Design and evaluation of the alignment stick*

Roope Raisamo
Kari-Jouko Räihä
Department of Computer Science
University of Tampere
P.O. Box 607 (Pinninkatu 53B)
FIN-33101 Tampere, Finland
{rr,kjr}@cs.uta.fi

ABSTRACT

Object alignment is one of the basic operations in drawing programs. Current solutions provide mainly three ways for carrying out this operation: either by issuing an alignment command, or by using direct positioning with the help of gravity active points, or by making use of constraints. The first technique has limited functionality, and the other two may be mysterious for a novice. We describe here a new direct manipulation tool for alignment. We show that while direct manipulation helps to make the tool intuitive, it has through iterative design evolved into a tool that also offers functionality not found in current commercial products. We also report on an experiment in which we compared the ease of use, intuitiveness, learnability, and efficiency of alignment menus, palettes and the alignment stick. Finally, we discuss our experiences in the two-handed control of the alignment stick and report on other interesting findings in our experiment.

KEYWORDS: drawing programs, alignment tools, direct manipulation, menus, tool palettes, two-handed interaction, iterative design, alignment stick, empirical tests

* This work was partially supported by the Academy of Finland (Project 38118, Grant 52936).

1. INTRODUCTION

Direct manipulation has for more than a decade been one of the preferred styles of interaction. Drawing programs were among the first applications that made heavy use of direct manipulation. Recent developments in drawing programs have concentrated on intelligent techniques, such as the use of constraints. The basic direct manipulation behavior has remained more or less the same.

A closer look at drawing programs shows that the real world analogy, on which direct manipulation is based, has not been used nearly as much as possible. In this paper, we shall focus our attention on the alignment tool. We shall develop a new interaction technique that is more direct and understandable than the traditional ways of carrying out the same operations.

In addition to the interaction technique itself, the process used in its development is also of interest. Through iterative design, we were able to create a tool that, besides being easy to use and intuitive, also offers new functionality.

This work had two driving forces. One was the examination of the new interaction idea. Another was related to a larger project that is carrying out research on multimodal interaction in general, and two-handed interaction in particular. Two-handed interaction increases a user's degrees of freedom, when the user can manipulate two closely related properties of the alignment tool at the same time [Chatty, 1994a, 1994b]. However, two-handed interaction is a technique that needs very careful planning, as the results of previous work in the area point out [Kabbash *et al.*, 1994]. Because of the many functions provided by our tool, the use of two hands for controlling it proved useful for a smooth handling of the various properties.

Of course, object alignment is not one of the major problems or obstacles in current user interfaces. However, while new applications bring new problems into focus, it is also important to sit back and analyze carefully the established methods and techniques. At the operating system level this is already common practice: window manipulation, menu presentation and scroll bars have been studied empirically and improvements to common solutions have been suggested. Our work focuses in a similar way on one common primitive of drawing programs.

The rest of the paper is organized as follows. First, we will present previous work related to our subject. We then discuss our first solution to the alignment problem. This solution had several problems, both conceptual and related to the implementation. We describe the problems and their solutions, and present our current version of the tool. We then describe the experiment that we have carried out. Next we discuss the results and analysis of the data. We conclude with a discussion of the lessons learned and work that needs to be done.

2. PREVIOUS WORK

Alignment is a basic property of current drawing programs. One frequently needs to have objects on the same level either horizontally or vertically. The objects are usually aligned by their center points or by the bounding box of the selected objects.

It is hard to track down the history of alignment tools in drawing programs. They were introduced to a larger user population through commercial products such as Apple's MacDraw [Apple, 1984], but had existed earlier in various research prototypes and in Xerox's workstations [Baecker and Buxton, 1987]. Although much of the development of Xerox Star was based on careful user testing [Bewley *et al.*, 1983], the early development of alignment tools has not been documented.

2.1. Command Based Alignment Tools

The most common way to align objects uses a two-step procedure:

(1) select the objects to be aligned, usually by pointing and clicking with the mouse; and

(2) issue the appropriate alignment command.

The second step of this procedure does not make use of direct manipulation, thereby differing from many other operations typically found in drawing programs. While our main interest is in studying whether other techniques could outperform the traditional command based approach, it is illustrative to study briefly different possibilities used in issuing the alignment command in step (2) above.

A popular technique is to place the alignment commands in menus, as in the Micrografx Draw for Windows program in Figure 1. A problem with menu commands is that they are usually textual, and thus provide no visual cues to the user. While horizontal and vertical alignment may be evident concepts to an expert user, the same does not hold for novice users, who typically need to think of the orientation of the alignment. In fact, our empirical tests and informal observations of novice users indicate that they often confuse vertical and horizontal alignment, particularly in aligning by the center points of the objects.

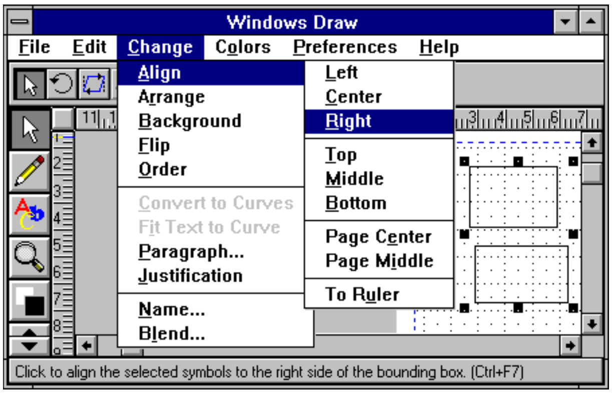

Figure 1: Traditional menu based alignment commands in Micrografx Draw for Windows.

Another possibility for issuing the alignment command is to use an alignment palette that provides alignment operations in graphical buttons. This method gives a user visual hints of what an operation would do if the user selects it. However, most palettes are so small that it may sometimes be quite difficult to see the meaning of the buttons. Figure 2 shows the alignment palette of ClarisDraw for Macintosh. In a new version of Micrografx Draw small icons have been added to menu options, which can help in choosing the right menu command.

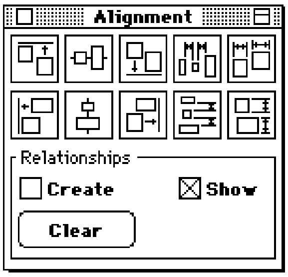

Figure 2: Alignment palette in ClarisDraw for Macintosh.

Visual cues are important, not only to make the choice of operations more intuitive, but to explain how the objects will move. The mere fact that the right sides of two objects are aligned does not indicate whether the leftmost object will move to the right, or the rightmost object to the left, or whether perhaps they will both move the same distance towards each other. This is something that has to be simply learned in menu based systems, but is directly visible in the palette.

The third method used in commercial products is to have only one alignment command in the menus, and to use a dialog box for providing additional parameters. This approach is used, e.g., in Adobe Illustrator for Macintosh (Figure 3).

The dialog box approach is obviously the slowest of the three for carrying out a basic alignment operation. On the other hand, it becomes possible to illustrate the effects of the operations, typically using some sample objects (Figure 3). Although these are not related to the actual selected objects, they may be helpful in teaching a novice the meaning of the operations. More importantly, it is possible to combine alignment in both dimensions into the same operation, a property that is not shared by either of the previous methods.

The dialog box of Figure 3 shares a conceptual problem with the hierarchical menus in Figure 1: the operations are always listed vertically, and the user has to map the command names to actual operations without visual cues that would support the mapping. Rotating one set of radio buttons so that the choices would be horizontal, matching the direction where the objects move, would already be an improvement [Foley *et al.*, 1990].

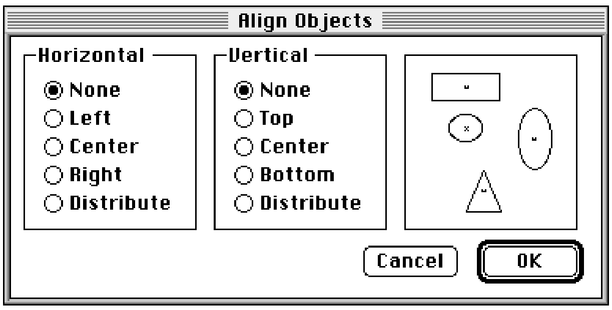

Figure 3: Alignment dialog in Adobe Illustrator 5.5 for Macintosh.

The visibility of palettes and the two-dimensional alignment with dialogue boxes can be combined into a single tool, as is the case with, e.g., Macromedia's FreeHand [Williams, 1998].

2.2. Advanced Techniques

In addition to being somewhat cumbersome to use, command based alignment tools offer only limited functionality. Advanced techniques for more complex alignment tasks have therefore been developed.

For a novice, the most natural way to align objects is simply to drag them to their desired positions. Such positioning is, of course, not very precise, but the technique can be enhanced by applying “gravity”. The drawing area can be equipped with gravity active points, and once an object is dragged close enough to such a point, it is snapped exactly to the point. To align a group of objects using this technique, each object still has to be dragged in position separately.

The most common form of gravity active points is a grid which can be turned on or off. Grid based alignment tools typically use a fixed grid, where the user is allowed to control only the density of evenly spaced grid lines. Adobe Illustrator goes one better: it provides so-called guide objects which can be positioned and moved manually by the user. Figure 4 shows three guide objects: one vertical ruler, one horizontal ruler and one rectangular guide object. The guide objects are drawn with dotted lines.

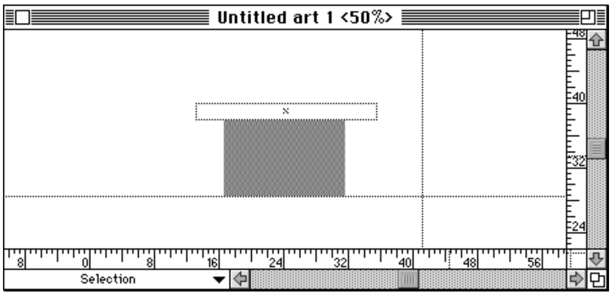

Figure 4: Three guide objects in Adobe Illustrator for Macintosh.

The power of gravity has been taken to a new level in a technique called snap-dragging [Bier and Stone, 1986; Pier *et al.*, 1988; Bier, 1989], where the user can draw new alignment objects and define their gravity active points. This allows precise positioning along a rich set of curves, not just rectangular objects. The metaphor for alignment objects is that of draftsmen using ruler and compass techniques for precision drawing. A somewhat similar technique has been implemented (and patented!) in the commercial drawing program DrawingBoard [Ashlar].

Grids and other gravity based techniques are direct: positioning is caused by the user’s direct actions. Another powerful, but indirect, technique makes use of constraints usually specified by the user. Constraint based tools relieve the user from fine grain positioning of objects. Instead, the system solves a set of constraints and automatically produces a transformed picture that satisfies the constraints. Such tools have a long history, going back all the way to Sketchpad [Sutherland, 1963].

Grids and constraints together provide so much power that to a novice, it may seem like magic. For grids alone, it may be hard for novices to understand their behavior or even to realize their existence [Springett and Grant, 1993; Springett *et al.*, 1993; Sutcliffe and Springett, 1992]. Still, many authors [Dieberger, 1995; Tognazzini, 1993] have recently proposed an increased use of magic as the way to create the right illusions and to make the user feel more at home with a powerful interface. In the case of alignment tools, however, we feel that the operation is in most cases so simple that it should be realizable with (almost) no magic. The maxim “Simple things should be simple; complex things should be possible” was attributed to Alan Kay in [Lipkie *et al.*, 1982]. While command based alignment tools do not make complex things possible, advanced techniques tend to achieve their power by making simple things harder, at least for a novice.

2.3. Direct Manipulation and Metaphors

The phrase “direct manipulation” was coined by Ben Shneiderman in 1982 [Shneiderman, 1982]. According to his definition, a direct manipulation application should:
- present the objects visually with an understandable metaphor;
- have rapid, complementary and reversible commands;
- display the results of an action immediately; and
- replace writing with pointing and selecting.

The concept of direct manipulation was carefully analyzed by Hutchins, Hollan, and Norman [Hutchins *et al.*, 1986], who concentrate on two key issues:
- short distances, both from goals to the physical system (“gulf of execution”) and from the physical system to goals (“gulf of evaluation”); and
- direct engagement.

The drawing programs discussed above do present the drawing objects visually, thereby providing the basis for direct manipulation. But it is also clear that in most of them the alignment operation is not visualized and does not give a feeling of direct engagement.

What is the reason for this? Is the alignment operation impossible to visualize? Certainly not. We shall present our solution later, but for now, think of this for a while: if you were asked to align a set of real world objects lying on your office desk, how would you proceed?

Metaphors are an essential part of human thinking and communication. That is why they are very important in direct manipulation interfaces, too. A proper selection of the metaphor can clearly help the learning and use of a computer system. Erickson [1990] and Halskov Madsen [1994] have listed issues that should be taken into account while designing a suitable metaphor. We have used these principles in the development of our alignment tool.

It is well known that metaphorical interfaces have their limitations: sooner or later the analogy breaks. Still, it is somewhat surprising that so little discussion on the current state of drawing programs can be found in the literature. The basic operations are based on direct manipulation; why is it so common to resort to commands when carrying out more advanced operations? Gravity based techniques use direct manipulation, but they extend the dragging operation to help in alignment, instead of providing a direct manipulation tool for the operation. For instance, one of the axioms in Alan Cooper’s popular book [Cooper, 1995] is: “A rich visual interaction is the key to successful direct manipulation”.

To an expert, this may not seem as much of a problem. However, studies of CAD programs show that productivity is hampered partly because the possibilities offered by the programs are used poorly. Users tend to follow the habits learned in manual drawing. Bhavnani and John [1996] suggest that the desirable practices should be taught to the users. Here we shall explore the other alternative, making the tools better fit the existing practices.

The system that is in spirit closest to ours is DrawingBoard [Dukic and Benest, 1991] (not to be confused with the commercial system mentioned above that has the same name!), developed by Davor Dukic and Ian Benest in the University of York, UK. This system introduced a computer-generated counterpart for the ruler that is used to draw straight lines. However, this ruler is used in a bitmap based paint program while drawing on the screen: there is no alignment operation or tool in the program.

Of course, it may be that even if a natural visualization for the alignment operation can be found, it may not prove efficient in use. We intend to show that through careful design, an intuitive, functionally powerful and efficient tool can be created.

3. THE DESIGN CYCLE

We started our work by considering the metaphor that would be appropriate for aligning objects. In this first stage we experimented with physical objects to guide us in the design.

Next, we developed the first prototype of a simple object-oriented drawing program called R2-Draw. It provides us with a convenient environment for exploring new interaction techniques. When the first implementation of our tool was ready, we ran a few informal user tests in our usability laboratory to see how the tool would behave in the hands of other users.

Several problems emerged, causing a redesign of many details (such as moving the center alignment property from a property of the tool into a property of the aligned objects), but the metaphor seemed to be intuitive. We tried to correct the previously found problems in the current implementation of the tool. We also added some new functionality that had not been implemented in the first prototype due to time restrictions.

In the following we first describe the initial design, and then elaborate on the role of two-handed control. The current version of the tool is then discussed in detail, followed by the description and results of an empirical study that compared different alignment techniques.

4. THE INITIAL DESIGN

The first major design decision was to choose an appropriate metaphor for aligning objects. We decided to adopt a metaphor that would be as close to the real world as possible. This way the user can have an advantage of the previous experience in similar tasks with real objects.

We chose a ruler as the metaphor for our alignment tool, because many people are familiar with it and may have used it to align, or at least to push away, miscellaneous items found on their desks. And, even if they hadn't done it earlier, this action should be quite intuitive. Moreover, the ruler has a scale that can be used to measure the distances between objects during the alignment operation. The ruler is not used to draw straight lines, as in DrawingBoard [Dukic and Benest, 1991], because object-oriented drawing programs already have rubber band style drawing tools. We differentiate our tool from the usual use of the ruler by calling it a stick instead of a ruler. The name *alignment stick* refers to the property of the ruler which is its primary use in the alignment operation.

We also considered the situation in which some objects could be pulled with the stick instead of being pushed. This added functionality would have complicated the metaphor. Since our main targets were novice users and our main goal was an intuitive tool, we decided against providing this mode.

Our real world based metaphor proved to be very useful since it allowed us to test the metaphor even before a single line of code was written. Our observations with a couple of bricks and one physical ruler showed us some of the potential problems related to the metaphor. To solve these, we used a pragmatic approach [Carroll *et al.*, 1988] in our metaphor design. According to this approach, if there clearly is a property of the metaphor which would be an obstacle to smooth functionality, it can be changed to act in a useful way. Following this approach, we chose to leave out physical effects between objects. One example of these is the fact that no solid object can move through another solid object, which would make overlapping objects very difficult to handle. Also, if an object is pushed in the real world with a physical stick that is not parallel to the pushed side of the object, the object will eventually rotate so that it is lined with the stick – another feature that was judged undesirable.

There are two other exceptions from the real world metaphor: the length of the stick can vary and the objects can be locked. These exceptions were made to facilitate the manipulation of only a subset of the objects. If you use a ruler to push objects on the desk, all of them will presumably move when they touch the stick. However, in a drawing program, some of the objects may already be where you want them to be. You don't want to spoil your previous work by pushing all of the objects. We therefore allow the user to change the length of the stick any time and to lock an object, so that it will not be affected by the stick. Each object has its own state (locked or unlocked) that can be changed from its pop-up menu.

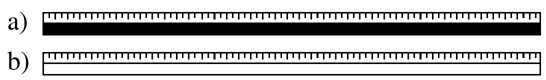

Figure 5: The alignment stick in the active (a) and inactive (b) state.

The alignment stick is shown in its two operational states in Figure 5. The lower stick is in the inactive state, displayed when the user moves the pointer over an object with the mouse button not pressed. The upper stick is in the active state which means that when the stick is pushed, it aligns every unlocked object that it touches. The stick is implemented as a large mouse cursor, which changes to the normal pointer when moved outside of the drawing area.

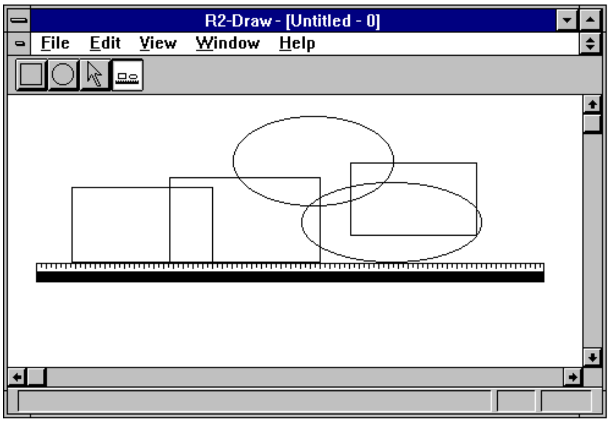

Figure 6: The first implementation of the alignment stick. The stick is moving up and has aligned three of the objects by their bottom sides.

The first implementation of the alignment stick is presented in Figure 6. The stick is activated and pushed upwards. Three of the objects have already been aligned by their bottom sides. The stick had two operating modes: border alignment and center alignment. In the center alignment mode all the objects were aligned by their center points until the stick was switched back to the border alignment mode.

Figure 7 shows the alignment stick in its typical use. First, the stick is in the inactive state (a). Next, the stick has been moved upward and activated. The stick has already touched the lower oval, which has moved a little upward (b). Finally, both of the ovals are aligned by their bottom sides with a single action (c).

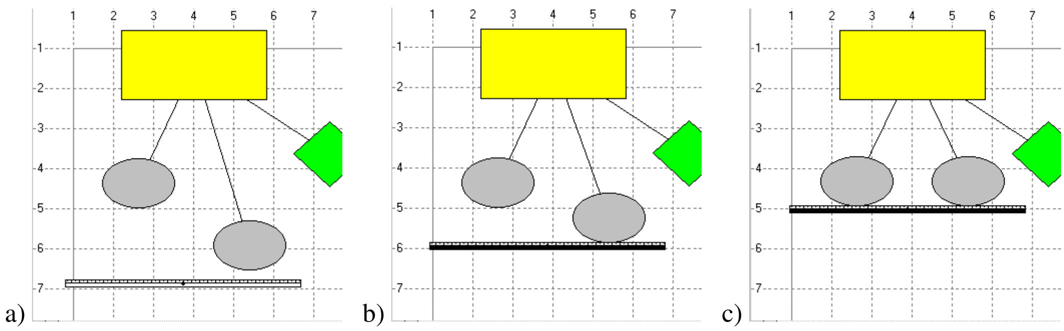

Figure 7: The alignment stick in use: a) The stick is inactive. b) The stick is activated and moving up. c) The stick has reached its intended position and the ovals are vertically aligned by their bottom sides.

Our approach to the alignment operation solves the problem of visual cues we presented in our discussion of the command based alignment tools. The problem was that the user had no way of knowing how the objects will move when they are aligned with a command. With the alignment stick the objects are moved while they are pushed. The user will see all the time how the objects are moving. We are using real time animation to show the current state of the drawing during these alignment operations. The user is also not likely to try to align the objects in the wrong direction (horizontal or vertical), because everything is expressed visually.

5. TWO-HANDED CONTROL

The alignment stick is a versatile tool. This can make it difficult to use with just one pointing device. This was one of our main reasons for implementing it with a two-handed technique. A brief discussion of two-handed interaction follows.

Two-handed interaction has not yet been used to its full potential in human-computer interaction. A well-known early experiment in the area was carried out by Bill Buxton and Brad Myers in 1986 [Buxton and Myers, 1986]. This experiment already showed that using two hands in parallel can clearly speed up the work and make learning individual tasks faster. One remarkable result of the experiment was that despite the sequential training of two operations, positioning and scaling, they were used in parallel by 6 of the 14 subjects without any suggestions by the researchers.

Stéphane Chatty [1994a, 1994b] has composed several guidelines for two-handed interaction. He divides two-handed systems into systems that use independent interaction, parallel interaction and combined interaction. In independent interaction the two hands are used simultaneously for two entire, distinct tasks, while in parallel interaction the tasks are equally important and somehow related. Combined interaction is the most sophisticated and preferable technique, where the two parallel tasks support each other and have a common goal. Kabbash *et al.* [1994] have pointed out that combined interaction is the most understandable and probably the only form of two-handed interaction that is clearly better than traditional one-handed sequential interfaces. Such an implementation also exists for the snap-dragging technique (see Figure 14 in [Bier *et al.*, 1994]).

We decided to use our primary pointing device (mouse) to control the position of the stick and the secondary pointing device (trackball) to accomplish actions that support the primary function of pushing objects to the desired position. The secondary action that we included in the first implementation was changing the width of the stick. This is done by rotating the trackball and can be accomplished while moving the stick with the mouse, resulting in combined interaction. The reason why these operations can be done in parallel is that they are used to accomplish the same goal and so the user need not divide his/her attention between many tasks, as would be the case in independent or parallel interactions. The choice of the trackball as the second input device was based on the results of Kabbash *et al.* [1993], showing that a trackball was equally accurate with the non-dominant hand as a mouse and a tablet-with-stylus. Furthermore, a large trackball is steady and this way overcomes some of the limitations (described in [Annett *et al.*, 1979; Guiard, 1987]) of the non-dominant hand.

By using two input devices we achieved smooth control of the alignment tool despite the fact that it has so many functions. The tool moves directly as a mouse cursor and is activated by pressing the left mouse button. It can be switched between the horizontal and vertical directions with the middle mouse button. The width of the stick can be changed with the trackball. Rolling the ball upwards or to the right increases the length and rolling downwards or to the left decreases it. The first implementation used the second input device only for this operation.

Following Chatty's [1994a] suggestion that everything should be possible without the second hand, when the right mouse button is pressed the mouse works as a substitute of the trackball. This behavior enables the use of the program while another hand or device is for some reason not available.

The user could switch between the border and center alignment modes by double-clicking the left mouse button or by issuing the command with a graphical button. Our user tests showed that it was difficult for users to remember that the center alignment mode existed, and even harder to remember how to switch to it.

6. THE REVISED DESIGN

We encountered a couple of problems while testing our first implementation with users. One major problem was that it was quite difficult to align and move objects exactly to the desired position. This was a consequence of our decision to not use any magic, but just intuitive direct manipulation. When the user tried to align the objects at a given position, the objects were frequently pushed too far. To fix this, the user needed to use another alignment operation and push the objects back a little. But since the objects were of different size, the previous arrangement was lost during this corrective operation. That is why still another alignment operation was needed to accomplish the goal.

We made several improvements to the tool to correct the problems of the first implementation. First, we added some magic: the common snap to grid operation. When applied with our alignment stick it functions so that when the stick is close to a line in the grid, the pushed objects are snapped to this line. This helps with those alignment operations that can use the grid, but it still does not solve the previous problem with arbitrary alignment operations.

Another way to correct the problem of pushing the objects too far is to switch to the pointer tool, select the previously pushed objects and drag them back to the intended position. As you can imagine, this was quite a complex and time-consuming operation and therefore was not preferred by the users. We were convinced that we should not add any more functionality to the tool itself. Therefore our solution at the time was to add an easier way to switch between the alignment and pointer tools. We decided to assign the left button of the trackball to this operation. The button could be pressed any time during the alignment operation and as long as it was pressed, the user could manipulate the objects with the pointer tool. When the button was released, the alignment stick was selected again. To make things easier, the latest pushed objects were automatically selected during this temporary use of the pointer tool.

We also eliminated the double clicking transition to the center alignment mode. In fact, we removed the center alignment mode from the stick in order to simplify the tool. The first implementation of the tool had too much hidden functionality that we wanted to make more explicit. We accomplished this by taking some of the functionality out of the tool and moving it to the objects, an approach that is similar to the one in [Houde, 1992]. There, too, the same basic operation behaves differently with various objects, and this is reflected in the way that the objects are displayed. In our case, we decided that center alignment should not be a property of the tool, but a property of the object that was aligned with the tool.

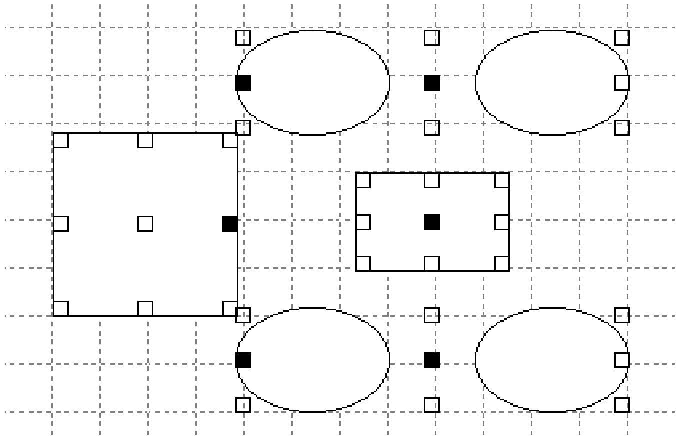

Figure 8: The alignment point setting tool in use. The user can click in the rectangles that represent points that need to be active during the alignment operation. The use of this tool is optional, but gives the user flexible alignment options. There are four different groups or individual objects with their own control points in the picture.

Normally the stick aligns the objects by their borders. However, it is possible to set so-called *alignment points* for each object, and when they have been set, the alignment is made using these points, not the borders. Alignment points are set using a separate tool, the use of which is shown in Figure 8. The user can click in the nine small rectangles, and set any combination of the alignment points. The most commonly used of these is the center point that allows carrying out the center alignment in the same way as the border alignment. Different alignment points can be selected in different objects, resulting in the kind of alignments that are not at all possible with the menu and palette commands: for example, the user could align a center point of one object with the right border of another. If several objects are grouped, they behave as one object and have only one set of alignment points.

We also gave the user more freedom in choosing the alignment direction. The stick can now be rotated to any angle in addition to the previous horizontal and vertical directions. The rotation is accomplished by the trackball when the user has selected free angle alignment with the middle mouse button. Simply rotating the trackball left or right rotates the stick around its center. After this addition, the alignment stick can be used just like the real one, only in two dimensions and remembering the few exceptions that we made to the ruler metaphor. Figure 9 summarizes the controls of the alignment stick.

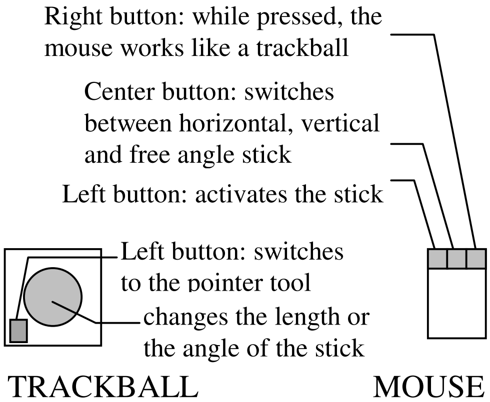

Figure 9: Controls of the alignment stick in our experiment.

Figure 10 presents the current implementation where the user is drawing an ER diagram. The rectangles have been aligned by their top or right sides. The group of ovals and the leftmost rectangle are centered horizontally. From left to right and top to bottom, the tool palette has the following tools: pointer tool, oval tool, rectangle tool, line tool, rounded rectangle tool, diamond tool, the alignment stick (chosen) and a tool for setting the alignment points. The buttons in the toolbar are used to manipulate files and the clipboard. In the upper right corner, 90° is the current angle of the stick.

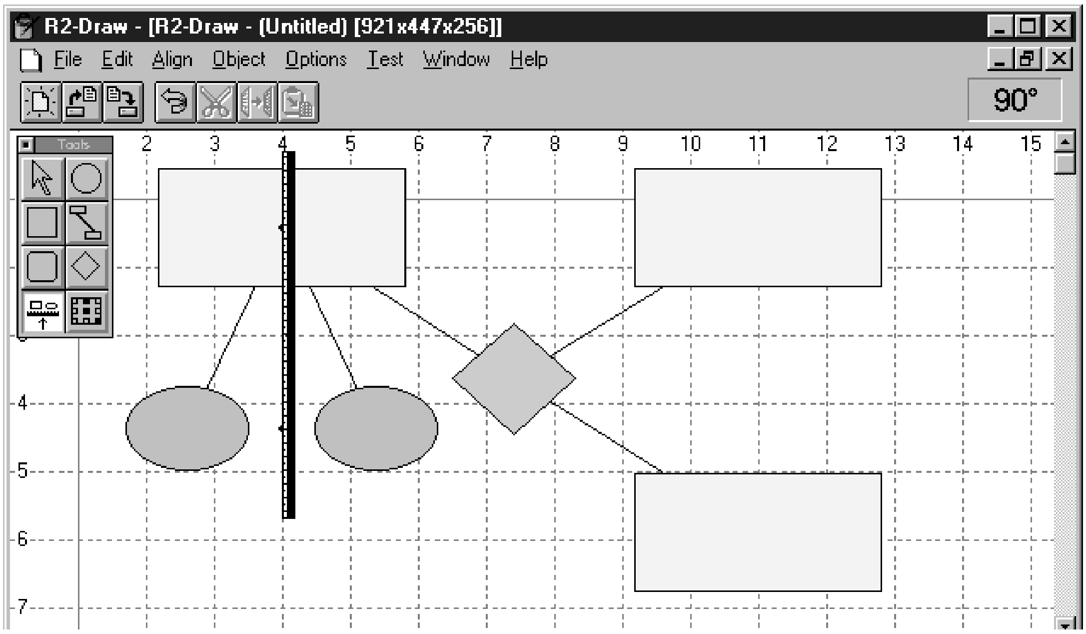

Figure 10: The implementation of the R2-Draw program that was used in this study. The user is aligning objects horizontally by their center points.

7. AN EMPIRICAL STUDY

We designed an experiment to compare the alignment stick to common alignment menus and palettes. The experiment was carried out as follows. First, we planned the tasks and other arrangements. Next, we performed six pilot tests and refined the tasks and other details for the final testing. Finally, the actual testing took place. This section describes the arrangements of our experiment, which is analyzed in the next section.

Training

It was difficult to decide the amount of training that should be given to the test subjects. If we trained them too much, we would loose any indication of the relative intuitiveness of different techniques. We were convinced that some training was necessary in order to compare these techniques, since all of the users had used some kind of menus and palettes, but had not seen the alignment stick before. Also, not all of the users were familiar with command based alignment tools.

In the beginning of each test we gave a general overview of the R2-Draw program (Figure 10), in which all three alignment methods are implemented. We decided to give the subjects a basic training for each method just before the method was tested. This way all of them had the necessary information when each method was being tested, but they did not have to remember all of the information during the whole experiment.

Tasks

We had four types of tasks. Each subject carried out the first three tasks in slightly different forms with all the three alignment methods. The last task was more versatile than the others, and allowed the subjects to choose any combination of alignment methods they wished. The motivation for the first three task types was to be able to compare the speed, accuracy, and suitability of the different methods for different tasks. The fourth type was intended to indicate personal preferences for using the tools in different alignment tasks.

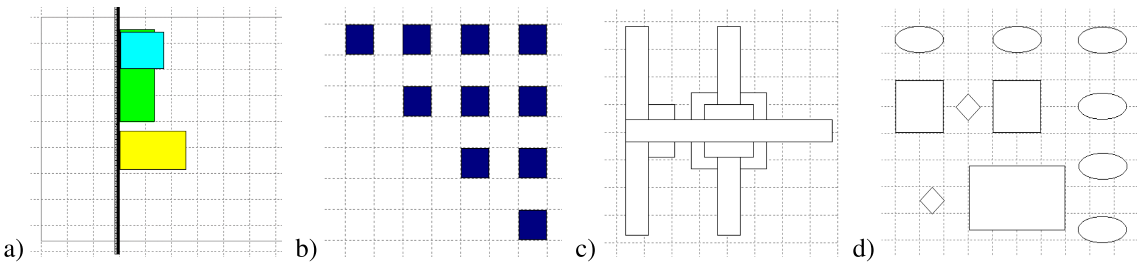

Figure 11: The goals of the four different tasks in the experiment. The alignment stick can be seen in picture a.

The first task was the simplest and involved aligning all three predrawn objects (Figure 11a) to the fourth grid line in each direction (left, right, top, bottom). In the second task, ten identical squares were scattered around the drawing window. The goal is presented in Figure 11b. The third task required aligning many overlapping objects and was more complex (Figure 11c). In the last task (Figure 11d) the user could choose any combination of alignment tools, and he or she also had to create the objects.

Subjects

The subjects were students of an introductory course in computing in our university. We grouped them in three categories based on their computer knowledge: novices, intermediate users, and experienced users. Novices had little or no experience of computers before the course. Intermediate users had 2-6 years of experience in computing and some experience in using the drawing programs. Experienced users had 2-12 years of experience in using computers and drawing programs. The subjects were unpaid, and their major field of study was diverse, ranging from economics to computer science, and from foreign languages to history studies. There were 12 female and 10 male subjects. The median age of the subjects was 20, ranging from 19 to 53.

Details of the Experimental Procedure and Setup

The experiment was carried out in our usability laboratory, and every trial was videotaped for further analysis. There was always at least one supervisor, who trained the user and was there to help the user out of a dead-end situation, should one occur. Otherwise the supervisor only watched and did not interfere in the task of selecting the right operations and tools.

The tasks were printed on separate sheets of paper. The users did not know the remaining tasks before the sheets were presented to them upon completion of the preceding tasks. The users were asked not to keep the sheet in their hands to remove this unnecessary obstacle for using two hands at the same time.

The test was essentially divided in six separate sessions:

1. general overview and training,
2. menu based alignment,
3. palette based alignment,
4. using the alignment stick,
5. using all the needed tools in task 4, and
6. filling an evaluation sheet and answering questions.

The second, third, and fourth sessions included training of the current method, and the order of them was varied so that all the six combinations were used equally often within the three subject groups.

8. RESULTS

This section is based on the subjective ratings that each subject gave after the test, and some objective measures recorded during the tests.

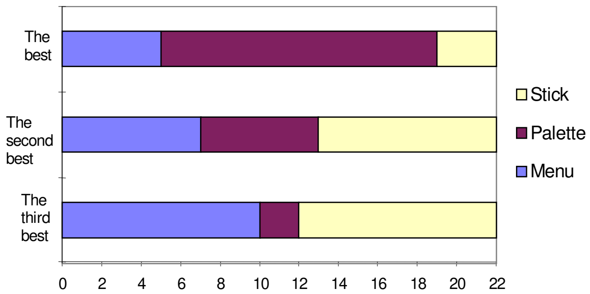

Figure 12: Personal preference of different alignment methods.

Figure 12 summarizes the overall ranking of the different tools by the subjects. The palette was for most subjects the preferred method of alignment, mostly because it was always visible, and presented the operations visually in buttons.

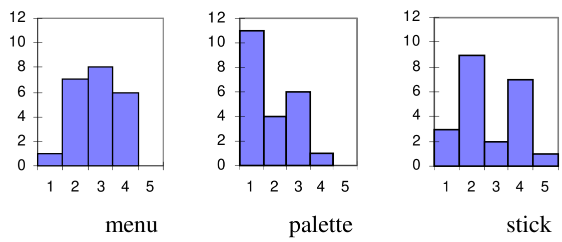

Figure 13: Rating of the naturalness of different alignment methods (1=best, 5=worst)

It is interesting to note that the ratings of the naturalness (Figure 13) follow a really different distribution with all three alignment methods. The menu seems to be a method that caused the smallest amount of reaction, and only one subject gave it the best rating (1). The palette was considered as the most natural. The alignment stick, however, seemed to divide opinions in two distinct groups: the most common rating was 2, but rating 3 was given by only two subjects. Another common rating was 4. It is interesting to see how this pattern can be observed within the three subject groups. We divided the ratings in three groups: positive (1-2), neutral (3), and negative (4-5). The ratings can be seen in table 1.

	positive	neutral	negative
experienced	5	0	2
intermediate	5	1	2
novice	2	1	4

Table 1: Subjective ratings of the naturalness of the alignment stick.

It can be seen that even if the experienced and intermediate users preferred the palette, they had a positive attitude towards the stick. The novices, surprisingly, found the stick less natural. This was mostly because they thought that the stick needed too precise handling. However, when the users were asked to select the method that was the easiest to use, they gave answers that are shown in Figure 14. More than 40% of the novices preferred the stick when assessing the ease of use. Within the other groups the palette was clearly superior.

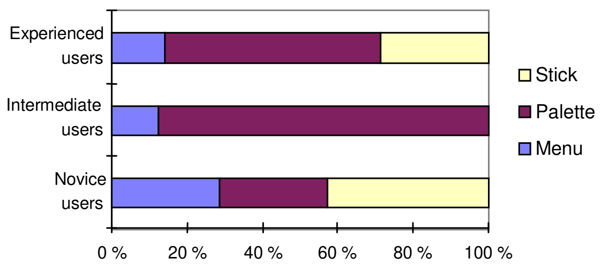

Figure 14: The rating of the ease of use of the different alignment methods.

Performance

Times for performing the tasks were measured by analyzing the videotapes. Here we discuss the most interesting findings. Figure 15 summarizes the mean completion times and shows how many operations were, on the average, used in the alignment process. It also shows the ideal number of operations were the tools used optimally. With the conventional alignment tools, these operations contain alignment commands, dragging operations and select operations. With the stick we have also counted changing the length or the orientation of the stick and selecting single alignment points. Figure 15 shows that time and the number of operations correlate. It is also clear that even though the most complex task 3 required only few operations, many more were actually used.

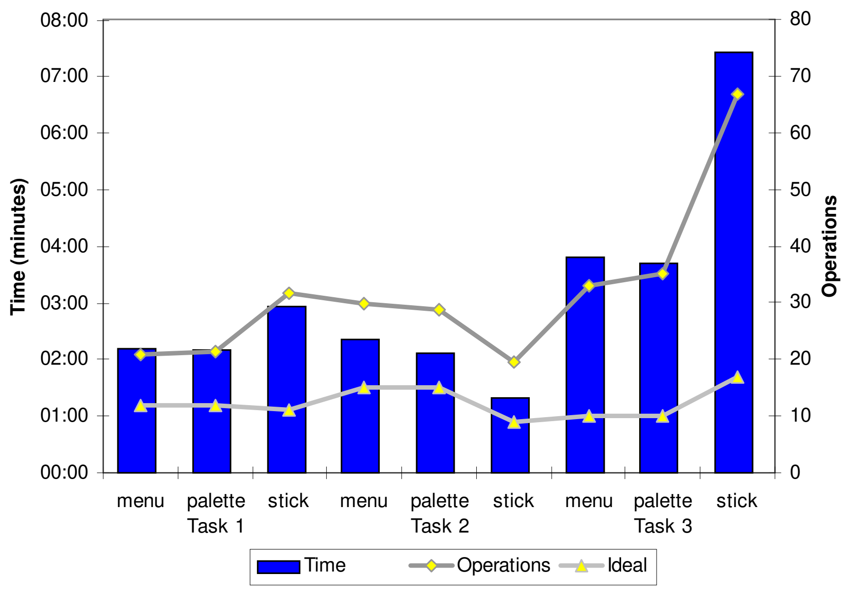

Figure 15: Mean completion times (in minutes) of the tasks using different alignment tools. The upper line shows the mean number of operations that was used in the alignment and the lower line the ideal number of operations in each task.

The difference in completion times was somewhat significant between the menu and the stick, and the palette and the stick in task 1 ($p<0.05$), but really significant in task 2 ($p<0.0005$) and task 3 ($p<0.00005$). As can be seen, the stick outperformed the other methods in task 2, but in task 3 the other methods were twice as fast as the stick.

In task 2, which involved aligning several non-overlapping objects, the stick was the fastest alignment method with all but 4 subjects. It was also interesting that the novices (mean time 1:17) actually outperformed the experienced users in task 2 (1:31), but the intermediate users were slightly faster (1:11). However, the differences between the groups were not statistically significant.

In task 1 the performance was partly hampered by it being the first task in using the tool and partly because many subjects pushed the objects too far needing additional corrective actions. In task 3 the benefit of removing the selection step was lost when the user had to choose alignment points for the objects – thereby actually making selection actions during the alignment. If the user wanted to preserve the alignment of some of the objects, a grouping action was also needed. This required selecting the objects and the grouping command from the pop-up menu.

It was expected that the tasks would differ in their difficulty and, consequently, execution time. Figure 16 shows the execution times per user for the three tasks when executed using the alignment stick. Black boxes represent task 3, gray boxes task 1, and white boxes task 2. We use the *x*-axis to indicate the experience of the subjects, with experience growing from left to right.

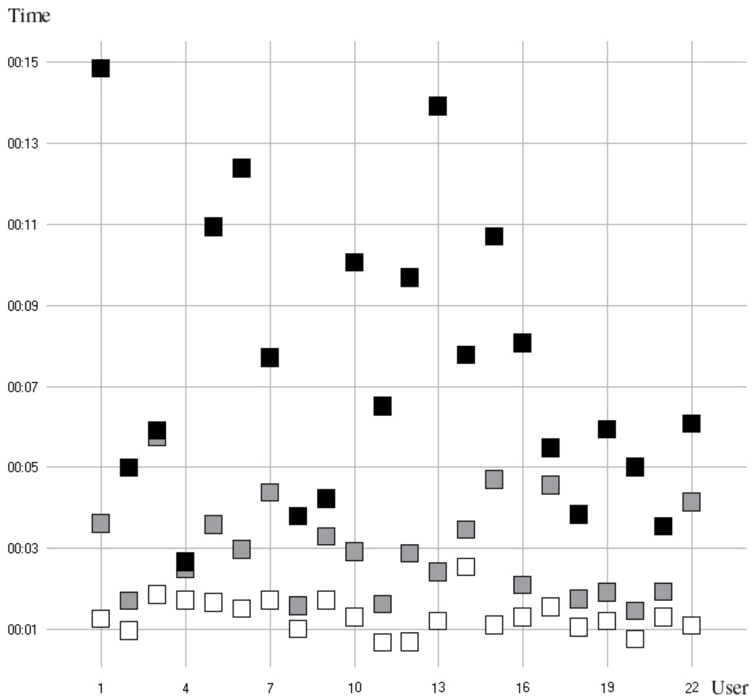

Figure 16: Scatter plot of the completion times of the tasks when carried out using the alignment stick.

It is noteworthy that *without exception*, task 3 required the longest time and task 2 the least amount of time. This was not the case with either the menu or the palette, for which the execution times did not show such a clear ranking between the difficulty of the tasks. One possible explanation for this is that users in all groups used the stick in a similar manner, whereas in the case of the menu and the palette the alignment tactics varied between users. It is also clear from Figure 16 that experience produces shorter execution times, even when the tool is new to the subjects.

Experience was not the only issue that had an effect on the users' performance. For instance, subject 4 from the left (one of the novices) was the fastest stick handler of the whole lot! In fact, in total execution time she was beaten by only one (experienced) user.

Another interesting result was the fact that variation among the users in total execution time of the tasks 1-3 was clearly higher with the stick than with the other methods. The variations with the stick and the menu can be observed in Figure 17, where black boxes denote tasks carried out using menus and white boxes denote tasks carried out using the stick. Subjects are ordered as in Figure 16. The variation with the palette was similar to the variation with the menu.

There is also no clear correlation between user satisfaction and execution times. For one experienced user the stick was faster than the other two methods, and he also used it more than the others in task 4. Still, he ranked the stick in third place in the overall ranking. Conversely, for one novice user the stick was the slowest tool, and yet she preferred it over the others and also used it extensively in task 4.

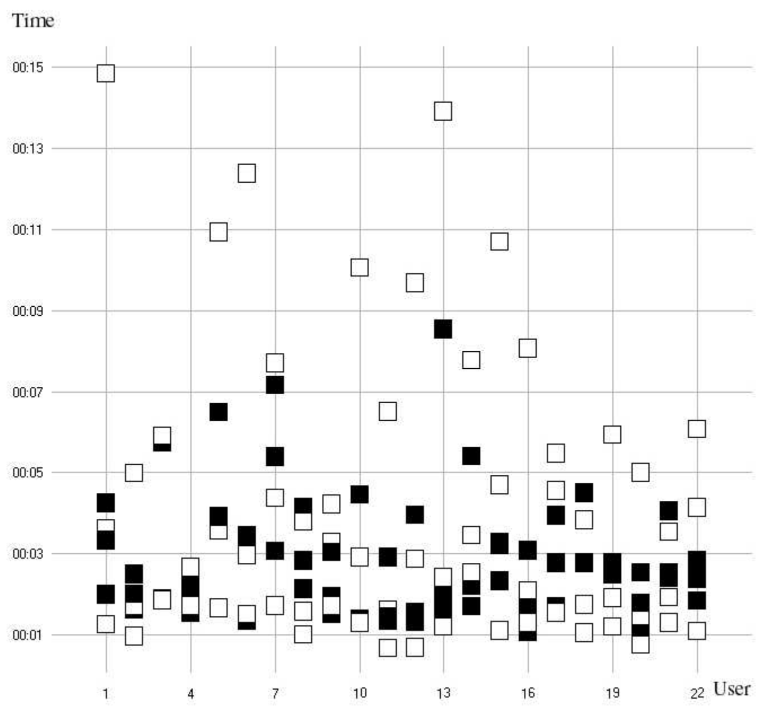

Figure 17: Total execution times for tasks 1-3 when using the menu and the alignment stick.

Intuitiveness

Our observations and interviews showed that all of the subjects understood the basic operation of the stick with their first try. In task 4, all but 4 users used the stick. Similarly, all but 4 users used the palette, too, whereas as many as 13 users did not use menus at all. This hints to the fact that menus are not an optimal solution for carrying out alignment operations.

Difficulties in understanding the use or manipulation of the stick were only noticed when the alignment points were needed. With the conventional alignment methods, many of the users had problems in deciding which menu command to choose, but most did not have problems with the palette. Alignment points seemed to diminish the intuitiveness of the stick. Their advantages would become more evident in more complicated alignment tasks, of which task 4 is a small example.

Errors

Functional errors

The intuitive way of aligning objects with the alignment stick removed conceptual errors in which the user aligns the objects in the wrong direction (horizontally or vertically). We noticed this behavior with both the menu and the palette with over half of the subjects. In more complex tasks (tasks 2 and 3) many novice users also tended to drag individual objects instead of selecting an alignment command. This resulted in a slightly wrong alignment that required fine-tuning at the end of each task in which it happened. However, the stick introduced new kind of errors that are related to it being a direct manipulation tool. Specifically, the users tended to push the objects too far, which required additional corrective alignment operations. After the tests we fixed this problem by having the stick act as a magnet that will stick to the objects as long as the user keeps it activated. This has turned out to be a working solution.

Accuracy

We measured the accuracy of task completion for task 1 by erroneous pixels offsetting from the goal position. The mean error in pixels was 0.88 pixels for the palette, 0.92 pixels for the menu, and 0.99 pixels for the stick. Paired 1-VA's showed no statistical significance for these differences (p-values between methods 0.42, 0.62 and 0.66).

It was striking, though, that this relatively good accuracy was achieved through careful manual operation, *not* by using the tools to their full potential. Most users carried out some manual dragging operations after they had executed their final alignment operation for a given task. Thus, in the majority of cases the fact that the objects really were aligned was not guaranteed by the software. This was particularly the case when the objects had to be aligned both horizontally and vertically. While the alignment tool was used for one dimension, many users resorted for manual operation in the other dimension.

Two-handed Control

Two-handed interface received high subjective ratings. Almost 80% of the subjects thought that it was beneficial, and 90% of the subjects thought that the trackball was suitable for the non-dominant hand. When carrying out the tasks, about 50% of the subjects used both hands in manipulating the tool, and 40% of them used them simultaneously. On the other hand, the total time of two-handed interaction was very small (measured in seconds). Longer studies that would allow the subjects to really get familiar with the tool are clearly needed to settle the issue more definitively.

9. DISCUSSION

Because of the very nature of the alignment stick, it removes one step in the alignment process when compared with command based tools. In command based alignment, the user has to select the objects to be aligned and then issue the command. With our tool, the selection is not needed in the basic operations. The only thing that the user has to do is to push the objects.

On the other hand, it should be pointed out that advanced gravity based techniques, such as snap-dragging, allow alignment along a rich set of shapes, whereas the stick in its current form only provides linear alignment. Furthermore, the stick aligns entire objects while snap-dragging can align at the level of individual control points, which means transforming the object that is manipulated. We have tried to achieve a balance between functionality and intuitiveness.

In the user tests it was interesting to observe that even though for us it was natural to consider alignment and movement as separate operations, this was not necessarily so for the users. We already pointed out that many users dragged objects even after aligning them with some other objects, thus breaking the alignment. Even more surprising was that alignment tools were used for movement. For the stick, this is natural – we often saw a very narrow stick used instead of dragging. However, we did not expect that the palette would be used for movement as well. For instance, suppose that users had to align objects by their left sides to a certain line, and suppose that the objects were initially on different sides of the target line. Then we observed several users to select all the objects, align them by their *right* sides (thereby moving them all on the same side of the target line), then align them by their left sides, and then drag them into position. It appeared that the users had first made up their minds on the desired dragging direction, and did not think whether there would have been a simpler sequence of operations to achieve the same result.

In general, the users found imaginative ways to use the tools. For instance, with the stick some users used the alignment points by turning them all off, so that the stick would not grab those particular objects. Users also suggested new possible alignment methods. One interesting suggestion was that after the selection of the objects that were to be aligned, the targets (e.g., the grid lines) could be active so that by clicking them, the target would snap the objects to itself.

In discussing the problems with direct manipulation, Hutchins, Hollan, and Norman [Hutchins *et al.*, 1986] state: "Direct Manipulation interfaces have problems with accuracy, for the notion of mimetic action puts the responsibility on the user to control actions with precision, a responsibility that is often best handled through the intelligence of the system, and sometimes best communicated symbolically." Our implementation of the alignment stick shows that indeed, a little magic is needed to handle precise positioning; but with this addition, direct manipulation provides the basis for a tool that is both intuitive and powerful.

Following our own goals of simplifying the alignment operations we came close to Chatty's suggestion that the magic should be disabled when two hands are at work [Chatty, 1994b]. He applied that principle to a line segment drawing tool, where an invisible magic hand normally holds the other end of the segment. In Chatty's work the magic was disabled so that the user could hold the other end with a secondary input device while dragging another with the primary input device. However, the magic that we added is not used in the same way as in the drawing of the line segments, but to help the user with alignment operations needing accuracy often difficult to achieve with human motor systems [Annett *et al.*, 1979; Guiard, 1987]. This decreases the need for strict concentration during the operation.

It was interesting to note that novice users were from the beginning as fluent with the new alignment tool as they were with the older tools, whereas users that had previous experience with drawing programs required a short time period (15-20 minutes) to feel at ease with the alignment stick. This is yet another indication of the simple fact that intuitive interfaces are based on learning and on previous experiences [Hutchins *et al.*, 1986]. Seasoned users take some time to "unlearn" old habits and customs. There is a need for a longitudinal study in which a group of users can really get used to the new way of aligning objects.

10. CONCLUSIONS

Careful design of the alignment tool gave us two valuable benefits: intuitive use and added functionality. This functionality developed even further during our iterative design process and informal user tests. Clearly, though, there was also a need to further develop the stick. The most important issue was getting rid of the most common error of pushing the objects past the goal. We have resolved this problem by having the stick work as a magnet that sticks to the objects as long as it is active. Also, the currently required grouping stage with the more complex drawings, as the one in task 3, forces the user to select objects during the alignment task. One idea that we will be experimenting with is applying the inductive groups [Olsen and Deng, 1996] which has potential for removing the grouping stage and therefore making the alignment stick more suitable for complex drawings.

The total lack of conceptual orientation errors supports the intuitiveness of our stick metaphor. Even though the novices found the stick less natural in the subjective ratings, they still managed to use it well, especially in task 2. The stick did not work well in task 3, which involved aligning many overlapping objects and required the use of alignment points and object grouping. Tasks 2 and 4 were close to the actual tasks that the user faces while using a diagram editor. Thus, our experiment shows that the alignment stick can be a useful tool in different diagram editors, but needs further development in order to be used in complex alignment tasks. Another notable result of the experiment is that the alignment menu and palette did not differ significantly in execution times, but the users strongly preferred the palette. Therefore, based on these results, we should favor palettes over menus in implementing the basic alignment operations.

The experiment also gave some support for the usefulness of two-handed interfaces. Half of the subjects chose to use both hands without any explicit request. Also, based on the subjective ratings, the large trackball that we have chosen for the non-dominant hand seems to be suitable for its task. Since the non-dominant hand was only used for a supportive action in the alignment stick, the small total time of two-handed interaction was inevitable.

An interesting general observation was that the customary separation of alignment and movement by dragging was not at all obvious to novice users. Perhaps new kind of interaction techniques could be developed where the basic operations are not so strictly separated as is customary today.

The alignment stick has been the base for an alternative way of drawing that is based on our stick metaphor. In our CHI'99 paper [Raisamo, 1999] we present these tools and how they can be used to draw creative drawings. The alignment stick is one of the tools in this metaphor.

REFERENCES

J. Annett, M. Annett, P. T. W. Hudson, and A. Turner (1979) The control of movement in the preferred and non-preferred hands. *Quarterly Journal of Experimental Psychology 31*, 1979, 641-652.

Apple Computer, Inc. (1984) *MacDraw Manual*.

Ashlar Inc., DrawingBoard. [http://www.drawingboard.com/info.html]

Ronald M. Baecker and William A. S. Buxton (1987) Case study D: The Star, the Lisa, and the Macintosh. In *Readings in Human-Computer Interaction, A Multidisciplinary Approach*, R. M. Baecker and W. A. S. Buxton (Eds.), Morgan Kaufmann, 649-652.

William L. Bewley, Teresa L. Roberts, David Schroit, and William L. Verplank (1983) Human factors testing in the design of Xerox's 8010 "Star" office workstation. *Human Factors in Computing Systems, CHI '83 Conference Proceedings,* 72-77.

Suresh K. Bhavnani and Bonnie E. John (1996) Exploring the unrealized potential of computer-aided drafting. *Human Factors in Computer Systems, CHI '96 Conference Proceedings*, ACM Press, 332-339.

Eric Allan Bier and Maureen C. Stone (1986) Snap-Dragging. *Proc. SIGGRAPH'86, ACM Computer Graphics, 20 (4)*, 233-240.

Eric A. Bier (1989) Snap-Dragging: Interactive Geometric Design in Two and Three Dimensions. Technical Report EDL-89-2, Xerox PARC.

Eric A. Bier, Maureen C. Stone, Ken Fishkin, William Buxton, and Thomas Baudel (1994) A taxonomy of see-through tools. *Human Factors in Computer Systems, CHI '94 Conference Proceedings*, ACM Press, 358-364.

William Buxton and Brad A. Myers (1986) A study in two-handed input. *Human Factors in Computer Systems, CHI '86 Conference Proceedings*, ACM Press, 321-326.

John M. Carroll, Robert L. Mack and Wendy A. Kellogg (1988) Interface metaphors and user interface design. In *Handbook of Human-Computer Interaction*, M. Helander (Ed.), North-Holland, 67-85.

Stéphane Chatty (1994a) Issues and experience in designing two-handed interaction. *Human Factors in Computer Systems, CHI '94 Conference Companion*, ACM Press, 253-254.

Stéphane Chatty (1994b) Extending a graphical toolkit for two-handed interaction. *ACM UIST '94 Symposium on User Interface Software and Technology*, ACM Press, 195-204.

Alan Cooper (1995) *About Face: The Essentials of User Interface Design*. IDG Books. [http://www.cooper.com/]

Andreas Dieberger (1995) On magic features in (spatial) metaphors. ACM SigLink Newsletter 4 (3). [http://www.lcc.gatech.edu/~dieberger/magic_features.html]

Davor Dukic and Ian Benest (1991) DrawingBoard: Adopting office metaphor for page composition. *Sun ´91 Conference Proceedings,* Sun UK User Group, Buntingford, UK, 227-239.

T. Erickson (1990) Working with interface metaphors. In *The Art of Human-Computer Interface Design*, B. Laurel (Ed.), Addison-Wesley, 65-73.

James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes (1990) *Computer Graphics: Principles and Practice*. Addison-Wesley.

Yves Guiard (1987) Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *The Journal of Motor Behavior 19 (4),* 486-517.

Kim Halskov Madsen (1994) A guide to metaphorical design. *Communications of the ACM, 37 (12)*, 57-62.

Stephanie Houde (1992) Iterative design of an interface for easy 3-D direct manipulation. *Human Factors in Computer Systems, CHI '92 Conference Proceedings*, ACM Press, 135-142.

Edwin L. Hutchins, James D. Hollan, and Donald A. Norman (1986) Direct manipulation interfaces. In *User Centered System Design*, D. A. Norman and S. W. Draper (Eds.), Lawrence Erlbaum, 87-124.

Paul Kabbash, I. Scott MacKenzie, and William Buxton (1993) Human performance using computer input devices in the preferred and non-preferred hands. *Human Factors in Computer Systems, INTERCHI '93 Conference Proceedings*, ACM Press, 474-481.

Paul Kabbash, William Buxton, and Abigail Sellen (1994) Two-handed input in a compound task. *Human Factors in Computer Systems, CHI '94 Conference Proceedings*, ACM Press, 417-423.

Daniel E. Lipkie, Steven R. Evans, John K. Newlin, and Robert L. Weissman (1982) Star graphics: An object-oriented implementation. *Proc. SIGGRAPH'82, Computer Graphics 16 (3)*, 115-124.

Dan R. Olsen Jr. and Xinyu Deng (1996) Inductive Groups. *ACM UIST '96 Symposium on User Interface Software and Technology*, ACM Press, 193-199.

Ken Pier, Eric Bier, and Maureen Stone (1988) An introduction to Gargoyle: an interactive illustration tool. In *Document Manipulation and Typography, Proceedings of the International Conference on Electronic Publishing, Document Manipulation and Typography (EP88)*, J. C. van Vliet (Ed.), Cambridge University Press, 223-238.

Roope Raisamo (1999) An alternative way of drawing. To appear in *Human Factors in Computer Systems, CHI '99 Conference Proceedings*, ACM Press.

Ben Shneiderman (1982) The future of interactive systems and the emergence of direct manipulation. *Behaviour and Information Technology 1*, 237-256.

M. V. Springett and A. S. Grant (1993) Interface semantics and users' device models: Identifying evaluation issues for direct manipulation design. *Proceedings of the HCI'93 Conference on People and Computers VIII*, 249-265.

M. V. Springett, A. S. Grant, and A. G. Sutcliffe (1993) Interface semantics and procedural knowledge: A study of novice understanding of MacDraw. *East-West International Conference on Human-Computer Interaction: Proceedings of the EWHCI'93*, Springer-Verlag, 241-256.

A. G. Sutcliffe and M. V. Springett (1992) From user's problems to design errors: Linking evaluation to improving design practice. *Proceedings of the HCI'92 Conference on People and Computers VII*, 117-134.

Ivan E. Sutherland (1963) Sketchpad: A man-machine graphical communication system. *AFIPS Conference Proceedings 23*, 323-328.

Bruce Tognazzini (1993) Principles, techniques, and ethics of stage magic and their application to human interface design. *Human Factors in Computer Systems, INTERCHI '93 Conference Proceedings*, ACM Press, 355-362.

Joe Williams (1998) Aligning two objects with Macromedia's FreeHand. Web page. [http://www.ruku.com/fhqkalgn.html]