equation regardless of what diagram the user created.

3.1 Sketch Interface

We created a batch-recognition pure sketch interface that simulated 100% recognition accuracy using a Wizard of Oz approach (see Figure 1a). The sketch interface did not perform any recognition to transform the input strokes into gates, wires, or labels. Participants sketched an entire diagram and then had the experimenter invoke recognition. Participants were free to sketch logic gates, wires, and labels as they saw fit without constraints. We implemented a scribble-erase gesture for participants to erase ink strokes. Feedback came in the form of a Boolean equation for the sketched diagram, which was part of the Wizard of Oz approach. We chose to forgo other more direct forms of recognition feedback because we wanted to eliminate distractions and to be as close to pen and paper as possible.

3.2 Drag-and-Drop Interface

We implemented an interaction model that would be familiar to most users, a traditional WIMP-based interface [5]. We based this interface on [13] (see Figure 1b). A static toolbar on one side of the interface provides gates that can be dragged onto the workspace. Wires are created by tapping on one terminal in a gate, and dragging the stylus to another gates's terminal. Labeling a gate was performed using the stylus and the keyboard. Wires and gates can be deleted by selecting the item(s) and using the delete key on the keyboard. Wires can also be deleted by tapping on them.

3.3 Hybrid Interface

We developed a hybrid drag-and-drop and sketch interface (see Figure 1c) similar to part of LogicPad [12]. We wanted to take the strengths of the above two interfaces and combine them to create a fast, easy to use interface. With the complex symbols in a logic diagram, we thought that users would find it quicker and easier to drag a gate from a toolbar than sketch it. However, with wires and labels, we thought that it would be quicker and easier to draw them with a stylus, so we combined these two methods in our hybrid interface.

While dragging and dropping gates from a toolbar can be easy, it is not particularly efficient to drag from a static menu on one section of the screen. In order to make our hybrid interface faster and more usable for stylus input, we adopted a gesture-based method to trigger the logic gate toolbar, a radial menu [14]. The radial menu appears in-place where the tap-and-hold gesture has been performed, keeping the user from having to move across the screen to create a gate. From our pilot studies, we determined a hold time of 0.5 seconds worked best for triggering the radial menu, as it was long enough that users rarely triggered the menu by accident, and short enough that they did not have to wait long.

We used the same gate visualization in the drag-and-drop and hybrid interfaces. Most gates have two input terminals and an output terminal, and terminals are used to start and finish wires. Wires are made in much the same way as in the drag-and-drop interface: the user draws from one gate's terminal to another gate's terminal. In contrast to LogicPad, we developed a visualization to help users to create wires more efficiently. While creating a wire, when the stylus is near a terminal, the terminal is highlighted yellow, indicating that the user is close enough for a wire to be created by raising the stylus from the screen. Also, in contrast to LogicPad, we thought that users might want to branch wires, so they can do so by making a dot on an existing wire and drawing toward a gate. This will create a wire from the starting terminal of the branched wire.

Handwriting recognition is used to provide gate labels; after recognition, we display a typeset label in place of the ink strokes. As with the sketch interface, wires, gates, and labels can be erased with a scribble-erase gesture.

4 Experimental Study

We conducted an experiment designed to explore the usability of our three interfaces. Based on our initial observations, we hypothesized that:

- Primary: Overall, participants will prefer the sketch interface over the hybrid and drag-and-drop interfaces.
- Secondary: The hybrid interface will be faster than the sketch and drag-and-drop interfaces.
- Secondary: The sketch interface will be rated more natural than the hybrid and drag-and-drop interfaces.

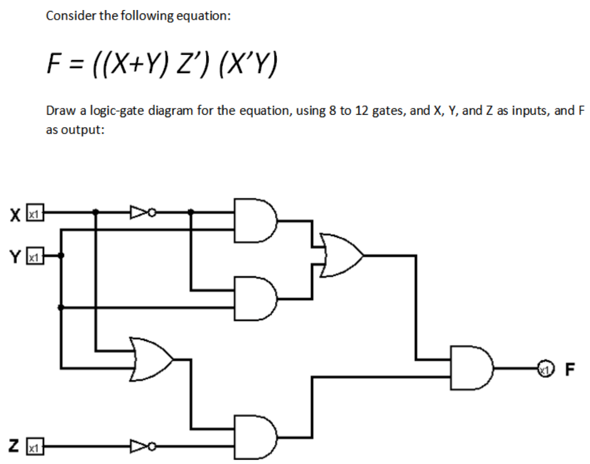
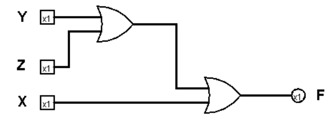
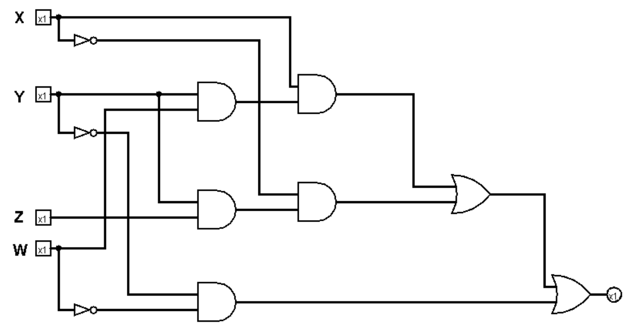


Figure 2: An example homework assignment problem participants were asked to grade. For each homework problem, participants were asked to copy the given diagram using the specified interface, and compare the Boolean equation given by the interface with the equation provided in the homework problem.



(a) Low Complexity Diagram



(b) High Complexity Diagram

Figure 3: Example logic diagrams used in our study. (a) A low complexity diagram with 6 gates and 5 wires. (b) A high complexity diagram with 15 gates and 18 wires.

While we thought participants would spend less time copying diagrams with the hybrid interface, they would prefer the sketch interface over the hybrid and drag-and-drop interfaces because of the sketch interface's naturalness.

4.1 Subjects and Apparatus

We recruited 18 participants (15 male, 3 female) to participate in our study. The mean participant age was 24, with the youngest 19 and the oldest 30. All participants were college students enrolled in electrical engineering, computer engineering, and computer science programs, and they all had taken a discrete mathematics course that covered Boolean algebra. We chose this population because they had knowledge of Boolean algebra and logic diagrams.

Participants worked on an HP Compaq tc4400, 12.1 inch tablet PC running Windows 7. The participant's computer was remotely monitored by the researchers using screen-sharing software.

4.2 Experimental Task

We asked participants to complete three copy-and-verify tasks, one for each interface. Each task was framed as a homework assignment participants were asked to grade. Each question on the assignment gave a Boolean equation and asked the "student" to draw a logic diagram for the equation (see Figure 2). Participants were asked to input the diagram using the specified interface and verify whether the student had drawn the diagram correctly. Therefore, each participant graded three homework assignments. Each homework assignment consisted of six questions divided into two sections, where each section was based on diagram complexity.

While copying a diagram, participants were free to input the diagram's components (wires, gates, labels) in any order they pleased. Once the participant finished inputting the diagram, and was satisfied that they had done so correctly, the researcher used the application to obtain the diagram's equivalent Boolean equation. This was done in a Wizard of Oz manner, where we did not perform recognition on the diagrams, but programmed the interfaces to provide the appropriate equation when prompted. Once the participant received the equation from the interface, they compared it with the equation given in the homework problem, and marked on the homework sheet if the diagram was correct.

We chose to use this task as it gave participants a clear motivation for why they were copying diagrams (grading a homework assignment), motivation to make sure that they had copied them correctly (not wanting to incorrectly mark a problem as incorrect), and motivation for why they would care about effort and completion time (grading many assignments takes time).

We designed eighteen pairs of Boolean equations. Each pair of equations was associated with a logic diagram, so we designed eighteen logic diagrams. For each pair of equations, one equation was equivalent to the associated diagram, and one was inequivalent. Each homework problem presented one of the two equations and the associated diagram. This simulated the student getting the homework correct (when the equation and diagram were equivalent) and incorrect (when equation and diagram were inequivalent). The diagrams were divided into two sets based upon the diagram complexity, low and high. We decided upon these complexity level divisions by using diagrams similar to those found in introductory problems in textbooks [18] as our low complexity diagrams, and building from there for high complexity diagrams.

All low complexity diagrams were designed to have three input gates. The nine low complexity diagrams were further subdivided into three subsets, where all diagrams in a subset had the same number of gates and wires. One subset had 6 gates and 5 wires; the second subset had 8 gates and 8 wires, and the third subset had 10 gates and 11 wires. The nine high complexity diagrams had four input gates and were also divided into three subsets. The first subset had diagrams with 12 gates and 15 wires; the second subset diagrams had 15 gates with 18 wires, and the third subset had diagrams with 17 gates and 22 wires. The diagrams made use of input, output, AND, OR, XOR, and NOT gates. A low complexity diagram is shown in Figure 3a and a high complexity diagram in Figure 3b. Diagrams on the homework assignment were presented to participants in a typeset form in order to provide a clear visualization. We randomly divided the diagrams from each complexity set such that each interface had three low complexity diagrams (one from each subset) and three high complexity diagrams (one from each subset). Each homework assignment was randomly assigned to have one to three (at most half) of the equation-diagram pairs inequivalent.

4.3 Experimental Design and Procedure

We used a 3 by 2 within-subjects factorial design where the independent variables were user interface and diagram complexity, and the dependent variable was completion time. The completion time is defined as the time interval from when the researcher pressed a 'start' button until participants finished inputting the logic diagram and pressed a 'stop' button.

The experiment began with participants completing a pre-questionnaire to collect demographic information, as well as their experience using a Tablet PC and their capability to solve Boolean equations.

1. How familiar are you with tablet PCs?
2. How familiar are you with Boolean algebra? I can solve any type of Boolean algebra problems, such as Boolean minimization problems.

These last two questions were assessed using 7-point Likert scales. For the question about experience using Tablet PCs, 1 equaled unfamiliar and 7 equaled familiar. For the question about capability to solve Boolean equations, 1 equaled none and 7 equaled any. The median for both questions was approximately 5 out of 7. We asked participants to assess their own Boolean algebra capabilities because we were interested in their own perceptions of their knowledge. One limitation of our approach is that participants may have over- or underestimated their Boolean algebra abilities. For each interface, participants were shown a short video explaining how to use the interface. They were then asked to familiarize themselves with using a Tablet PC and stylus by inputting the same sample logic diagram using the three user interfaces. During the practice session, participants were encouraged to ask the researcher questions about using the interfaces. We also informed participants that their sketches would be recognized by the sketch interface without mistakes. The researcher also monitored participants remotely to remind them about each interface's features.

Once they had familiarized themselves with the three interfaces, each participant was given a series of tasks to perform. When ready to copy the diagram, they informed the researcher so that timing could begin. During diagram input, the researcher monitored the participants in order to make sure that they correctly copied the diagram. When they finished entering the logic diagram, the participant informed the researcher who stopped timing, and clicked the 'recognition' button on each interface. Only after the researcher invokes recognition does the system display any feedback, which is in the form of a Boolean equation. The participant then compared the system-generated equation to the equation on the homework assignment and marked the correctness of the homework problem.

The order in which participants worked through the different interfaces was randomized and balanced such that one-third of the participants worked with each interface first, second, and last. The order of diagram complexity was also randomized and balanced across interfaces. After completing the task for an interface, participants were asked to fill out a questionnaire where they rated their agreement with seven statements on a 7-point Likert scale.

1. I found it easy to use this interface to make the logic gate I want to use.
2. I found it easy to use this interface to make the wire I want.
3. I found it easy to use this interface to make the label I want.
4. I found it easy to use this interface to spatially arrange the logic gates.
5. I found this interface easy to use for creating logic gate diagrams.
6. It was quick to create the logic gate diagram.
7. It was frustrating to use this interface.

After completing all three tasks, a final interview was conducted to allow participants to summarize their perceptions and feelings on the three interfaces. Participants were also asked to rank the three user interfaces based on ease of use, speed of entry, naturalness, and overall preference and gives reasons for each ranking.

1. Please rank the user interfaces in order from most easy to use (1) to least easy to use (3).
2. Please rank the user interfaces in order from fastest (1) to slowest (3) on speed of entry.
3. Please rank the user interfaces in order of most (1) to least natural (3).
4. Please rank the user interfaces in order from most (1) to least (3) on overall preference.

4.4 Results

4.4.1 Primary Hypothesis

Did participants prefer the sketch interface over the hybrid and drag-and-drop interfaces? To analyze the overall rankings (and the three other rankings) of the interfaces (see Figure 4), we performed a Chi-Square test on each ranking, which tells us whether there is variance in how the interfaces were ranked. In other words, these tests tell us whether participants thought the interfaces were equally easy to use, speedy, etc. In overall ranking, we had a close race between the sketch and hybrid interfaces. The hybrid interface had nine first place rankings, while the sketch interface had seven. In second place rankings, both had eight. The drag-and-drop interface was preferred the least with fourteen third place rankings. We saw significance in the Chi-Square test on overall preference ($\chi^2_4 = 24.667, p < 0.001$). Our central hypothesis, that the sketch interface would be preferred was not confirmed by these rankings, as there was no clear winner in overall ranking.

4.4.2 Secondary Hypothesis: Naturalness

We also saw significance in the Chi-Square test on naturalness ($\chi^2_4 = 52.667, p < 0.001$). Fourteen out of eighteen participants ranked the sketch interface as the most natural, as it is similar to pen and paper. This confirms our second hypothesis, that participants would find the sketch interface most natural. Fourteen participants ranked the hybrid interface second in naturalness, and fifteen participants ranked the drag-and-drop interface third.

4.4.3 Secondary Hypothesis: Speed

In terms of speed of entry, we saw significance in the Chi-Square test ($\chi^2_4 = 24.000, p < 0.001$). Ten participants ranked the sketch interface first and seven ranked the hybrid interface first. Ten participants ranked the hybrid interface second, and thirteen ranked the drag-and-drop interface third fastest. The participants' perceptions of speed ran contrary to our first hypothesis that participants would complete the diagrams faster with the hybrid interface than the sketch and drag-and-drop interfaces. While participants **thought** the sketch interface was fastest, was it? We analyzed the timing data (see Table 1) using a Repeated Measures ANOVA and used the average completion time for the three diagrams per complexity level. The ANOVA showed that interface ($F_{1.350,22.946} = 28.580, p < 0.001, \eta^2_p = 0.627$), complexity ($F_{1,17} = 353.499, p < 0.001, \eta^2_p = 0.954$), and their interaction ($F_{2,34} = 12.576, p < 0.001, \eta^2_p = 0.425$) were all significant factors. In order to see which interface was quicker, we performed paired t-tests where we looked at the times across all accuracy levels (see Table 2). To correct for Type I errors, we applied Holm's Sequential Bonferroni Correction [10]. Contrary to the participant rankings, the hybrid interface took less time than the sketch interface and the drag-and-drop interface, which matched our first hypothesis.

Interface	Drag-and-Drop		Hybrid		Sketch	
Complexity	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev
Low	92.9	19.5	70.8	12.1	70.0	14.8
High	226.4	38.7	186.6	33.3	202.4	34.6
Overall	159.7	74.1	128.7	63.7	136.2	72.1

Table 1: Mean completion time in seconds.

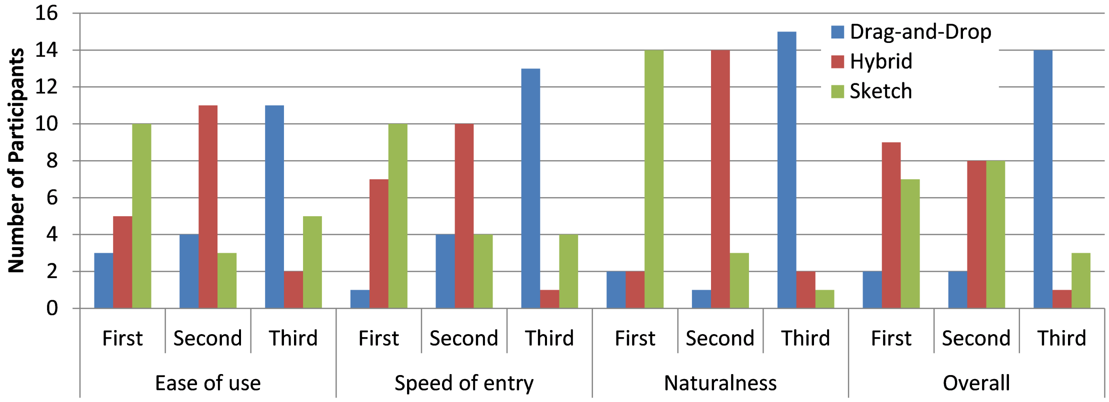


Figure 4: Participant rankings of the interfaces on various criteria. A ranking of first indicates greatest agreement with that criteria.

Interface	DD-H	DD-S	H-S
Low	$t_{17} = 5.478$ $p < 0.017$	$t_{17} = 4.745$ $p < 0.025$	$t_{17} = 0.310$ $p = ns$
High	$t_{17} = 6.542$ $p < 0.017$	$t_{17} = 3.913$ $p < 0.05$	$t_{17} = -4.721$ $p < 0.025$
Overall	$t_{35} = 7.938$ $p < 0.017$	$t_{35} = 6.094$ $p < 0.025$	$t_{35} = -2.922$ $p < 0.05$

Table 2: T-tests showing whether each interface pair was significantly different (DD=Drag-and-Drop, S=Sketch, H=Hybrid). A negative t-value indicates the first listed interface is faster than the second.

Interface	Drag-and-Drop		Sketch		Hybrid	
Statement	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev
Easy Gate	5.72	1.447	6.17	.924	6.39	.698
Easy Wire	5.00	1.680	5.89	1.231	5.94	1.305
Easy Label	**4.67**	**1.815**	**6.89**	**.323**	**6.33**	**1.085**
Easy Arrange	**6.22**	**.878**	**4.56**	**1.854**	**5.78**	**1.263**
Easy Diagram	**5.39**	**1.243**	**5.67**	**1.138**	**6.28**	**.752**
Quick	5.61	1.335	5.83	1.249	6.11	1.023
Frustrating	3.06	1.731	2.50	1.249	2.11	1.023

Table 3: Mean responses for Likert statements. Bold results indicate statements that had significant differences in responses.

Statement	
Easy Gate	$\chi^2_2 = 1.9224, p = ns$
Easy Wire	$\chi^2_2 = 3.250, p = ns$
Easy Label	$\chi^2_2 = 21.382, p < 0.001$
Easy Arrange	$\chi^2_2 = 9.480, p < 0.01$
Easy Diagram	$\chi^2_2 = 7.000, p < 0.05$
Quick	$\chi^2_2 = 1.161, p = ns$
Frustrating	$\chi^2_2 = 1.846, p = ns$

Table 4: Results from the Friedman test on each Likert statement. Three statements had significance (shown in bold), Easy Label, Easy Arrange, and Easy Diagram.

4.4.4 Other Results

With the Chi-Square test on the ease of use rankings, we saw significance ($\chi^2_4 = 17.667, p < 0.005$). Many participants ranked the sketch interface as the easiest to use (10 participants), with the hybrid interface second (11 participants), and the drag-and-drop interface third, where 11 participants ranked drag-and-drop last.

After analyzing the Likert-item data (see Table 3 for mean responses) using Friedman tests (see Table 4), we found significance for three statements: 3, 4, and 5. All three statements related to ease of use: "I found it easy to use this interface to make the label I want", "I found it easy to use this interface to spatially arrange the logic gates," and "I found this interface easy to use for creating logic gate diagrams." We then performed Wilcoxon Signed Rank Tests for each statement and corrected using Holm's Sequential Bonferroni Correction (see Table 5). For making labels, the sketch interface was easiest to use, followed by the hybrid interface, and finally the drag-and-drop interface. Participants found the sketch interface harder to arrange gates in than in the drag-and-drop interface. This is easy to understand, as the only way to perform spatial arrangement in the sketch interface is to erase and redraw. For the statement on ease of use in making diagrams, we saw a single significant result in the Wilcoxon Signed Rank test. The hybrid interface was easier to create diagrams with compared to the drag-and-drop interface. This matches with the participant rankings of the interfaces on ease of use, where drag-and-drop was ranked last 11 times. In contrast, the hybrid interface was primarily ranked second, though it was rated highly.

We hypothesized that participants would prefer the sketch interface over the other interfaces, and speculated that they would do so because of the sketch interface's naturalness and inspite of the hybrid interface's speed. Our analysis of the rankings shows that the sketch interface was indeed natural and that the hybrid interface was fast, even if participants did not notice that speed. In order to provide some insight into participant reasonings on why they preferred one interface over another, we calculated Spearman's rank correlations between overall ranking and the other measurements we took (completion time, the seven ratings, and the other three rankings) (see Table 6 for the correlation coefficients). The correlation with the greatest coefficient was between overall ranking and the ease of use ranking. Overall ranking with naturalness was next, followed by speed. This implies that ease of use was actually more important than naturalness or speed. However, it does not tell us how much ease of use was impacted by naturalness or speed.

5 Discussion

5.1 Naturalness

Our second hypothesis that participants would find the sketch interface most natural was supported by the participant rankings on

Interface	DD-H	DD-S	H-S
Easy Label	$Z=-2.610$ $p<0.025$ $r=-.435$	$Z=-3.542$ $p<0.017$ $r=-.590$	$Z=-2.264$ $p<0.05$ $r=-.377$
Easy Arrange	$Z=1.867$ $p=ns$ $r=.311$	$Z=2.925$ $p<0.017$ $r=.488$	$Z=2.057$ $p=ns$ $r=.343$
Easy Diagram	$Z=-2.676$ $p<0.017$ $r=-.446$	$Z=-0.819$ $p=ns$ $r=-.137$	$Z=1.895$ $p=ns$ $r=.316$

Table 5: Wilcoxon Signed Rank Tests showing whether each interface pair was significantly different (DD=Drag-and-Drop, S=Sketch, H=Hybrid) on the indicated statement. A negative Z-value indicates the first listed interface was rated higher than the second.

Correlation with Overall Ranking	ρ	p
Ease of Use	.778	.000
Naturalness	.694	.000
Speed	.639	.000
Frustration	.411	.002
Easy Diagram	-.409	.002
Easy Label	-.397	.003
Completion Time	.397	.003
Quick	-.342	.011
Easy Gate	-.314	.021
Easy Wire	-.281	.039
Easy Arrange	.056	.690

Table 6: Correlations between overall ranking and other rankings, ratings, and completion time.

naturalness, where fourteen participants found the sketch interface most natural. As one participant wrote for why they selected the sketch interface as most natural, "sketch allows you to write as if you are drawing on paper." In contrast to the similarity of the sketch interface to pen and paper, one participant, who selected hybrid as the most natural interface, commented that "after getting used to hybrid, it was more natural to select [a] gate at [the stylus's] location, and connect inputs to gates. Sketch could arguably be more natural but took more time to make [sure] wires were correct." Another participant who also ranked the hybrid interface as most natural commented that "naturalness should be redefined in pen-based user interface for drawing logic diagram as the combination of gesture-based drag-and-drop visual controls and sketching of wires and labels." Two users who thought the drag-and-drop interface was most natural, did not like to draw too much, and felt that the drag-and-drop interface was best because it resembled their previous experience using desktop environments.

5.2 Speed and User Perceptions

Participants worked fastest with sketch and hybrid for low complexity diagrams, and with hybrid for high complexity diagrams and overall. This matches our first hypothesis that the hybrid interface would allow participants to create diagrams most rapidly. Examining the Likert statement on quickness, the Friedman test did not show significance (see Table 4), meaning that we cannot reject the possibility that participants did not find a difference in speed between the interfaces. The mean response for the hybrid interface was high, so participants thought it was quick, but the other interfaces were rated fairly high as well. Quite interestingly, participants ranked the sketch interface fastest over half the time (10 participants) and the drag-and-drop interface fastest only once. We can see a disconnect between user perception of speed and actual speed. We think this can be explained by several factors.

First, when using the sketch interface, participants did not have to switch tasks; that is they only drew. In contrast, the drag-and-drop and hybrid interfaces required participants to move to or open a menu, drag and drop gates, arrange gates, and draw wires and labels. Participants likely felt that they were doing more, and doing more made them feel like they were taking more time.

Second is the issue of error blame; when sketching, we think there is little perception of computer interaction (except for the erasing gesture), so participants would feel as though nothing was impeding them from completing the task. Mistakes would be attributed to themselves, since they didn't feel as though they were using a computer. When using the drag-and-drop and hybrid interfaces, more complex interactions are required, and participants likely had a sense of using a computer application. When a mistake was made, participants could shift blame to the computer application, and these "external" mistakes would cause participants to feel as though they were working slower.

Finally, we think that participants may have slowed themselves down with the sketch interface by making dense drawings. This is especially true for participants that did not draw well. They had to recheck their drawings repeatedly because of their unclear drawings. When a mistake was made, a scribble-erase gesture might unintentionally overlap with several strokes, causing the participant to have to redraw various items. This disconnect in perceived speed and measured speed is also interesting because we provided a sketch interface that was always correct in its ability to recognize and understand the logic diagram. Even with 100% recognition accuracy, the sketch interface was slower than the hybrid interface for many situations.

5.3 Ease of Use

5.3.1 Making Diagrams

For the statement on ease of use in making gates, the analysis did not show significance between the interfaces (see Table 4), which is interesting, since all three interfaces had quite different methods for creating gates. The drag-and-drop interface had a traditional method for creating gates that can be found in diagram-creation applications like Visio[1]. A small number of participants had a few issues with dragging the gates from the toolbar, "dropping" the gate while it was still on the toolbar. Perhaps this was a result of using a stylus. Both the sketch and hybrid interfaces had a mean response of over 6, indicating that the interfaces were quite easy to create gates with. This leads us to pose the question of whether and how they can be improved upon.

Analysis of the statement on ease of use in making wires did not show significance either. Both the drag-and-drop and hybrid interfaces used a UI visualization for the wires and required participants to start and end their wires at a specific location (the gate terminals). The drag-and-drop interface required the participant to be more accurate in starting the wire at the terminal compared to the hybrid interface. The UI visualization also constrained the shape of wires to simple curves and straight lines. In contrast to the drag-and-drop and hybrid interfaces, the sketch interface allowed the participants to begin their wires anywhere and to make them look any way they wished. Finally, the hybrid interface provided a visualization (a yellow highlight) when a wire was being created and the stylus was sufficiently close to a terminal, which allowed participants to know when they had successfully created a wire. We thought the user interface differences made to the hybrid interface would cause it to be easier to use, particularly over the drag-and-drop interface. However, we cannot say whether the sketch interface's freeform wire

[1]Microsoft Visio, http://office.microsoft.com/visio

drawing compares favorably with the hybrid's easy wire creation.

As previously discussed, we did see significant differences in how easy it was for participants to make labels in the three interfaces. While the sketch and hybrid interfaces both allowed the participants to write their labels with the stylus, there were minor user interface differences between the two. Using the hybrid interface, participants had to write their labels a bit away from input and output gates in order to not move the gates. This accidental movement likely led to the difference in perceived ease of use between the sketch and hybrid interfaces. In contrast to those interfaces, the drag-and-drop interface required the participant to switch input modes to use the keyboard for typing the labels. Having to make a modal switch slowed down users (as did having to use the barrel button on the stylus to select the rename function from a context menu) and made it harder to use.

5.3.2 Arranging Diagrams

While the hybrid interface provides similar mechanisms for spatial rearrangement to the drag-and-drop interface, it has a flaw in erasing gates. When scribble-erasing, if the participant starts too close to a gate, the gate will be moved instead of triggering a gesture. While participants practiced the scribble-erase gesture during the training session, they still made more mistakes in deleting gates and wires in the hybrid interface compared to the drag-and-drop interface. This might be improved through the use of mode inferencing techniques such as those described in [19].

Some participants also expressed that they preferred the sketch interface because of their own drawing abilities. In contrast, some participants drew cluttered diagrams and worried about their drawings. These participants might have benefited from recognition feedback to show them that that their drawings were neat and formal enough. One participant commented that he does not like to draw, so he did not like using the sketch interface as much as the others.

Making neatly aligned diagrams can be beneficial, as it can make it easier to copy the diagram. Therefore, some participants moved the gates in order to align them. Another participant commented out that he wanted to spatially move and align the gates for every interface, and the sketch interface did not provide a way for participants to move their strokes without erasing them. Thus, aligning gates was harder than with the other interfaces, and he did not like the sketch interface.

Finally, for the hybrid interface, some participants triggered the radial menu without intending to do so, holding the stylus to the screen without moving while thinking about their next actions. Sometimes participants even generated gates when this happened. Though they had practiced bringing up the pie menu during the practice session, a few participants still had this issue. This issue should be alleviated with more practice.

5.3.3 Frustration

Our analysis did not show significance between the interfaces (see Table 4) for the statement about frustration. The mean responses for frustration were low for all three interfaces, indicating that there was little that was frustrating about using the interfaces, as agreement with the statement indicates higher levels of frustration. Hybrid had the lowest mean frustration rating (though, again, there were not statistically significant differences between the interfaces in our test). We think this is because it allowed participants to label gates in a natural way, spatially arrange the gates, and create gates with little work.

5.4 Overall Ranking

Looking at the rankings, we can see that there was no overall "winner," which is contrary to our hypothesis. The hybrid interface was ranked first nine times and second eight times, and the sketch interface was ranked first seven times and second eight times. The drag-and-drop interface had fourteen third place rankings. While the hybrid interface was fastest in our timings, as previously discussed, participants felt otherwise. Instead, they found it to have speed similar to the sketch interface. The sketch interface was strongly perceived to be most natural, but the sketch interface was not significantly chosen as best overall. It is however clear that drag-and-drop was ranked last. It is clearly indicated that participants took a number of factors into consideration when deciding on an overall winner. Participants thought the sketch interface was easy to use, quick, and natural, but did not overwhelmingly rank it best overall.

Is the naturalness of a sketch interface more important than the other factors we examined? It does not appear to be more important to several of the participants, as seven ranked the sketch interface as first in naturalness, but chose another interface as first in overall preference (all chose hybrid). Our correlations also show that ease of use was more closely correlated with overall preference than naturalness or speed, but because the sketch and hybrid interfaces did not have equivalent ease of use, we cannot say for certain.

The perceptual gaps on speed (and to some degree, ease of use) make things quite interesting. Participants who ranked sketch first overall (seven participants) also ranked sketch first for all three other criteria, except for one participant who did not rank sketch first for speed. For the drag-and-drop interface, two people ranked it first overall. One participant ranked it first for the other three criteria, and the other participant ranked it first on ease and naturalness. The hybrid interface exhibits the most interesting ranking behavior. Nine participants ranked the hybrid interface as best overall, of those, only one participant ranked hybrid first in all three other categories. Two participants who ranked hybrid first overall did not rank hybrid as first in any other category. We think this further indicates a strong gap in user perception on the various criteria.

Participants had a hard time deciding how to rank the interfaces according to participant comments. Some participants felt that the differences in overall experience between sketch and hybrid was so tiny that they could not distinguish easily between the two. Therefore, they tried to rank based on their previous rankings on other criteria. One participant who ranked hybrid as the best overall commented out that "sketch was easy to use, but drawing everything out took much time, lack of ways to fix mistakes." Another who rated hybrid as the best overall commented that "hybrid combined the way the diagram looked, how to input it, correct errors, easier than the others. Sketch was nice because it was fast for small diagrams."

5.5 Sketch Recognition Accuracy

Finally, after asking the participants for their rankings, we explained to each participant our motive for the experiment, and asked whether they would change their rankings if the sketch interface did not have 100% accuracy. Except for three people who rated sketch as the worst overall interface even with 100% accuracy, all other participants expressed that decreased accuracy would cause them to change their rankings. Some participants wanted to change all four rankings and place sketch as the lowest interface. Other participants wanted to change their rankings only for ease of use and overall preference, and rank sketch as the lowest, but make no change on naturalness and speed of entry. Clearly, participants thought that 100% sketch recognition accuracy was important, though it remains to be seen how dramatically it affects user perceptions. In order to provide a natural sketch interface, we need to make further improvements to recognition accuracy, as a perfectly accurate pure sketch interface was not overwhelmingly better than the hybrid interface.

5.6 Recommendations

While pure sketch interfaces are extremely natural and relatively fast, improvements can be made, as can be seen by the hybrid interface used in our experiment. Perhaps from a performance point of view, pure sketch-based interfaces are not always the best interface for a structured 2D language. However, from a preference point of view, there is still a powerful aesthetic that makes them useful even when they are not the most efficient.

Since many of the problems participants had with the sketch interface dealt with spatial arrangement, sketch interfaces will have to provide tools to help users in this regard. For instance, when sketching logic diagrams, drawing many intersecting wires can be tricky, particularly when you are going back to make sure you drew correctly. Visualizations of wire intersections could make complex diagram creation easier for users. In addition, by utilizing domain knowledge, sketch interfaces could help spatial arrangement by inferring how wires should be connected. One way in which we might further explore these interfaces is examining how often users make errors with the different types of interfaces; perhaps naturalness will lead to fewer errors and a greater feeling of satisfaction. Another research avenue to explore is looking at another type of diagram creation task (i.e. synthesis), where the interfaces might perform differently in terms of ease of use, speed, and naturalness.

Further research into understanding user perceptions of pen-based interfaces is needed and an understanding of why perceptual gaps exist needs to be reached.

6 Conclusion

We have presented a study comparing three pen-based interfaces designed for creating logic gate diagrams in order to determine whether naturalness is more important than speed and ease of use when accuracy is not a factor. The results of our study show that overall, participants created logic diagrams more rapidly with the hybrid interface than the drag-and-drop and sketch interfaces. However, many participants subjectively thought that the sketch interface was fastest and most easy to use. This result indicates that perceptions of speed may be more important than actual speed in task completion time in creating diagrams. Our study also shows that ease of use, naturalness, and speed are all important to users when using pen-based interfaces for creating logic diagrams. Research into pen-based interfaces should try to leverage the naturalness in sketch interfaces with the improvements in ease of use and speed found in other interface paradigms.

7 Acknowledgments

This work is supported in part by NSF CAREER award IIS-0845921 and NSF awards IIS-0856045 and CCF-1012056. We would also like to thank the members of the ISUE Lab for their support and the anonymous reviewers for their useful comments and feedback.

References

[1] J. Accot and S. Zhai. Beyond fitts' law: models for trajectory-based hci tasks. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, CHI '97, pages 295–302, New York, NY, USA, 1997. ACM.

[2] C. Alvarado and R. Davis. Sketchread: a multi-domain sketch recognition engine. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, UIST '04, pages 23–32, New York, NY, USA, 2004. ACM.

[3] G. Apitz and F. Guimbretière. Crossy: a crossing-based drawing application. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, UIST '04, pages 3–12, New York, NY, USA, 2004. ACM.

[4] J. N. Bott, D. Gabriele, and J. J. LaViola Jr. Now or later: An initial exploration into user perception of mathematical expression recognition feedback. In *Proceedings of the 8th Eurographics Symposium on Sketch-Based Interfaces and Modeling*, SBIM '11, pages 125–132. ACM, August 2011.

[5] B. Buxton. Artists and the art of the luthier. *SIGGRAPH Computer Graphics*, 31(1):10–11, Feb. 1997.

[6] A. S. Forsberg, A. Bragdon, J. J. LaViola Jr., S. Raghupathy, and R. C. Zeleznik. An empirical study in pen-centric user interfaces: Diagramming. In *Proceedings of the 5th Eurographics Symposium on Sketch-Based Interfaces and Modeling*, SBIM '08, pages 135–142. Eurographics Association, 2008.

[7] L. Gennari, L. B. Kara, T. F. Stahovich, and K. Shimada. Combining geometry and domain knowledge to interpret hand-drawn diagrams. *Computers and Graphics*, 29(4):547–562, Aug. 2005.

[8] W. I. Grosky, R. Zeleznik, T. Miller, A. van Dam, C. Li, D. Tenneson, C. Maloney, and J. J. LaViola. Applications and issues in pen-centric computing. *IEEE MultiMedia*, 15(4):14–21, Oct. 2008.

[9] T. Hammond and R. Davis. Ladder, a sketching language for user interface developers. *Computers and Graphics*, 29(4):518–532, Aug. 2005.

[10] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

[11] J. Jorge and F. Samavati. *Sketch-based Interfaces and Modeling*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[12] B. Kang and J. J. LaViola Jr. Logicpad: a pen-based application for visualization and verification of boolean algebra. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, IUI '12, pages 265–268, New York, NY, USA, 2012. ACM.

[13] S. Kollmansberger. Logic gate simulator. Computer program, June 2011.

[14] G. Kurtenbach and W. Buxton. User learning and performance with marking menus. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, CHI '94, pages 258–264, New York, NY, USA, 1994. ACM.

[15] J. J. LaViola, Jr. and R. C. Zeleznik. Mathpad^2: a system for the creation and exploration of mathematical sketches. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 432–440, New York, NY, USA, 2004. ACM.

[16] J. Lin, M. W. Newman, J. I. Hong, and J. A. Landay. Denim: finding a tighter fit between tools and practice for web site design. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, CHI '00, pages 510–517, New York, NY, USA, 2000. ACM.

[17] I. S. MacKenzie, A. Sellen, and W. A. S. Buxton. A comparison of input devices in element pointing and dragging tasks. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, CHI '91, pages 161–166, New York, NY, USA, 1991. ACM.

[18] M. M. Mano and C. R. Kime. *Logic and computer design fundamentals*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1997.

[19] M. Negulescu, J. Ruiz, and E. Lank. Exploring usability and learnability of mode inferencing in pen/tablet interfaces. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, SBIM '10, pages 87–94, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.

[20] R.-D. Vatavu, D. Vogel, G. Casiez, and L. Grisoni. Estimating the perceived difficulty of pen gestures. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part II*, INTERACT'11, pages 89–106, Berlin, Heidelberg, 2011. Springer-Verlag.

[21] P. Wais, A. Wolin, and C. Alvarado. Designing a sketch recognition front-end: user perception of interface elements. In *Proceedings of the 4th Eurographics workshop on Sketch-based interfaces and modeling*, SBIM '07, pages 99–106, New York, NY, USA, 2007. ACM.

[22] J. O. Wobbrock, A. D. Wilson, and Y. Li. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, pages 159–168, New York, NY, USA, 2007. ACM.

[23] S. W. Zamora and E. A. Eyjlfsdttir. Circuitboard: Sketch-based circuit design and analysis. In *IUI Workshop on Sketch Recognition, 2009*, Feb. 2009.