

In Search for an Ideal Computer-Assisted Drawing System

†Takeo Igarashi, ‡Sachiko Kawachiya, †Satoshi Matsuoka, †Hidehiko Tanaka

†Dept. of Information Engineering, ‡Dept. of Information Science

The University of Tokyo

Hongo 7-3-1, Bunkyo, Tokyo, Japan

+81-3-3812-2111 ext 7413

{takeo, tanaka}@mtl.t.u-tokyo.ac.jp, {sachiko, matsu}@is.s.u-tokyo.ac.jp

ABSTRACT Diagram drawing with conventional computer-assisted drawing(CAD) editors often tend to take considerable amount of time despite their seeming ease of use. We analyzed the problems of such systems focusing on the problem of cognitive overload, and observed that 1) the necessity of cognitive *planning* process in current CAD system causes the problems and that 2) reducing the overload can lead to fundamental improvement in overall drawing efficiency. We have conducted an experiment to verify these observations by comparing a typical drawing system and our prototype drawing system called *Interactive Beautification*, which combines the ease of freehand drawing and precision of traditional drawing editors by extracting various constraints in input strokes, and generating the desired diagrams automatically. Results show that significant amount of time is spent for cognitive *planning* process, and reduction of such planning time by *Interactive Beautification* can significantly improve the efficiency of CAD.

KEYWORDS CAD, Interaction technique, Cognitive workload, Constraint hierarchy, Beautification.

1 INTRODUCTION

Computer-assisted, Object-Oriented(OO) drawing editors are now used widely, allowing precise drawings not possible with traditional pen-on-paper drawing techniques. Furthermore considerable research and development have been conducted on creating new and innovative drawing systems resulting in those with abundance of functionalities. Despite such developments, drawing with such systems remains time-consuming and non-trivial, especially for novice or casual users.

The problem is that most research and development efforts on drawing systems have basically been 'feature wars', adding new drawing capabilities and interaction techniques, without deep, profound insights into their usability in practice. Instead we need studies to find "where does the fundamental problem exist" based on the close analysis of drawing processes on practical drawing systems.

Based on such a motivation, we propose an abstract model of drawing to explain the problems with computer-assisted drawing(CAD) systems focusing on the cognitive overload (Suchman, 87) (Norman, 86), and verify it through several experiments. In the model, we assume that the funda-

mental problem with CAD lies in considerable requirements imposed on the user's cognitive *planning* process. By *planning* we mean the cognitive process during drawing, where a user plans on the combination of the physical operations available in drawing editors to obtain the desired image. In addition, we confirm through an experiment with our prototype system that the reduction of such cognitive *planning* process can improve the efficiency of CAD.

Our prototype system frees the users of the cognitive *planning* process by automatically generating the desired diagram from freestrokes, but detailed descriptions of the salient technical features of the system itself is not the purpose of this paper; rather, we generalize on the effectiveness of the approaches to remove user's cognitive *planning* process from drawing editors.

This paper is organized as follows. First, we analyze the problems with CAD and propose an abstract model that captures the problems of excessive cognitive workload. Second, we conduct an experiment to quantify the workload. Third, we introduce a new drawing system where the objective is to reduce the workload, and conduct an experiment to quantify its effectiveness versus traditional OO-based

drawing editors. Finally, we conclude with the future directions of our the research.

2 A TASK MODEL OF DIAGRAM DRAWING

In this section, the problems of excessive cognitive *planning* workload in current drawing systems are analyzed and a task model to explain the problems is proposed.

2.1 Problems with Current Drawing Systems

In some informal studies, the following problems have been observed for drawing on typical OO-based CAD systems.

Problem 1: it takes considerable amount of time to figure out how to combine the functions of editor to draw desired diagrams.

Problem 2: it takes considerable amount of time for users to realize that she has come to a “dead end” with the current drawing “plan”, with which the desired diagram can never be obtained.

For example, to draw a line perpendicular to a slope, one can either a) count grids, b) duplicate the slope and rotate it 90 degrees, c) draw a horizontal and vertical lines and rotate them, etc. The user has to identify and select at least one of the strategies, requiring cognitive processing time. On the other hand, if the user fails to select an appropriate strategy (for instance, to draw a slope and a line separately and adjust the line with free rotation, which is often a wrong strategy because it is often difficult to obtain the exact desired angle with free rotation), a considerable amount of time is spent on ‘wasteful operations’.

We explain these problems as the effect of the overhead of the cognitive *planning* process. By *planning*, we mean the translation of image in one’s mind into physical operations, where image is the concrete visual image of a diagram to draw, and physical operations are the executions of various commands (draw lines, draw rectangles, copy, flip, rotate, etc.) or grid adjustment, etc. Problem 1 can be explained as the *planning* process causing additional overhead for each operation, and problem 2 can be explained as failure during the *planning* process resulting in wasteful operations.

2.2 An Abstract Model for Drawing Systems

To explain the problems and to clarify the structure of a drawing task, we propose a simple abstract model of diagram drawing with a typical OO-based editor (Figure 1). This model explains the interactions between the user’s mental image of the desired

result of drawing, and physical drawing operations the user performs during the drawing process.

Initially, the user has a visual image of the intended diagram in her mind. Then, to materialize the image on the editor, she must translate the visual image into a sequence of commands available on the editor (*planning* process)(A), and perform physical operations based on the planned command sequence. The operations are usually accompanied by direct manipulation of visual images on the display(C). Notice that the resulting diagram on the screen and the initial image in her mind has a visual representation, while the command sequence has a symbolic representation. That is, the user must go back and forth between the two representations.

For example, if a user tries to draw a tilted rectangle, the user first plans a series of operations such as creating a rectangle first, and then rotating it(A). Next, she executes the command operations physically, based on the *plan* by selecting the appropriate menu items(B). Then, she directly manipulates the drawn object to get the desired placement, size, and tilt(C). In this way, the drawing progresses repeating the sequence (A),(B), and (C). This model does not capture the entire aspects of diagram drawing, but is sufficient to explain the problems in CAD.

According to this abstract model, what we pointed out in the previous section are the problems pertaining to the *planning* process (A), i.e., the transition from visual representations to symbolic representations. However, many research activities in diagram drawing systems focused on the process (B) and (C), while the *planning* process has not been extensively explored. Our experiments show that large overhead is due to such a *planning* process, and imply that the removal of the commands and *planning* process dramatically improves drawing efficiency.

2.3 Expert Users and Casual Users

Recently, Bhavnani,S.K. and John,B. extensively studied the improvement of CAD productivity, focusing on the mental model of users (Bhavnani, 96). The main observation is that CAD experts are employing suboptimal strategies in drawing diagrams. Then, they identified a set of actual strategic knowledge for using CAD effectively and concluded that teaching these strategies to CAD users may improve the overall productivity.

On the other hand, the basis of their cost analysis is the number of operations expert users have to perform in the editor to complete a particular task, in a accordance with KLM (Card, 80) and GOMS

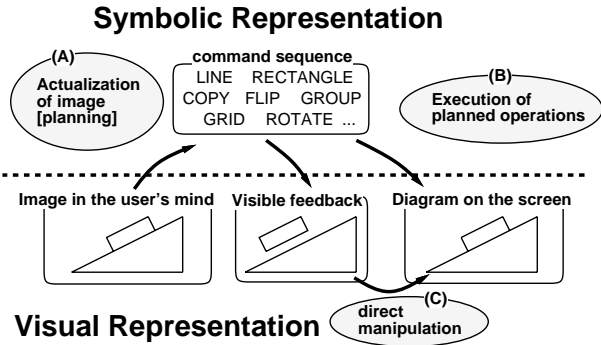


Figure 1: An abstract model of drawing editor

(Card, 83) (Olson, 90). As a result, they do not cover the task model where casual users may spend extra time to find out the appropriate operations, or time being wasted for making and recovering from erroneous operations. Furthermore, their proposal, i.e., to make strategic knowledge explicit and teach it to users, may not be applicable to casual users. We next describe an experiment to quantitatively determine the overhead of planning for casual users.

3 AN EXPERIMENT TO ESTIMATE THE AMOUNT OF COGNITIVE WORKLOAD

3.1 Experimental Set-Up

In this experiment, we observe and video-record the users drawing a given set of sample pictures using a typical OO-based drawing editor.

Systems. The machine used is an Apple Powerbook, and the drawing software is CANVAS 3.5 by Deneba Software.

Subjects. 14 casual CAD users who are accustomed to typical window-based GUI, and have used drawing tools occasionally. They vary in their proficiency in using CANVAS.

Task. Each user is first given a brief tutorial of CANVAS and conducts a series of short exercises several times. Then the users are instructed in the following manner:

- To draw as rapidly and clearly as possible,
- To draw in a natural way, as he would with drawing programs he is accustomed to.
- There is no need to reproduce perfectly the given diagram, if the task appears to be too difficult.

Then the users draw the two given diagrams(Figure 2), verbally speaking out the task he intends to perform. No time limits or feature restrictions are imposed so as to simulate standard drawing activities.

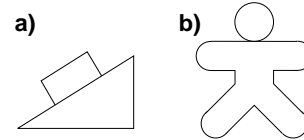


Figure 2: Diagrams used in the experiment

We analyze the video recording of each subject, dividing and categorizing time spent for the following behaviors into three categories: the time spent not pressing the mouse button(**NON**), the time spent during which the objects are directly operated upon with a mouse(**OBJ**), and the time during which the menus are selected with a mouse(**MENU**). Next, we add explanations of the user's activities to each unit by referring to the video recordings. For example, to draw a line, a user first moves the cursor to a tool menu (**NON**), and selects the 'line' palette(**MENU**). After the selection, the cursor moves to the starting point (**NON**), and then to the end point(**OBJ**).

3.2 Results

Based on the basic data collected as described above, we perform detailed analysis of the diagram drawing. The analysis consists of two parts. One is *local level analysis*, where the time consumed during each individual operation is calculated without considering its respective meaning or role in the total drawing sequence. From this analysis, we can examine detailed information on how quickly subjects performed each step. The other is *global level analysis*, where the underlying meaning of each step is analyzed with respect to its role in the total drawing sequence. This analysis is expected to reveal the nature of trial and error sequences during the drawing process.

Local Level Analysis

Figure 4 shows the histogram of the categorized operation steps classified by the time they required. First, we observe that some operation steps consume considerable amount of time. If each step consists of a simple physical operation which does not require cognitive processing, the time consumed during each step should stay relatively constant. However, the histogram shows that the time for each step varies widely, where there are some steps that exceed 6-8 seconds, indicating that extra time is consumed for cognitive processing.

We investigated such time-consuming steps by analyzing the videotaped verbal protocols and observed some common phenomena:

- A user fails to obtain his desired result because of inappropriate grid settings(Figure 3); and further-

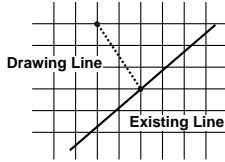


Figure 3: An example of time consuming operation: the user wants to draw a perpendicular line, but it is impossible in this grid setting. Further, it takes time for him/her to realize it.

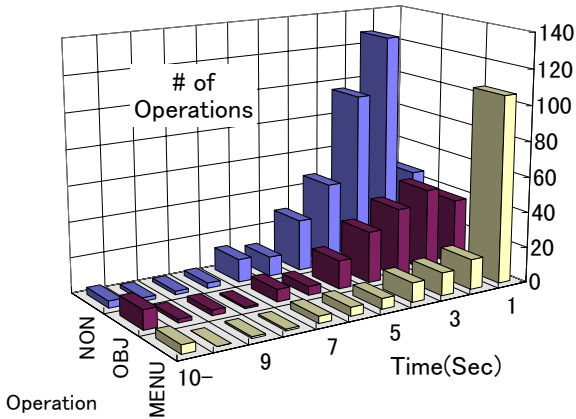


Figure 4: The number of operations categorized by the time they consumed: this figure shows that some operations are much more time consuming than others, which can be seen as the result of effect of cognitive processing.

more fails to realize this fact and repetitive futile attempts are made to draw.

- The mouse pointer makes aimless movements and/or momentarily pauses, failing to make any meaningful contributions to the intended drawing. Rather, time is seemingly spent on *planning*, e.g., wondering whether to draw the slope first or the box first in Figure 2a.

We regard these phenomena as the manifestation of cognitive workload because these time losses are not directly relevant to physical operations or the final result. In the following, we calculate an estimate of the percentage of the time spent on cognitive processing during the entire drawing process.

In order to obtain the basis of the estimate, we performed a series of experiments, where subjects are specifically instructed to perform a certain operation, such as to select a rectangle in the tool menu and draw a rectangle, or to select an object and ‘flip’ it along the x-axis by selecting the flip command in

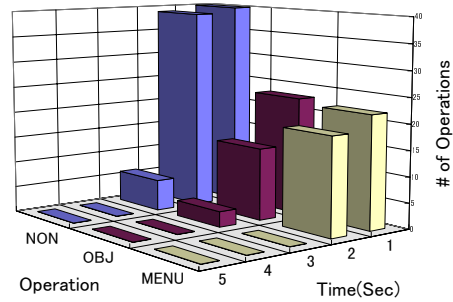


Figure 5: The number of operations categorized by the time they consumed in a task where no planning is necessary: all operations terminate within 2-3 seconds.

the menu bar, etc., thereby eliminating their cognitive planning workload. The results are shown in Figure 5, which show that time spent for such physical operations is short and uniform, taking approximately 2-3 seconds. This result is in accordance with the results reported in previous work (Card, 83)(Olson, 90).

Based on this result, the time spent for cognitive planning process is calculated by basically subtracting the physical operation time from the total drawing time. The actual calculation is based on the following equation:

$$[\text{Eq.1}] \text{ cognitive processing time} = \sum_{[x \in \text{all steps}]} (\max(0, t(x) - \text{threshold}))$$

where

$t(x)$ = time spent for step x ,

threshold = assumed average time required for each physical operation.

In the equation, we make a simplification that the assumed average time for each physical operation remain constant. This is because with the current experiments, the distinction between the physical operation versus cognitive planning cannot be clearly determined. So instead, we assume a constant average threshold for physical operations, and any time spent for each operation exceeding this threshold is assumed to be spent for cognitive planning process. Based on the baseline experiments in Figure 5, we varied the threshold value between 2-4 seconds. The result is shown in Table 1 — here, we estimate that roughly 20% ~ 40% of the time is spent for cognitive planning process.

	Assumed threshold (Sec)		
	4.0	3.0	2.0
Figure 2a	22.50%	30.90%	45.10%
Figure 2b	17.6%	26.7%	41.5%

Table 1: The ratio of time required for cognitive processing to the total drawing time: this figure is the result of the calculation of cognitive processing time based on [Eq.1].

Global Level Analysis

In this analysis, the role of individual operations in the context of the entire drawing processes is examined. Then the ‘wasteful’ operations, whose effects are not reflected in the final results, are extracted, and the total time spent for such wasteful operations is calculated.

For example, a subject in the experiment tried to draw Figure 2a by drawing a rectangle and a triangle separately, and then rotating the rectangle so that its edge will be parallel to the slope of the triangle. However, he could not do so because it is not possible to specify a rotation angle explicitly in CANVAS other than integer values (e.g., if the necessary rotation angle was 35.5 degrees while it is only possible for the user to specify either 35 or 36 degrees.) We observed that it took considerable amount of time for the subject to become aware of the nature of the problem. After numerous futile attempts, the subject gave up and deleted the rectangle. Here the time used for the operations from the generation to the deletion of the rectangle is calculated as ‘wasteful’ time.

Figure 2b posed even more difficulties for most users. For example, it was difficult to draw the legs, and achieve proper symmetry; either the limbs would not properly join, the half-circle signifying the foot could not be merged, or the leg lines would end up not being parallel, depending on the user’s initial plan. This seemed to surprise and frustrate the users, because of the apparent simplicity of the figure, plus the fact that minor local adjustments usually failed.

Figure 6 shows the results. The amount of wasteful time varies among subjects, but the existence of wasteful operations, and its effects are clearly observed. (One may notice that there is not a strong correlation between time for fig 2a and fig2b. This is because the amount of wasting time greatly changes depending on accidental failures rather than subject-specific reasons which are constantly observed.) Thus, we claim that, it is not only essential for the user interface of drawing systems to help in decreasing the

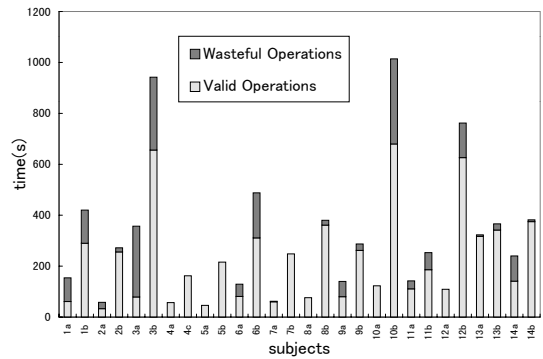


Figure 6: The time used in the wasteful operations: numbers 1a-14a indicate the drawing of Figure 2a by the 14 subjects and numbers 1b-14b that of Figure 2b accordingly: this figure shows that some subjects spend considerable time on wasteful operations.

amount of time required for an optimal drawing procedure, but also to prevent users from making a plan which would actually not lead to the desired results.

4 REQUIREMENTS FOR AN IDEAL DRAWING SYSTEMS

The experiments showed that considerable overhead is due to the cognitive *planning* process for casual users of OO-drawing systems. Thus, alleviating the overhead of *planning* process can fundamentally reduce the overhead. In other words, helping users to translate images to operations and freeing user of this translation are some of the most essential approach to improve drawing systems.

Based on the proposed abstract model of Figure 1, this means that improvement of the process (A) is critical for improvement of the overall drawing systems. However, most research focused mainly on process (B) and (C), whereas only few has explicitly intended to improve (A).

4.1 Previous Efforts on Improving the Drawing Systems

Previous constraint-based layout approaches such as (Bier, 86), (Bouma, 95), (Nelson, 85), (Borning, 81), (Sutherland, 63), etc. allow drawing precise diagrams without wasteful operations during (C), but most systems required the users to state necessary constraints explicitly, which can increase the user’s cognitive burden during (A). Adding new features and functionalities, which are often the strategies with

commercial editors, can decrease the number of required operation steps in (B) or possibility of mistakes in (C), but the added complexity also can increase the cognitive burden in (A). Programming-by-example approaches (Kurlander, 92) (Lieberman, 92) (Maulsby, 89) could be regarded as decreasing the burden of (B) and (C), but in practice could impose considerable overload when the user has to carefully plan what are the ‘right’ set of examples to present to the system. Gesture based systems (Apte, 93) (Zao, 93) (Rubine, 92) (Gross, 96) can free the users of menu selection in (B), but gestures in these systems correspond to the individual commands in OO-based systems, and the problem of planning remains. Beautification systems (Pavlidis, 85) intend to free users of fine placement of objects in (C), but the initial drawing process would be basically the same as in OO-based systems.

The work of Saund et al. (Saund, 94) can be viewed as helping the users during process (A). Their system, PerSketch, allows the system to understand the perceived image of the drawing in the user’s mind and translate it into a system representation. However, it only handles the movement of sketch objects, and does not deal with precise drawings.

5 INTERACTIVE BEAUTIFICATION: an Experimental System to Reduce the Planning Overhead

We are currently developing an experimental drawing system called *Pegasus*, which assists the planning process (A) of user drawings. We call the interaction technique *Interactive Beautification*, which reduces the overload of the planning process by removing editing commands entirely from diagram drawing process. Here, we investigate how the system could improve the drawing task.¹

We first give an overview of the interactive beautification in *Pegasus*. First, the user draws an approximate shape of his desired diagram with free stroke (Figure 7(A)) using a pen or a mouse. Then, the drawing system infers the desired precise diagram based on the *perceptual constraints* between the input stroke and surrounding context, and displays the multiple possible beautified candidates on the screen (Figure 7(B)). In this diagram generation, the system automatically detects higher level constraints such as similarity or symmetry. Among the several possible candidates, the system returns the

¹As we have mentioned, the evaluation of the system itself is *not* the purpose of this section, but confirmation of the validity of our proposal is the current objective.

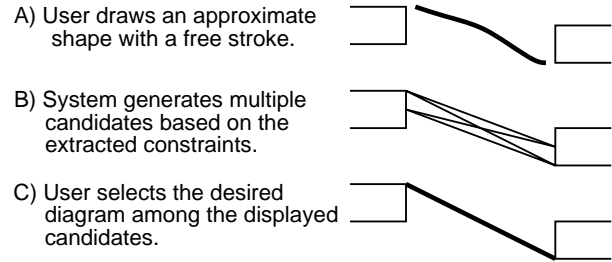


Figure 7: Interaction of the proposed system

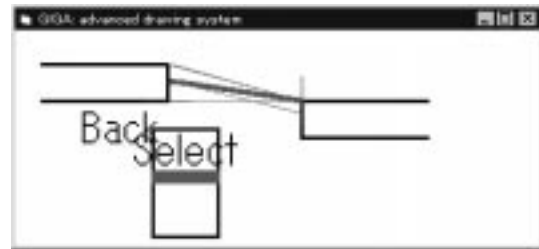


Figure 8: Sample image of the prototype system on Windows 95: the user is selecting a desired candidate using a slider.

most confident one as the primary candidate. If the user is satisfied with the candidate, he can proceed on to the next stroke without any additional operations (Figure 7(C)). Otherwise, the user can select the desirable one using a slider that appears near the cursor (Figure 8). During the selection, the system displays the constraints satisfied by the currently selected candidate visually, which makes rapid and precise interaction possible. Overall, users can draw a precise diagram without abstract manipulations such as gridding or rotation.

Interactive Beautification resolves the ambiguity of input strokes and allows the usage of complex constraints that were not easily specifiable in previous OO-based drawing systems. We construct a constraint hierarchy consisting of a set of weighed constraints extracted from the input stroke and surrounding context. Multiple candidates are generated by changing the strength of the constraints in the constraint hierarchy (Sannella, 94). The use of a constraint hierarchy is superior to other ad hoc strategies in that it effectively generates plausible candidates by separating constraint solving and constraint abstraction. (Detailed description of this algorithm and prototype system will be given in another paper.)

At a glance, the system may seem similar to ex-

isting sketch-based interfaces including commercial ones such as Apple Newton, GO's Penpoint, and other freestroke mode in drawing softwares (SmartSkech, Corel Draw, etc.). These systems accept freestroke input and convert them into vector segments. They also automatically connect nearby segments, straighten lines, and recognize geometric primitives (circle, rectangle, etc.). However, *Pegasus* is different in that it handles complex, global constraints such as parallelism, symmetry, or similarity, which requires global analysis of the drawing, besides basic constraints such as connecting adjacent points. Considering such higher-level constraints is essential in freeing the users of the planning process. For example, if one wants to draw a symmetric diagram precisely in existing systems, he must draw one half of the desired diagram, and use explicit commands (copy, flip, and move) to complete the drawing. If any editing occurs, such command actions have to be repeated every time. In contrast, desired symmetry could be automatically accomplished in our system without little or no additional operations.

5.1 Implementation

The prototype system is being developed with Microsoft Visual Basic and Visual C++ on Windows 95. Sample drawings are shown in Figure 8. The currently implemented constraints are coincidence of edge points, horizontal / vertical alignment of edges, equivalence of segment length, parallelism / perpendicularity among segments, and horizontal and vertical symmetry. Time required for generating multiple candidates is about 100 msec (IBM PC/AT, Pentium 75MHz), sufficient for interactive drawing. Constraints for curves are currently being worked on.

5.2 Experiment

Using the prototype system described above, we performed an experiment to investigate whether reduction of *planning* process could really reduce the cognitive workload and improve efficiency of drawing processes.

In the experiment, the same 14 subjects as the previous experiment were instructed to draw the diagram shown in figure 1a, and the drawing processes were recorded and analyzed as had been with the previous experiment. Subjects were given a brief explanation of the prototype system, and practiced with the system for approximately one minute.

Figure 9 shows the time required for each subject to complete the drawings. On the average, drawing time for our prototype system was approximately 43% of that for the OO-based system. This indicates

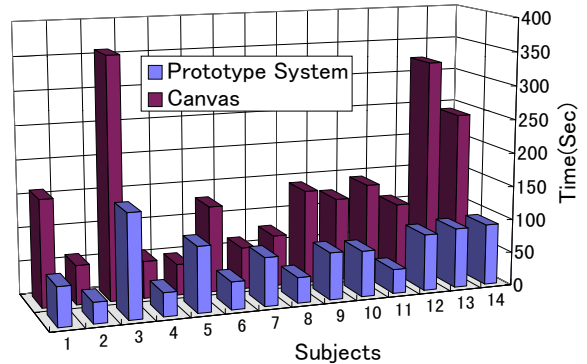


Figure 9: Time required to draw figure 2a in CANVAS and in the prototype system: per average, drawing in the prototype system requires only 43% of the drawing time in CANVAS.

	Assumed threshold (Sec)		
	4.0	3.0	2.0
Figure 2a	11.8%	19.3%	34.3%

Table 2: The ratio of time required for cognitive processing to the total drawing time in the prototype system.

that the time required in our system is shortened and more stable. The result of local level analysis is shown in Figure 10 and Table 2. The ratio of cognitive processing time is reduced to 12% ~ 34% compared with 17% ~ 45% for that of CANVAS. Investigation of Figure 10 shows that **SELECT** operations (selection of a candidate by a slider), is the sole operation that takes some time to complete, and other operations (**NON** and **STROKE**) finish quickly. Furthermore, future improvements of both the interface for **SELECT** and the associated constraint solving algorithm may improve the time for **SELECT**. Lastly, we observed that the wasteful operations due to the misunderstanding of the system behavior and wrong planning are greatly reduced. The quality of the drawings were not quantified in these experiments, but no conclusive differences in quality between the two experiments were observed.

The result shows that reduction of planning time achieved in the prototype system improved the efficiency of diagram drawing; it remains to be seen, however, whether our proposed *Interactive Beautification* is applicable to a wider range of applications.

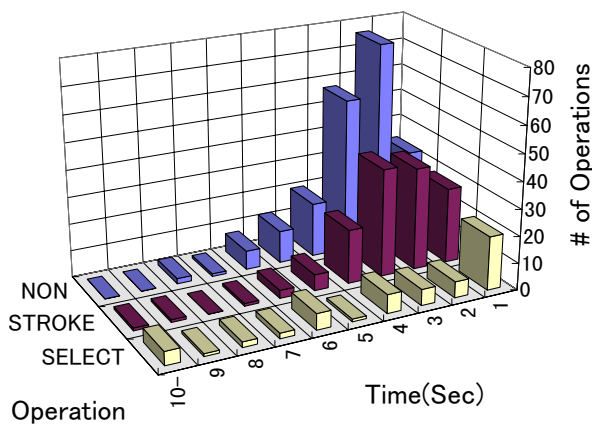


Figure 10: The number of operations categorized by the time they consumed in the prototype system: STROKE means the free stroke input, and SELECTION means the candidate selection by the slider.

6 CONCLUSION

We have conjectured that, for casual users, the necessity of cognitive planning process results in major overhead in current OO-based drawing systems, and interaction techniques to fundamentally reduce planning overhead improve the overall drawing process.

Our experiments to verify the conjecture showed that *planning* process consumed about 17% ~ 45% of the total drawing time, and that wasteful operations consumed up to 78% of total drawing time. By contrast, our prototype *Pegasus* drawing system with *Interactive Beautification*, exhibited significantly reduced *planning* process time, or approximately 12% ~ 34% of the total drawing time. Furthermore, due to the reduction of wasteful operations, the overall drawing time is reduced to 43% of the drawing time with conventional OO-based drawing system.

As a future work, we must perform a more detailed and comprehensive comparative studies among other types of drawing systems, using a more extensive set of drawing samples. We also need to make our drawing system more robust by incorporating other kinds of constraints and their solvers especially curves, and furthermore improve the **SELECTION** interface for multiple candidates.

References

- Apte, A., Vo, V., and Kimura, T.D. (1993) "Recognizing Multistroke Geometric Shapes: An Experimental Evaluation," *Proc. of UIST'93*, pp. 121 - 128.
- Bhavnani, S.K., John, B.E. (1996) "Exploring the Unrealized Potential of Computer-Aided Drafting", *Proc. of CHI'96*, pp. 332-339.
- Bier, E.A., Stone, M.C. (1986) "Snap Dragging", *Computer Graphics*, Vol.20, No.4, pp. 233-240.
- Borning, A. (1981) "The Programming Language Aspects of ThingLab, A constraint-Oriented Simulation Laboratory", *ACM Trans. on Program. Lang. Syst.*, Vol.3, No.4, pp.353-387.
- Bouma, W., Fudos, I., Hoffman, D., Cai, J., Paige, R. (1995) "Geometric constraint solver", *Computer Aided Design*, Vol.27, No.6, pp. 487-501.
- Card, S.K., Moran, T.P. (1980) "The Keystroke-Level Model for User Performance Time with Interactive Systems", *Communications of the ACM*, Vol.23, No.7, pp. 396-410.
- Card, S.K., Moran, T.P., Newell, A. (1983) "The Psychology of Human-Computer Interaction", Lawrence Erlbaum Associates, Inc. .
- Gross, M.D., Do, E.Y. (1996) "Ambiguous Intentions: A Paper-like Interface for Creative Design", *Proc. of UIST'96*, pp. 183-192.
- Kurlander, D., Geiner, S. (1992) "A History-Based Macro by Example System", *Proc. of UIST'92*, pp. 99-106.
- Lieberman, H. (1992) "Dominoes and Storyboards: Beyond "Icons on Strings"", *Proc. of VL'92*, pp. 65-71.
- Maulsby, D.L., Witten, I.H., Kittlitz, K.A. (1989) "Metamouse: Specifying Graphical Procedures by Example", *Computer Graphics*, Vol.23, No.3, pp.127-136.
- Nelson, G. (1985) "Juno, A Constraint-based Graphics System", *Computer Graphics*, Vol.19, No.3, pp. 235-243.
- Norman, D.A., S.W.Draper (1986) "User Centered System Design", Lawrence Erlbaum Associates, Inc. .
- Olson, J.R., Olson G.M. (1990) "The Growth of Cognitive Modeling in Human-Computer Interaction Since GOMS", *Human-Computer Interaction*, Vol.5, pp. 221-265.
- Pavlidis, T., VanWyk, C.J. (1985) "An Automatic Beautifier for Drawings and Illustrations", *Computer Graphics*, Vol.19, No.3, pp. 225-234.
- Rubine, D. (1992) "Combining Gestures and Direct Manipulation", *Proc. of CHI'92*, pp.659-660.
- Sannella, M. (1994) "Analyzing and Debugging Hierarchies of Multi-way Local Propagation Constraints", *PPCP '94*, pp. 63-77.
- Saund, E., Moran, T.P. (1994) "A Perceptually Supported Sketch Editor", *Proc. of UIST'94*, pp. 175-184.
- Suchman, S.A. (1987) "Plans and Situated Actions: The Problem of Human-Machine Communication", Cambridge Univ. Press.
- Sutherland, I.E. (1963) "Sketchpad: A Man-Machine Graphical Communication System", *Proc. of Spring Joint Comp. Conf.*, No.23, pp.329-346.
- Zao, R. (1993) "Incremental Recognition in Gesture-Based and Syntax-Directed Diagram Editors", *Proc. of InterCHI'93*, pp. 95-100.