RICE UNIVERSITY

Computational Modeling of Icon Search

by

Michael D. Fleetwood

A THESIS SUBMITTED

IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE

Master of Arts

APPROVED, THESIS COMMITTEE:

Michael D. Byrne, Assistant

Professor Psychology

David M. Lane, Associate Professor

Psychology

James R. Pomerantz,

Professor Psychology

HOUSTON, TEXAS

DECEMBER, 2001

DECEMBER, 2001

ABSTRACT

Computational Modeling of Icon Search

by

Michael Fleetwood

As the use of graphical user interfaces expands into new areas, icons are becoming an increasingly important aspect of GUIs. Oddly, little research has been done into the costs and benefits associated with using icons. A set experiments was conducted in which various attributes of icons were examined, including simple icon borders, icon "quality" and set size (number of "distractor" icons). An eye tracking study of the task was also conducted to examine the icon search strategies of computer users. Based on the results of the studies, two models were then constructed in ACT-R/PM to carry out the same task as in the experiments. The final iteration of the models was predictive of human performance in icon search tasks. Insights into icon design and computational modeling of icon search are discussed.

Table of Contents

Beyond desktop computers, the use of icons is becoming more widespread and increasingly important. The technology once only associated with desktop computers is popping up in a variety of new locations, including mobile telephones, automobile navigation systems, kiosks, handheld computers, etc. However, the portability and size of all these new computer systems has put significant constraints on the viewable area available to the user, rendering many of the text-based interfaces commonly used on desktops ineffective. The task then for designers has been to get more information to the user in less space. Icons have been, and likely will continue to be, the preferred method in this respect.

An understanding of how users employ icons in their tasks would clearly be of great value to the community involved in designing these new interfaces. It would be of value to identify the features of icons that allow users to locate and select icons efficiently (quickly and without making mistakes). It would also be of value to have an understanding of the search process that users go through when looking for a specific icon or group of icons. The studies outlined in this paper are aimed at getting at these two issues. Initially, our studies were looking at the effect of one element of icons on icon search—simple icon borders. Over time, our studies have become increasingly concerned with the visual search process involved in locating icons on the computer screen.

A basic blueprint of the following document is as follows. Section 1 covers much of the relevant research relating to visual search and to computer icons. Sections 2 and 3 cover the first two studies that were conducted over the past year and a half. These two studies are subject-based experiments that look at how the borders of icons are utilized by computer users. In Section 4 we examine the development of a computational model of searching for icons on a computer, which was developed based on the results from the first two experiments. Section 5 describes a third experiment, which was conducted so that the results from the model and those of real subjects could be directly compared. Next, in Section 6, we will cover some the relevant research leading to an eye tracking study, which is discussed in Section 7. Finally, in Sections 8 and 9, we delve into the process of revising our ACT-R/PM models.

1. OVERVIEW OF RELEVANT RESEARCH

1.1 Visual Search Literature

It would be a difficult and very large task to cover all aspects of visual search as they have been examined in the field of psychology. People have dedicated significant portions of their lives to the subject, and the scope of this section certainly does not compare with such dedication. However, an attempt will be made to give a general review of the literature with a focus on the author's research interests, specifically those applicable to visual search in human-computer interfaces and to a general understanding of the field.

1.1.1 The Paradigm

In a standard visual search task, subjects look for a target among distractors. The total number of items in the display composes the set size. On some given percentage of trials, usually 50%, only distractors are present. When the target is present, the subject makes one response. When the target is not present, a different response is given. Two dependent measures are commonly used to examine visual search, reaction time and accuracy.

When RT is used as the dependent measure, it is generally analyzed as a function of set size. This produces two functions, one for target present and one for target absent trials. The mechanics of the search are inferred from the slopes and intercepts of these functions (See Fig. 1.1). In an efficient, or "parallel," search, reaction time should be consistent across different set sizes. This indicates that the subject is able to search for the target among all of the distractors regardless of the set size. Apparently, all items can be processed at once to a level sufficient to distinguish targets from non-targets. For example, if a subject is searching for a red object among green objects, the subject's reaction time will be roughly the same regardless of whether there are ten green objects that act as distractors or thirty green objects that act as distractors. In an inefficient search, the reaction time increases as the set size increases. This indicates that the subject must search each item or group of items serially until the target is encountered, a serial self-terminating search.

Another method, in addition to using reaction time as the dependent variable, uses accuracy as the dependent variable. In this case, the search stimulus, or the total set of objects, is presented for only a brief interval and is followed by a mask that is presumed to terminate the search. The stimulus onset asynchrony (SOA) between the initial presentation of the stimulus and the stimulus mask, is varied, and accuracy is plotted as a function of SOA. In a relatively efficient search, where all of the items in the set can be processed in a single "parallel" step, the target can be detected regardless of set size even when the SOA is relatively short. However, in an inefficient search, when each item must be uniquely examined, accuracy will be sacrificed at short SOAs that do not allow the subject ample time to examine the entire set size. Thus, the experimenter is provided with some idea as to the efficiency of the search.

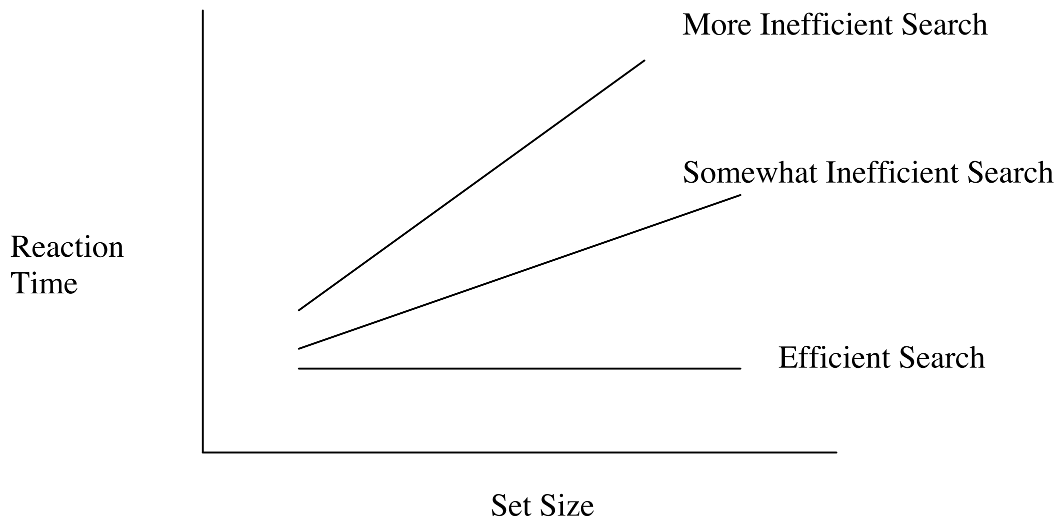

Fig. 1.1 When reaction time is used as the dependent variable and is plotted against set size, the slope of the line can be used to infer the relative efficiency of the search.

The notion of serial and parallel searches has a long history and gained strong theoretical prominence when Anne Treisman proposed her original Feature Integration Theory (FIT; Treisman and Gelade, 1980). Treisman proposed that many feature searches are parallel and that all other visual searches are serial. Feature searches are characterized as searches where targets can be distinguished from distractors on the basis of a single, basic feature, such as color, motion, or orientation (a list of basic features will be discussed later). All other searches were characterized as searches where features and targets are only distinguishable on the basis of a conjunction of several different features. For example, in a search for a red "X" among green "X"s and red "O"s the target is only distinguishable by a conjunction of color and form. Neither color nor form alone defines the target.

The original form of the FIT is no longer accepted as the basic theory of visual search. Additionally, the terms "serial" and "parallel" have gone out of favor among many psychologists as accurate terms to describe visual search, having been replaced by the terms "inefficient" and "efficient," respectively. This author, however, will continue to use the terms "parallel" and "serial" with some frequency. The terms are not accurate terms to describe the actual neural processes that underlie visual search, which is why they have gone out of favor in certain circles--where a distinction of the neural processes is being discussed. However, the terms "parallel" and "serial" seem to be good descriptive terms that convey the general notion why certain searches are efficient and others inefficient. It is this descriptive characteristic of the terms, not their accuracy, that makes them useful in relating these concepts to an audience unfamiliar with visual search, such as an audience interested in the applied aspects of visual search, as in icon search.

1.1.2 Basic Features in Visual Search

This section will describe some of the features that are considered basic features in visual search as well as survey some of the evidence that shows that these features can be preattentively searched for.

Color

A long history of basic and applied research points to color as one of the most effective ways to make a stimulus "pop-out" from its surroundings (Wolfe, 1994). This has led to a large body of research dealing with the specific properties of color as a basic feature. For instance, Nagy and Sanchez (1990) identified the smallest color difference that could support efficient visual search, the preattentive just noticeable difference (preattentive JND), and found that small color differences could support visual search.

When there is more than one distractor color, efficient search is still possible but there are constraints. A number of experiments have shown efficient search for targets of unique color among at least nine distractor colors (Smallman and Boynton, 1990). These searches with heterogeneous distractors are efficient only if the colors are widely separated in color space. When more similar colors are used, search is inefficient if the distractor colors flank the target color in color space (Wolfe, 1998). For example, a search for a white object among purple and aqua colored distractors would be relatively efficient. Whereas a search for a blue object among purple and aqua objects would be relatively inefficient because the target lies relatively close to the distractors in color space.

Orientation

Orientation is another well-accepted and well-researched basic feature in visual search. Like color, preattentive JNDs have been mapped out and examined in depth. These preattentive JNDs are larger than traditional JNDs. Subjects can discriminate between lines that differ by one or two degrees in orientation but require a difference of about 15 degrees in order to preattentively select targets based on orientation (Wolfe et al., 1999).

Curvature

Curvature is not as well accepted as color or orientation as a basic feature, but strong evidence exists that says it should be. Treisman and Gormican (1988) found that curved lines could be found in parallel among straight distractors. However, a search asymmetry exists. A search for straight lines among curved lines is much less efficient. This suggests that the presence of curvature is easier to detect than its absence. The objection to curvature as a basic feature lies in the idea that a curved line is simply a line whose orientation changes rapidly along the line. Thus, orientation is really the basic feature being distinguished.

Vernier Offset

Subjects have been found to be very good at detecting small departures from the colinearity of two line segments--a "broken" line or vernier offset. Further, subjects have been found to be able to preattentively select targets based on vernier offset (Fahle 1991). However, like curvature, preattentive detection of vernier offset may be a special case of orientation processing.

Size

If the difference in size between target and distractors is sufficient, a target of one size will be found efficiently among distractors of another size (Bilsky, Wolfe, and Friedman-Hill, 1994). However, a relatively limited amount is known about size as a basic feature. In Treisman's work on search asymmetries, she found that it was more difficult for subjects to find small targets among large distractors than it was for subjects to find large objects among small distractors (Treisman and Gormican, 1988). However, given one size of distractors, it was no easier to find bigger targets than smaller ones. Thus, the initial research done on size as a basic feature seems difficult to interpret (Wolfe, 1998).

Motion

Motion as a basic feature is relatively uncontroversial. Subjects can preattentively detect a moving stimulus among stationary distractors. Also, it is more difficult to find a stationary target among moving distractors (Dick, Ullman, and Sagi, 1987). However, because of the complexities of motion, it rapidly becomes a much more complex issue. Motion can be altered according to a number of parameters, short-range motion vs. long-range motion, direction of motion, speed of motion, etc. As a result, one cannot generalize theories across all categories of motion, but specific theories for each different parameter of motion and its relation to preattentive visual search must be examined.

Depth Cues

It is interesting to note that while preattentive processing develops only a minimal representation of an object's form, depth cues that give three-dimensionality to objects seem to be rather well represented. Enns and Rensink (1990) conducted a series of experiments that showed that both three-dimensionality based on line drawings and shading could be preattentively detected. Stereoscopic depth also seems to provide preattentive visual information. Efficient search is possible when the target lies at one depth and the distractors at another or when the target and distractors have different directions of stereoscopic tilt (Wolfe, 1998).

Shape

While motion was discussed as problematic with regard to visual search because of the number of parameters involved, form or shape is even more complex and problematic. There are a large number of experiments on objects that show preattentive visual search patterns where the features of the object are not reducible to curvature, orientation, or size (Wolfe, 1994). Although, there is little agreement as to what aspects of form characterize basic features, several candidates have been proposed.

Some of the strongest evidence suggests that line termination is a basic feature. Treisman and Gormican (1988) had subjects search for a "C" among "O"s and visa versa. Search was more efficient when "C" was the target, suggesting that the gap or line terminators were the basic feature being searched for. Of course, the opposite of line termination is closure, and some evidence exists that closure is also an important feature in preattentive search (Wolfe, 1998).

In relation to form as a basic feature of search, several other aspects of form have been examined, illusory contours, "holes", intersections, juncture, convergence, and containment. Each of these areas show some conflicting evidence as to whether they are features that can be preattentively detected. According to Wolfe (1998) the jury is more or less still out on these characteristics, and more research in the field must be conducted before one can definitively state whether they can be considered basic features.

1.1.3 Preattentive View

If all one could visually detect were preattentive basic features of objects, what would the world look like? Clearly, that will never be known for sure, but it is clear that the preattentive view of the world is not a very precise one. The preattentive perception of an object might tell if that object were “big” or “small”, not three degrees of the visual angle, that the object was “greenish”, and that the object was tilted steeply (not some wavelength and rotated 15 degrees relative to vertical). However, preattentive processing is not intended to provide an accurate picture of the world; rather, it should help divide the world into simpler components and basic objects.

Many of the concepts developed in the visual search literature are quite theoretical and can only loosely be transferred to an applied setting. After all, the goal of much of visual search research was to develop a theoretical understanding of the field and the neural process involved. However, the author’s aim is to eventually study the applied field of icon search. In order to study icon search, one needs to have an understanding of the theories behind the visual search process. Unfortunately, though, the concepts discussed in the previous section are not directly applicable to icon search for reasons that will be made explicitly clear in the following section.

1.2 Icon Search Literature

As previously mentioned, with the development of more mobile and smaller computer devices, icons are becoming an even more ubiquitous element of computer technology. The use of icons in all of these new applications has given rise to a number of interesting issues. One of the most prominent of which is determining the costs that are associated with the use of icons. These costs can be examined in terms of the time it takes a user to accomplish a task and the number of errors the user makes in the completion of the task. In this case, time and errors are only symptoms of a deeper issue—the amount and the effectiveness of the information represented by the icon. Ideally, the more information contained in the icon, the more effective the users' utilization of the icon will be. However, there are some limitations and trade-offs associated with additional detail and information contained within the icon.

In GUIs, the user often utilizes the information contained within the icon in one of two ways. In one manner, icons are simply used as a target for visual search. In this context, the icon conveys no interpretive meaning other than its association with a file or application, rather, the icon is simply a design that aids the user in icon search. Icons can also be used to convey information other than simple associative meanings. Houde and Salomon (1993) offer an example of how this can be done using simple icon borders.

The basic premise behind Houde and Salomon's use of icon borders to convey information is that real world objects come in a variety of shapes and sizes, yet icons have become standardized to be one size and often one shape on the computer screen. Small alterations to simple pieces of the icon can give the user quite a bit more information. Some examples are provided in Figure 1.2.

Figure 1.2 Examples of using icon borders to convey information.

If the file that an icon represents is a three-dimensional model, possibly created for engineering or architecture purposes, the addition of a few lines to create a three dimensional cube as a border could potentially give the user some information about the file the icon represents. Also, adding a few lines to the back of the icon border may help users differentiate from multiple-page and single-page documents. Finally, the folded corner of the icon border as a representation of documents is a current example of using icon borders to convey some simple information.

Clearly there are opportunities to make icons more informative. However, as alluded to earlier, there is evidence that more information is not always advantageous to the user. Much of this evidence centers on a trade-off between the level of complexity in the icon and the effectiveness of the icons.

1.2.1 The Complexity Trade-off

The complexity trade-off in icon design is a factor as additional detail is added to an icon. On one hand, the addition of greater detail and complexity to the icon picture should give the user more information and help him or her locate the target icon quickly and effectively (Arend, Muthig, and Wandmacher, 1987). However, the global superiority effect, taken from the visual search literature, postulates that global features of figures can be selected and responded to considerably faster than local features (Treisman and Gelade, 1980). Also, the addition of greater detail to icons, and hence, the addition of local features, adds to the number of features that the target shares with its distractors, and this will presumably slow down search. Lastly, the addition of local features adds to the general visual clutter of the display, making search for the target icon more difficult (Arend et al., 1987). Thus, adding detail to icons should help the user by increasing the information contained in the icon, but additional detail may "hurt" the user by increasing the use of local features and adding visual clutter to the display. It is critical for the effective design of icons that a greater understanding of this trade-off is achieved.

The studies described in this paper delve into the complexity trade-off and icon search by investigating the costs associated with icon borders. Certainly, many of the basic concepts and ideas from the visual search literature are applicable to icon search. For example, as features shared between the target and distractors increases the more difficult the search becomes. Also, when search can be based on a distinction between basic features of the target and distractors, search is most efficient (Treisman et al., 1980). However, icon search is a deceptively simple process. What appears to be a simple point-and-click task is actually quite complex. And due to this complexity, icon search cannot be based solely on the visual search literature. Byrne (1993) classified the factors of icon search that account for its greater complexity into three categories, general factors, graphic factors, and text factors. Each of these three classifications and the factors that make them up will be discussed to give a sense of the number of complex factors involved in icon search and why icon search demands a field of study of its own.

1.2.2 General Factors

Mixed Search

When searching for a target icon, the subject must use two search processes, which may or may not be related. They must search the graphic picture of the icon and the text label below the icon representing the file name.

Variably Mapped Task

Icon search is a variably mapped task in that the icon that is the target on one search will be a distractor on other searches, and visa versa--icons that are distractors on one search may be a target in another search.

Target Knowledge

While in visual search tasks, the subject knows exactly which object he or she is searching for, a green “X” for example. The subjects’ knowledge of the target is not always nearly so precise on every icon search. Subjects may not know the exact name of the file, or the exact picture of the icon, before they begin the search.

Multiple Matches

It is common in icon search for the target icon to share its icon picture with other icons in the visual display. Imagine searching for a document created by your favorite word processing program. It is likely that there will be a number of other documents created by this same program, with the same icon picture, in the display. In this case, the subject must use the semantic information provided by the icon label to distinguish the target icon from the other icons that share the same icon picture.

Motor task

In an applied setting, even after the subject has identified the target icon, he or she must move the cursor to the icon and click on the icon with the input device, usually a mouse. The time it takes to complete this task is affected by the size of the icon, the distance that the cursor must be moved, and the quality of the mouse.

1.2.3 Graphical Factors

Size: It has been established in the visual search literature that objects can be preattentively distinguished based on varying sizes. In icon search, generally, all of the icons on the screen are roughly the same size, so little can be accomplished in terms of selecting particular icons. However, Jacko (1999) showed that increasing the size of all of the icons decreases reaction times. Presumably, reaction times are decreased because features of the larger icons are more easily discernible. In practice, larger icons also reduce the distance that the cursor must be moved to click on the icon, further reducing reaction times. Increasing icon size is not an ideal solution to decrease reaction times, however. There is a law of diminishing returns associated with continually increasing icon size (Jacko, 1999)--i.e. icons can only be so big before their increased size is no longer helpful, but rather adds to the visual clutter of the display.

Color

Color has become a popular mode of differentiating icons among icon designers. However, caution should be exercised when using color. Despite its distinctiveness as a feature, the use of color can rapidly add visual clutter to the display.

Display Shape

This was a larger field of study at the onset of the "computer era" than it is now. Bloomfield (1970) found that subjects were quicker to locate objects when searching for them on a horizontal rectangularly shaped display than on other shapes of displays (square, vertical rectangle, etc.). Clearly, today much of this research is outdated; the shape of computer displays has become relatively standardized. However, specific cases exist when this might be a factor.

Spatial Organization

On many modern computers it is possible for the user to organize the icons in the display to his or her liking. For example, they may be presented in a grid-like or staggered organization. Interestingly enough, a number of studies in the early 1980's put forth the recommendation that icons be arranged in a circular manner because that was how users could most quickly move the cursor to the icon using a mouse (Scott, 1993). Despite the applicability of the concept, this idea obviously did not become part of the organizational scheme of modern computer displays.

A different perspective on the spatial organization of displays comes from Tullis (1983). He proposed four basic geometrical characteristics that may affect how well users are able to extract information from a display.

(1) Overall density: the number of characters displayed, expressed as a percentage of the total spaces available;

(2) Local density: the average number of characters in a five-degree visual angle around each character;

(3) Grouping: the extent to which characters on the display form well defined perceptual groups;

(4) Layout complexity: the extent to which the arrangement of items on the display follows a predictable visual scheme.

It was found that there is a high correlation between the geometrical characteristics of a display and the search time for an item on the display. Although these studies did not deal specifically with icons, it can be inferred that the visual spatial organization of the display is an important factor to be considered.

Number of Objects

One of the most studied aspects of visual search, and applicable to icon search, is that of varying the number of distractors among which the target must be selected. Varying set size, or the total number of objects in the display, including the target and distractors, is often used to describe this type of study.

The reason that set size has been studied at such an in depth level is due to the inferences that can be drawn about the efficiency of the search. If reaction times do not increase substantially as set size increases, the search can be termed as a relatively efficient one. Conversely, if reaction times substantially increase as set size increases, the search can be termed a relatively inefficient one.

Form

One of the principal dimensions that icons vary on is that of their shape or form. As in the visual search literature, form is a very complex factor and contains a number of sub-dimensions, such as the level of detail in the icon and the meaningfulness of the form.

1.2.4 Text Factors:

Icon labels

It has been previously noted that icon search relies on both a graphical search of the icon picture and a text search of the icon label before the target icon can be identified. One of the interesting aspects of icon labels is that they often are not made up of complete words. They may be abbreviated words, parts of words, strings of letters only meaningful to their creator, or even numbers or a combination of the above.

Sorting

An additional source of complexity that must be dealt with before icon search is fully understood is the use of sorting features by users. On many computers, icons may be arranged or sorted alphabetically according to their label, by the date that they were last modified, by the type of application that was used to create the document, or even by the type of document the icon represents.

2. EXPERIMENT 1

Each of the factors discussed in the previous section of this paper is worthy of future research and individual examination. A good starting point for studying icon search would ideally examine icon form, or shape, because, as previously mentioned, it is the primary characteristic by which computer users differentiate icons. It would also be beneficial to the field of icon search to begin to examine the apparent dilemma in the complexity trade-off.

The aspect of icons chosen by the author to be initially examined meets these two criteria. Icon borders are currently a key element of the form of icons and worthy of further examination on that end. Additionally, the borders of the icon picture provide an example of the complexity trade-off. It may be that icon borders, as they are currently used, do little to aid the user and only add visual clutter to the display. Lastly, it would be valuable to gain some understanding of the relative efficiency or inefficiency of icon search in general.

The following experiment was designed to test these hypotheses: (1) that icon search is a relatively inefficient process and (2) that icon borders only add to the visual clutter of the display and hence increase reaction times for target identification.

Fig. 2.1 Some examples of icons with different borders. Note that these icons are identical to some of the icons used in the experiment. (The file names are selected randomly from a list of words.)

2.1 Method

2.1.1 Design

The design of the experiment was intentionally kept relatively simple, although a certain level of complexity was necessary to examine the process. Three independent variables were manipulated, all of which were within-subjects factors. The first of these factors, set size, had four levels, 6, 12, 18, or 24 icons. A second within-subjects factor, target type, had three levels. The target icon to be searched for could be presented without a border (no-border condition), with a circle as a border (circle), or with a box as a border (square). Refer to Figure 1 for examples of each border type. The final within-subjects factor was termed distractor type and was varied at two levels. In the matched condition, the distractors among which the user searched for the target, had borders matching that of the target—i.e. if the target icon had a square border, then all of the distractors would also have square borders. In the mixed condition, the borders of the distractors were varied—i.e. the user searched for the target among icons without borders and with circles and squares for borders. In this condition, each of three border types was randomly assigned to the distractors in the display. Each block in the experiment thus consisted of 24 trials. Each independent variable was examined at each level of the other independent variables (4 x 3 x 2 = 24). The order of presentation was randomized.

The dependent variable being measured was the response time of the users—specifically, the time from when they clicked on the "Ready" button to indicate that they were finished examining the target icon to when they clicked on the target icon among the set of distractor icons.

One potential independent variable that was held constant in this experiment was the number of icons matching the target in the search display. On each trial one-third of the icons in the search display had the same pictorial icon and matching border. Thus, ultimately the user was forced to differentiate among the icons by the file name.

2.1.2 Procedures

Users were initially given some instructions as to how to perform the task, then were given one block of practice trials to develop some familiarity with the task and with the mouse used to point and click on the target icon.

Each trial had two stages. On the initial screen of the first stage, users were presented with a target icon and a corresponding file name. After 1500 milliseconds, a button labeled "Ready" appeared in the lower right corner of the screen. Users could move the mouse and click on the button whenever they felt they had sufficiently examined the icon and were ready to move on to the next stage of the trial.

Immediately after clicking on the Ready button, the users were presented with a screen that contained a number of icons (6, 12, 18, or 24), one of which was the target icon. The user’s task was to identify the target icon and click on it as quickly as possible. Clicking on an icon brought them to the first stage of the succeeding trial. Response time was measured from the time they clicked on the ready button to the time they clicked on an icon in the distractor set.

The location of the target icon was randomly selected for each trial. Also randomly selected were the file names for the icons. The distractor file names and the target file names were randomly selected without replacement from a list of 750 names until the list was exhausted. At which time, the list was recycled.

Each user completed four blocks of trials in addition to the practice block for a total of 120 trials.

2.1.3 Users

The users in the experiment were 25 undergraduate students at Rice University who were participating in order to meet a requirement for a psychology course. Although some variation with regard to computer experience was expected, users in this population are generally familiar with computer use.

2.1.4 Apparatus / Materials

The experiment was conducted on Apple Macintosh iMac personal computers. The icons used in the experiment were standard Macintosh sized icons (32 pixels by 32 pixels). They were subjectively designed as arbitrary shapes by the experimenters. An effort was made to design the icons so that they did not represent any specific well-known shape or object. Additionally, they were created entirely in grayscale—none of the icons contained any color other than white, black, and shades of gray.

2.2 Results

When the user did not correctly identify the target icon, the trial was considered an error and removed. Outliers were also removed when the response time was more than three standard deviations from the 5% trimmed mean of the user for the corresponding set size. In total, less than 5% of the trials were removed due to errors and outliers. For statistical tests, where response times had been removed as errors or outliers, they were replaced with the user's grand mean.

Figure 2.2 provides a graphical summary of the data. Examination of the data provided evidence for some interesting patterns, described subsequently.

One interesting pattern in the data is a reliable effect of set size, $F(3, 72) = 117.89$, $p < 0.001$. It took users an average of 1422 ms to locate and click on the target icon in the smallest distractor set size. For each additional six icons added to the distractor set, the users took approximately one-half second longer to locate the target icon. This consistent increase in response time across set sizes is confirmed by a reliable linear effect of set size, $F(1, 24) = 168.55$, $p < 0.001$.

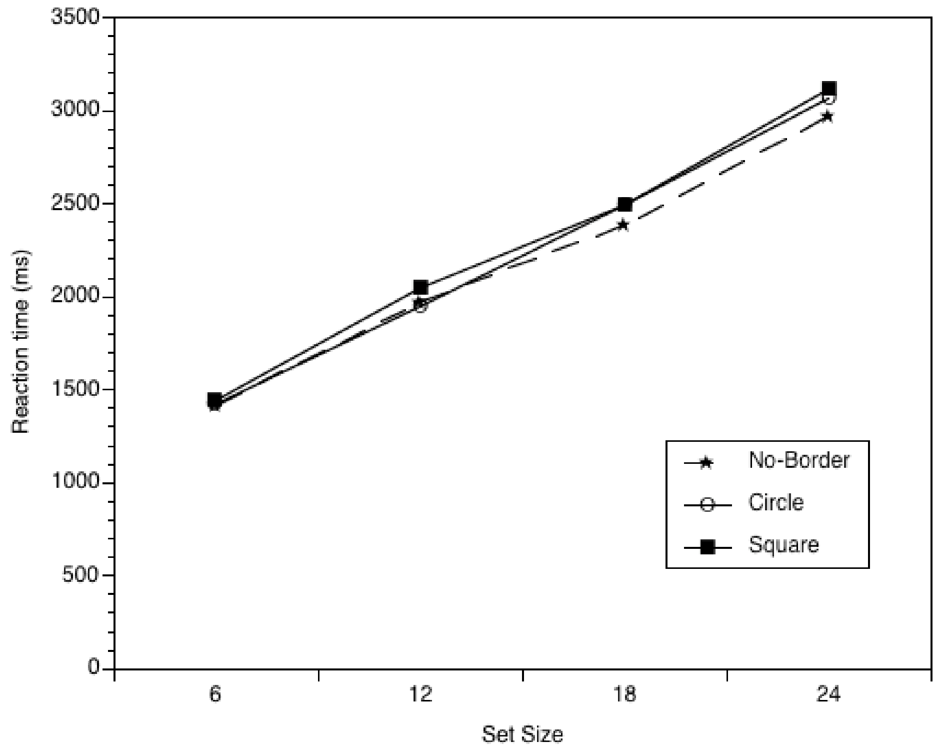

Figure 2.2 Plot of mean reaction times by set size and border type.

Another interesting result observed in the data is the lack of any reliable effect, $F(2, 48) = 1.66$, $p = 0.20$, or interaction involving border-type. Hence, no evidence is provided to support the hypothesis that the type of border affects the icon search.

Whether the distractor borders matched the target border or were a mixed set of the three different border types did not significantly affect the users' performance on the task. No evidence was found for an effect of distractor-type, $F(1,24) = 1.48$, $p = 0.24$, nor for any interactions involving distractor-type.

2.3 Discussion of Experiment One

The results of Experiment 1 provide some insights into icon search. First, the cost in time of a search for a target icon is a linear function of the number of icons in the display. Also, the lack of any evidence for simple icon borders affecting icon search response times is an interesting one. This result is contrary to one of the original hypotheses of the experiment, that icon borders, where they only add arbitrary detail to an icon, will only slow users by adding addition clutter and shared features between target and distractors to the display. Additionally, it suggests that simple icon borders have the potential to break the complexity trade-off—i.e. a means of getting information to some users through increased detail at virtually no cost to those users who do not take advantage of it. However, before drawing definitive conclusions, there should be some additional examination of the process. Specifically, there should be an examination of whether such a result can be found across other types of icons. This was the motivation for a second icon search study.

3. EXPERIMENT 2

The experiment was designed to investigate the effect of simple icon borders on a broader range of icons (relative to Experiment1). The range of icons examined varied according to their relative quality—defined here as their distinctiveness from other icons.

3.1 Method

3.1.1 Design

The design of the experiment was very similar to that of the first experiment. The basic paradigm was not changed, but one independent variable, termed icon quality, was added, and the independent variable of distractor-type (mixed or matched conditions) was removed. The other independent variables, set size and target type, were not changed.

The within-subjects factor that was added to the design, icon quality, had three levels. Icons were designed that varied in their level of distinctiveness. On one end of the spectrum were icons of “good” quality. These icons were designed to be easily distinguishable from other icons based on the basic visual (“pop-out”) features of color and shape (specifically curvature). Icons in the good quality set were one of six colors (red, blue, green, yellow, brown, or black) and one of two shapes (circle or triangle). Examples are shown in Figure 3.1. On the other end of the quality spectrum were icons that were not easily distinguishable (referred to as “poor” quality icons). They were designed to be distinguishable in a set of two icons, but quite indistinguishable in a large distractor set. These poor quality icons were all of the same basic shape and did not include color (other than white, black and shades of gray). The “fair” quality icons were designed to be representative of the area in between these two ends of the spectrum. They were generally of a distinct shape, although more complex than the simple circles and triangles in the good quality icons, and none of them contained any color outside of the spectrum of gray scale colors.

The within-subjects factor of distractor type was removed from the analysis. The first experiment found no evidence of an effect or interaction involving distractor type so this factor was not considered further.

Each block in the experiment thus consisted of 36 trials. Each independent variable was examined at each level of the other independent variables (4 x 3 x 3 = 36).

As in the first experiment, the dependent variable being measured was the response time of the users—specifically, the time from when they clicked on a button to indicate that they were finished examining the target icon to when they clicked on the target icon among the set of distractor icons.

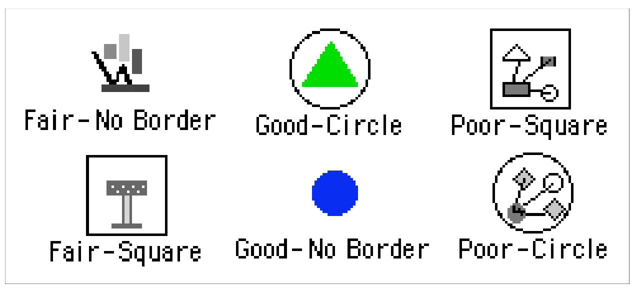

Figure 3.1 Examples of icons of good, fair, and poor quality used in the experiment. The good quality icons were each a single solid color, whereas the fair and poor quality icons were drawn in grayscale.

3.1.2 Procedures

The task is nearly identical to that performed in the first experiment. The only changes being the addition of the icon quality factor, the removal of the distractor type variable, and thus an increase in the number of trials in each of the five blocks. Each user completed four blocks of trials in addition to the practice block for a total of 180 trials.

The paradigm of the presentation of the target icon and the search among distractor icons remained exactly the same, however.

3.1.3 Users

The users in the experiment were 20 undergraduate students at Rice University who were participating in order to meet a requirement for a psychology course. Although some variation with regard to computer experience was expected, users in this population are generally familiar with computer use.

3.1.4 Apparatus

The experiment was conducted on Apple Macintosh iMac personal computers.

3.2 Results

User errors and outliers were removed from the data according to the same procedure as Experiment 1. In total, less than 5% of the trials were replaced due to errors and outliers. For statistical tests, where response times had been removed as errors or outliers, they were replaced with the user's grand mean.

A summary of the data, collapsed across the icon quality variable, provides results quite similar to those in first experiment. (Refer to Figure 3.2.) Response times range from approximately 1500 ms for the smallest set size of six icons, and increase by 400 to 500 milliseconds for each additional six icons added to the distractor set, up to about 3000 ms for a set size of twenty-four icons. As in the first experiment, there is a significant effect of set size ,$F(3,57) = 210.45$, $p < 0.001$. Also, there is a reliable linear effect of set size, $F(1,19) = 426.36$, $p < 0.001$. Additionally, as in the first experiment, there is not a main effect of border type, $F(2,38) = 0.46$, $p = 0.63$.

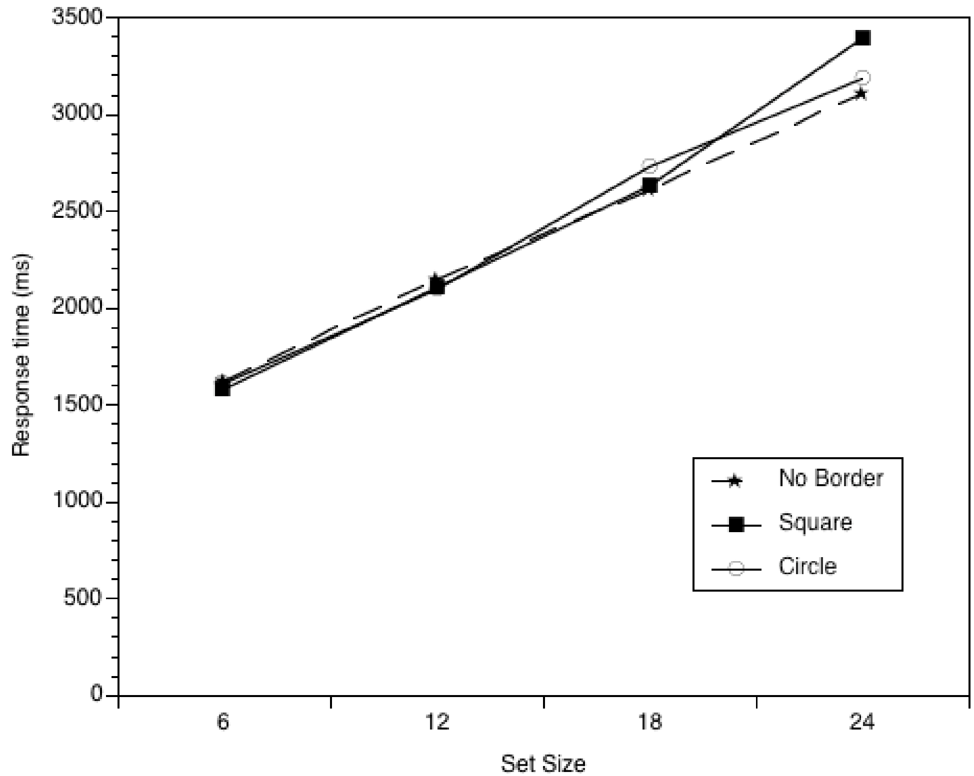

Figure 3.2 Mean reaction times by set size and border type.

In Figure 3.3, mean response times are presented as a function of set size and icon quality. Here, it is evident that as icon quality decreases (good to fair to poor), response times increase. This is confirmed by a significant main effect of quality, $F(2,38) = 52.14$, $p < 0.001$. Also, not only are the three qualities significantly different, but the slopes of the lines appear to be different, as confirmed by a reliable quality by set size linear interaction, $F(6,14) = 5.20$, $p < 0.01$.

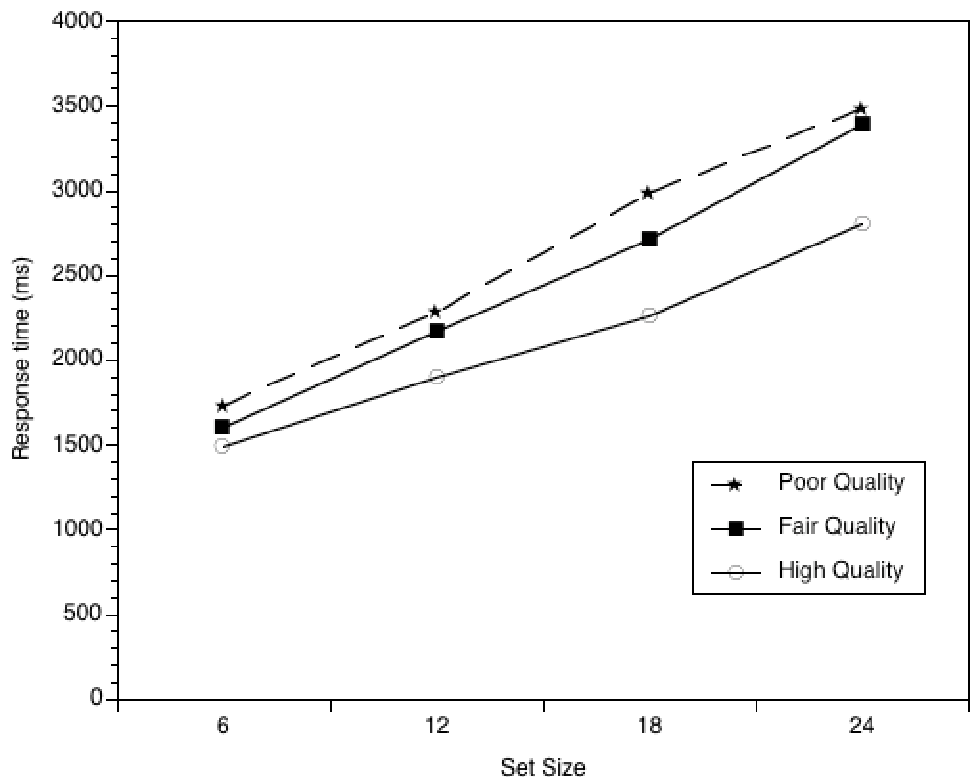

Figure 3.3 Mean response times by set size and icon quality, illustrating a main effect of icon quality.

In Figure 3.4, mean response times are presented across icon quality and target border type. This chart once again displays the main effect of icon quality—as quality decreases, response times increase. Here again, there is no effect of border type, and no interactions involving border type. Also, the difference in response times across the different icon qualities is relatively consistent, leading to a reliable linear effect of icon quality, $F(1,19) = 103.06$, $p < 0.001$. One key aspect to note from this chart is the relatively large effect of icon quality. Any potential effects of border type would be far outweighed by the main effect of quality.

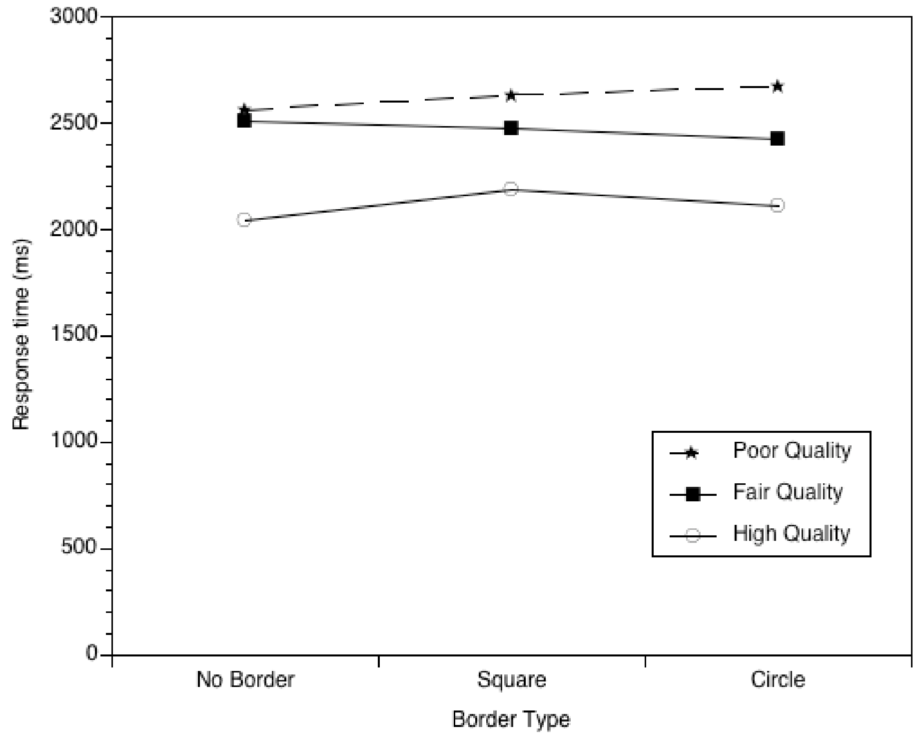

Figure 3.4 Response times by icon quality and border type.

This lack of an effect involving target border is very clear at lower set sizes. However, at the largest set size of 24 icons, the data become quite a bit muddier, particularly for poor quality icons. Here again it is important to note that even at these large set sizes, where target border type may begin to have an effect on response times, this effect would be far outweighed by the relatively larger effects of set size and icon quality.

3.3 Discussion of Experiment Two

Experiment 2 confirmed that users' response times to selecting target icons is unaffected by simple icon borders. While the data do not definitively rule out a small effect of borders at large set sizes, the potential effect was far outweighed in magnitude by the effects of set size and icon quality. These results confirm the findings of Experiment 1, but further generalize them across a broader range of icons varying in quality.

Additionally, the two most prominent effects produced in Experiment 2 are those of set size and icon quality. As in Experiment 1, Experiment 2 produced a very consistent effect of set size. It took users approximately 1500 milliseconds to locate and click on the target icon in a set of six icons. For each additional six icons that were added to the distractor set, response times increased by approximately 500 milliseconds. The mean response times involved are likely specific to the icons that were used. However, the linear relationship between set size and response time is an important one to note. Second, an effect of icon quality was also produced, indicating that the quality of the icons has a significant effect on user response time.

3.4 General Discussion

The conclusion that simple icon borders do not affect icon search suggests that methods of adding information to icons through borders , such as those suggested by Houde and Salomon (1993), are quite promising. The reason being that the cost of presenting this information to the user may be trivial. For the purposes of search, users are able to ignore borders, and thus, this set of experiments indicates that the presence of borders will not adversely “harm” users in terms of response time.

These conclusions may not apply to any and all icon borders. Certainly, as borders become more extravagant and move beyond the very simple shapes used in these experiments, they may become more visually demanding of the users' time and interest [4]. However, the conclusions drawn here should be leveraged to suggest that the judicious use of icon borders is unlikely to cost more in time than what could potentially be saved if the information is effectively used. In comparison to the effects of set size and icon quality on search time, any effect of icon borders is likely to be trivial. Thus, if an icon designer can use icon borders to transmit some information, it is likely worth the effort to try it, because the effect of the additional detail on search time, if any, will be minute.

The two previously discussed experiments together have given us a fairly good concept of how simple icon borders affect icon search and allowed us to even begin to develop a proposed theory for how simple borders can be used in an applied setting. We would like, however, to be able to generalize even further from our results, and fortunately, the data from these two experiments provide us with a platform for doing so. Specifically, we would like to have a thorough understanding of how users employ icons. Ideally, if we can understand users' search strategies where icons are used, then we will be able to make predictions about the factors that allow for the efficient use of icons. One method of examining the icon search process in greater depth is to develop a computational model of the task.

4. COMPUTATIONAL MODELING OF EXPERIMENT 2

The system that was used to model the experiments was ACT-R/PM (for ACT-R Perceptual Motor). The ACT-R architecture has been used to successfully model a variety of behavioral phenomena and has proven particularly successful at modeling tasks with a demanding cognitive component (Anderson and Lebiere, 1998). A thorough description and discussion of the ACT-R framework is given in Anderson and Lebiere's *The Atomic Components of Thought* (1998). In ACT-R/PM, the original system has been combined with modules for perceptual and motor actions (vision, audition, motor, and speech; see Chap 6 of *The Atomic Components of Thought*, Byrne and Anderson, 1998) for a discussion of the functioning of the different modules.). Because icon search is relatively light on the cognitive demands of the user, it is certainly a task that must be modeled under an architecture that accounts for the perceptual and motor components inherent in the task—i.e. searching a set of distractors for the target icon and pointing and clicking with a mouse. Also, by adding perceptual and motor modeling capabilities to an already well-developed cognitive system, tasks such as icon search can be modeled that incorporate the properties of three things: the cognitive, perceptual, and motor capabilities of the user, the task the user is engaged in, and the artifact the user is employing in order to do the task (Byrne, 2001). Each of these three components, embodied cognition, the task, and the artifact, must be considered to model a task such as icon search with some accuracy.

Such a system as ACT-R/PM provides three distinct advantages over other systems that allow experiments such as the one this paper focuses on to be modeled.

1 Parallelism. People can move both the mouse and their eyes at the same time. The distinct modules in ACT-R/PM can accurately model such parallel activities.

2 Display Interaction. ACT-R/PM can "see" the computer display and dynamically react to what is presented on the screen.

3 Accurate timing. Each of the subcomponents in ACT-R/PM reflects the most current knowledge of human performance regarding the timing information pertaining to the perceptual and motor capabilities of the system. One of the implicit goals of modeling an experiment based on empirical data, such as the icon search experiment discussed here, in ACT-R/PM is to examine the accuracy of the timing parameters.

The logic and the processes behind each of these distinct advantages will be discussed in some additional detail. The concept of "parallelism" will be discussed in relation to the system organization of ACT-R/PM. The concept of "display interaction" is related to the vision module in ACT-R/PM, and because the task we are modeling is primarily a visual one, this module will be covered in greater detail. The value of accurate timing for events in ACT-R/PM will be discussed as well, particularly in the context of the timing of specific parameters that relate to the experiment we are modeling.

System Organization

ACT-R/PM is organized as depicted in Figure 4.1 In ACT-R/PM, there are four perceptual-motor modules which communicate with central cognition, which is realized as a production system (in this case ACT-R). Central cognition is more or less serial (spreading activation processes work in parallel) and each module is itself more or less serial, but the various components all run in parallel with one another. Thus, the production system could be retrieving something from long-term declarative memory while the Vision Module is shifting attention in the visual array and the Motor Module is preparing to press a key.

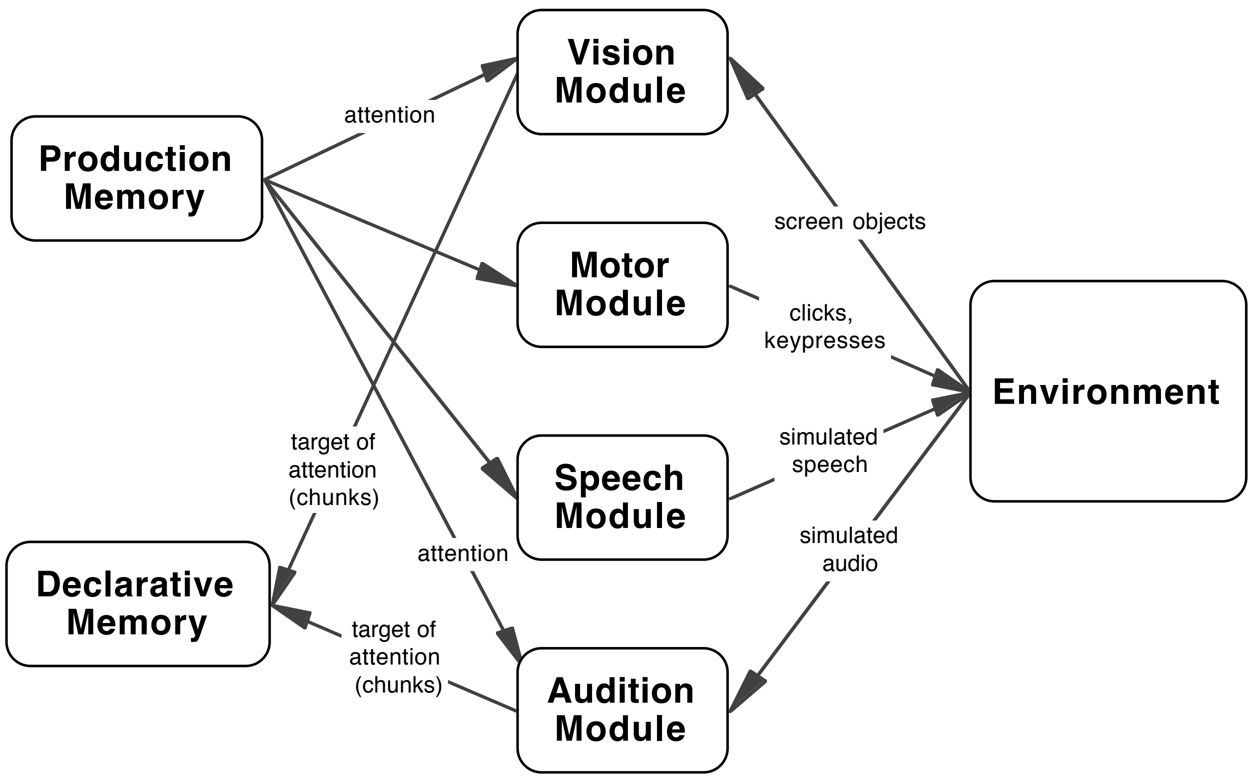

Figure 4.1 ACT-R/PM system diagram. From Byrne (in press).

<u>The Vision Module</u>

Given the visual nature of the interface we are working with in this instance, the Vision Module is of key importance in modeling the task. As one might expect, the Vision Module is used to determine what ACT-R/PM “sees.” Each object on the display will be represented by one or more features in the Vision Module. The Vision Module creates chunks from these features which provide declarative memory representations of the visual scene, which can then be matched by productions.

For the Vision Module to create a chunk representing an object, visual attention must first be directed to the location of that object. In order to do that, the Vision Module uses a MOVE-ATTENTION operator. However, in order to shift attention to a location, ACT-R/PM also must have a representation for visual locations. The state of the visual array is reflected in "virtual chunks" which can be matched by the IF side of production rules. When ACT-R/PM is "looking" for an object, in the case of these experiments, an icon, on the display, one or more locations may be matched by this process, depending on the number of objects on the display and the specificity of the chunk match.

The Vision Module represents basic information about the visual features in the display such as their location, their color, their size, etc. A production may specify values (or ranges of values for continuous properties such as location or size) that are acceptable for a match. When a match is found and visual attention is shifted to the object, a chunk is added to ACT-R's declarative memory, and this chunk represents the knowledge that there is something at a particular location. Attention shifts happen asynchronously with respect to the production system and have a duration that defaults to 135 ms (but can be set as a parameter). If attention shifts to a location, and there is an object at that location, then a chunk will be created which represents that object (a visual-object chunk), and that chunk is considered the focus of visual attention.

If there is more than one object at a location when an attention shift completes, then the Vision Module will encode—create chunks representing—all the objects at that location. However, only one of those objects can be the focus of visual attention, so one object must be selected. The selection is based on the properties of the feature that was originally used to generate the location chunk. Thus, if the request to shift attention is based on a location originally specified to have a red object, then if there is a red object at the location when a shift completes, it will tend to be preferred as the focus of attention.

Accurate Timing

As mentioned, great care has been taken in the development of ACT-R/PM to accurately model the real time of events. In this respect, the motor module is quite complex. Each of the actions in the motor module is carried out in three phases—preparation, initiation, and execution. In executing a movement, it must first be prepared by the motor module, which takes a minimum of 50ms and ranges upward depending on the movement. Once the movement has been prepared, the amount of time that a movement takes to execute depends on the type and possibly the distance the movement will traverse. Simple movements have a minimum execution time (also 50 ms, called the "burst time") and more complex movements (such as pointing with the mouse) have a longer execution time based on Fitts' Law.

Other than the motor movements of mouse movements and clicks, there are two specific times that should be noted for discussion of the modeling of the previously described experiment. One of these times has already been mentioned, the time it takes to shift attention, which has been estimated at 135 ms. Also of key importance is the time it takes to for one production to be executed, which is estimated to be 50 ms. The relevance of these two times will become apparent as we move into the discussion of our model.

A cognitive model of icon search, as it is presented here, will clearly benefit from the capabilities of ACT-R/PM. It is an activity requiring little explicit cognitive strain, yet demands quite a lot of the perceptual and motor capabilities of the subject. A cognitive modeling system that can meet such perceptual and motor demands is necessary for icon search to be accurately modeled. ACT-R/PM meets such demands.

4.1 The Models

We chose to develop two slightly different models of the icon search task, each representing a slightly different search strategy. This was done for two reasons. First, as previously mentioned, icon search is a deceptively simple task. Because our goal in modeling the task is to learn something about the strategies that users employ in the task, we chose to begin our modeling study broadly, with two different models, and hone in on the correct strategy as we learned more about the process. Additionally, there is some degree of complexity added to the interpretation of a model as the model itself becomes more complex. For this reason, two relatively simple models were developed initially, and complexity would be added later where it was seen as necessary. Both of the two models developed will be considered here in some detail.

The "Double-Shift" Model

Both of the models discussed here follow the same basic control structure. Where they differ is in the specific strategy that they use to search for and identify the target icon. The double-shift model is so named because it requires two shifts of attention to examine each icon. First, the model directs its attention to an icon that has at least one feature in common with the target icon (a red circle for example). Then the model shifts attention to the filename directly below the target icon. This filename is compared with the target filename which has been "remembered." If the filenames match, then attention is shifted back to the target icon so that it can be clicked on.

It is worth noting what is meant by an "element" or "feature" of the icon that the models note in order to guide later search. The elements do not directly correspond to low-level features processed by the human visual perception system, where something such as "gray rectangle" would presumably be represented by a number of separate lower-level features. Rather, icons are represented on a coarser level by a set of "features" designed simply to provide the models with some aspect of the icon that can be used to distinguish a specific icon in later search.

The "Text-Look" Model

The text-look model is so named because attention is focused directly on the filename below the icon, and the actual icon is never actually attended. As in the double-shift model, an icon that matches the text is located, but rather than shifting visual attention to the icon, it is shifted directly to the filename below the icon. This process is meant to simulate the process of preattentive search. It is assumed in this model that under conditions where the target icon shares few features with the distractor icons—i.e. they are somewhat unique—then users do not need to examine the icon in detail, rather, they just look directly at the filename. The model does shift attention to the icon eventually, in order to move the mouse and click on it, but this attention shift only occurs after the target filename (and thus the target icon) has been identified.

The two models make different predictions about the reaction times of users. Clearly, the text-look model should be faster at locating the target icon because it requires one less shift of visual attention for every distractor icon that is examined. This advantage in speed in the text-look model is accounted for the in the double-shift model by preparing the mouse movement to click on the target icon—i.e. the system is given some knowledge about the direction and distance the "hand" will be moved and this corresponds to a decrease in the time it takes to make the movement. Such an implementation in the model is cognitively plausible because real subjects certainly have prior knowledge about the mouse movement and, in many instances, probably even begin to make the movement during the search process.

4.2 The Task

It is worth examining in detail how the icon search task was modeled in ACT-R/PM. The basic task, in simple terms, requires the subject to look at an icon and a file name, remember it and its corresponding file name, click on the "Ready" button, then find the icon on the succeeding screen and move the mouse to the target icon so that it can be clicked on. The next section of this paper provides a brief description of the relevant productions in the double-shift model that implement this sequence of events in ACT-R/PM. The descriptions of productions described below are only those productions that are unique to the double-shift model, and thus the descriptions begin with the productions that fire after the "ready" button has been clicked and the model is faced with the distractor screen. (To see the complete set of the productions and their descriptions, refer to Appendix 1 at the back of this paper.)

FOCUS-ICON-1
Now in distractor screen, move attention to a specific icon that has a feature matching that of the target icon stored in the goal, and also prepare the hand to move the mouse in the towards the center of the distractor set. (This production will only be called once.)

IF the goal is icon-search and the goal-state is “distractor1”
And there is a visual-location that has not been attended and whose feature and color matches that stored in the goal.
And the vision module is free
And the motor module is free
THEN move visual attention to the location.
And prepare the motor movement of the right hand
And change the goal-state to “down”.

LOOK-TEXT
Find the filename below the attended icon and attend to it.

IF the goal is icon-search and the goal-state is “down”
And there is a visual-object at a specific screen position
And there is a visual location of the kind text, which has not been attended, and which is nearest to the visual-object
and the vision module is free
THEN move visual-attention to the visual location
And change the goal-state to “name”.

From here, if the text matches that of the target filename, then attention is shifted back to the icon above the text, and it is clicked on to end the trial. If the text does not match the target filename, then the following production is called (which is quite similar to Focus-Icon I, only that the hand movements are not prepared again).

FOCUS-ICON-2
Now in distractor screen, move attention to a specific icon that has a feature matching that of the target icon stored in the goal. (This production is just like Focus-icon-1 except the hand movement is not prepared.)
IF the goal is icon-search and the goal-state is “distractor1”
And there is a visual-location that has not been attended and whose feature and color matches that stored in the goal.
And the vision module is free
THEN move visual attention to the location.
And change the goal-state to “down”.

The productions for the text-look model are very similar. In fact, they are identical up to the Focus-Icon production and from the Examine-Text-N production on. However, the Focus-Icon (1 and 2) and the Look-Text productions have been replaced with the following two productions.

CHECK-ICON
Locates an icon on the screen with a feature that matches the feature of the target icon previously stored in the goal, but does not move visual attention to that icon.

IF the goal is icon-search and the goal-state is “distractor”
And there is a visual-location that has not been attended and whose feature and color matches that stored in the goal.
And it is visual-location is not the location of the previously attended icon (stored in the goal)
And the vision module is free
THEN move visual attention to the location.
And store the location of this icon in the goal.
And change the goal-state to “distractor2”.

TEXT-FIRST
Moves visual attention to the filename corresponding to the icon located in Check-Icon.

IF the goal is icon-search and the goal-state is “distractor2”
And there is a visual location of the kind text, which has not been attended, and which is nearest to the location stored in the goal
and the vision module is free
THEN move visual-attention to the visual location
And change the goal-state to “name”.

One thing worth explicitly noting about the way the model accomplishes the task is the strategy that the model uses to find the target icon in the set of distractors. In the FOCUS-ICON and the CHECK-ICON productions, the left-hand side of the production matches to an icon that has the value and feature stored in the goal and corresponding to the target icon. The production conducts this search in a random fashion. The production could just as easily search in a left-to-right, right-to-left, top-to-bottom, or bottom-to-top manner. However, the manner that search is conducted is somewhat arbitrary in this context, and because the location of the target icon is random, which search strategy is used will have little effect on the mean reaction times. (Clearly, there is an opportunity for improving the model here, and this will be discussed later in the paper.)

It is also worth explicitly noting one prediction made by the double-shift model—the time that it predicts it will take users to look at an additional icon. After the model “looks” at an icon its filename and determines that it is not the target icon, it will search for another icon that matches the features of the target icon it has stored. This corresponds to running the FOCUS-ICON production again and will take 50 ms—the estimated real time to run one production. Next, the model must shift visual attention to the icon that it has found with the correct features, and as noted before, this shift of visual attention will take 135 ms. Then, the model will match and run the LOOK-TEXT production to shift attention to the filename. Again, this will take 50 ms to run the production and 135 ms to shift visual attention. From here, a production will examine if the filename matches that of the target icon (taking another 50 ms), and then the model will begin the set of productions aimed at clicking on the icon, otherwise, it will repeat the process of examining another icon. Because there is no feature overlap in the “good” quality icons, and each set size adds two icons that match the target to the display (meaning on average one of them will be visited), the model clearly predicts a 420 ms slope for the RT by set size function for “good” icons. For other icon qualities, the slope depends on the degree of feature overlap between the target and the distractors. That is, if the target icon contained a gray square and the model selected that feature to guide search, then all icons on the display containing gray squares are candidates, referred to as "target-matching" icons. The number of such icons will vary from trial to trial depending on the features in the target icon and the composition of the distractors, so Monte Carlo simulations are required to produce RT predictions. We should be able to compare this model-predicted time with the average time to examine an additional icon observed in the data collected from subjects in Experiment 2. Indeed, this has been done and will be discussed in detail as we discuss the results of our model.

Quantitative predictions from the TL model are not so easy to compute because this model may re-examine icons. This is because the icons themselves are never actually attended, and ACT-R/PM only "remembers" locations to which it has shifted attention. Because this revisitation is probabilistic, analytic predictions are difficult to derive and again Monte Carlo simulations are required.

4.2.1 Comparison of the Models

The two models represent slightly different strategies for the visual search. The DS model makes two shifts of attention per each additional icon examined, while the TL model makes only one, suggesting the TL model may be more efficient. However, the DS model does not revisit previously-seen items, which could make the DS model more efficient. We had no *a priori* predictions about which model would actually be faster.

The models have some key similarities as well. The production, which selects the next icon to be examined, selects randomly from all the candidates that match the remembered feature (e.g. "gray circle"). That is, there is no right-to-left or top-to-bottom pattern employed by the models. Because the location of the target was random, incorporating such a strategy would have made little difference in the ultimate predictions of the model in this experiment. Furthermore, the models employ the same strategy for all set sizes and icon qualities. These properties will be discussed further later in the paper.

Finally, both models depend on the representation of the icons themselves. Each icon is “seen” by ACT-R/PM’s Vision Module as a list of features. For the “good” quality icons, this list is a singleton; that is, each icon is represented by a single feature (e.g. “red circle”). In contrast, more complex icons will have a number of features and colors associated with them, gray triangles, white circles, etc. What makes these more complex icons “poor” icons in the experiment is not the number of features the icon has per se, but rather the number of features the icon shares with other icons in the distractor set. For example, many of the icons in the poor quality set have gray triangles and white circles. As a result, the model will often examine icons that do not match the target icon exactly, but rather only share one particular feature with the target icon. It is this overlap of features, or “similarity,” that makes such icons poor icons in context. In contrast, the good quality icons have no feature overlap with other good quality icons, and thus, only icons exactly matching the target icon are examined by the model. The exact nature and number of the features used to represent each icon in the “fair” and “poor” conditions are free parameters in the models; however, we used the same feature sets for both the DS and TL models.

4.3 Model Results

The following figures provide a graphical comparison of the model data with the "real" data. Note that the model data was gathered by running the model for 80 blocks of trials, which approximates the number of real blocks of subject data involved (80-84).

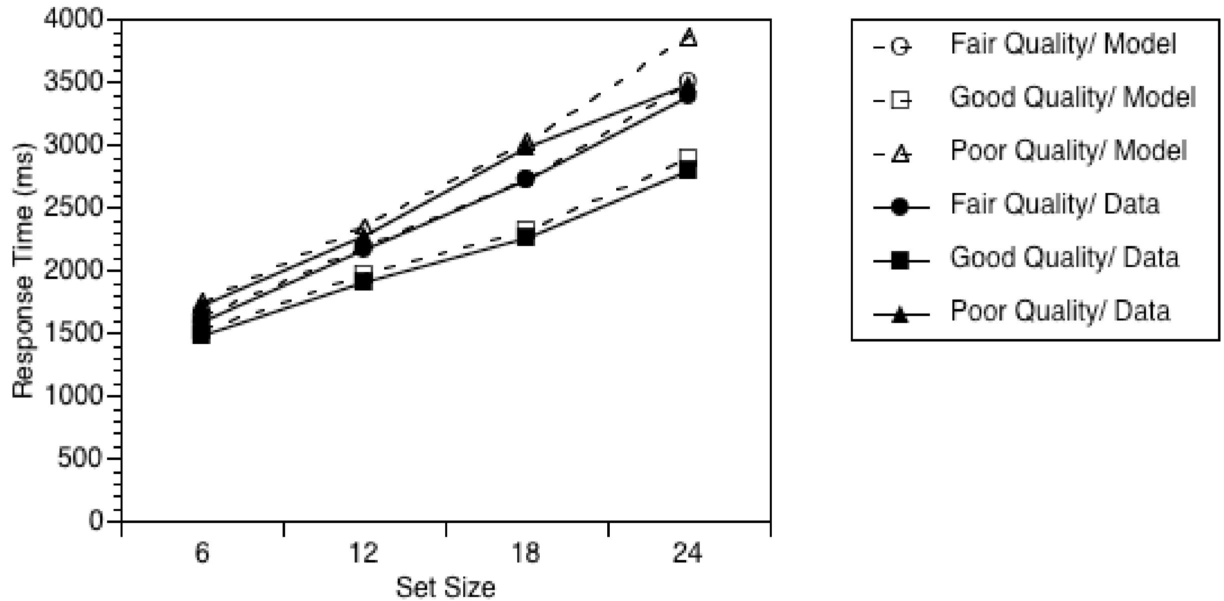

Figure 4.1 Double-Shift Model. Reaction Time by Set-Size and Icon Quality.

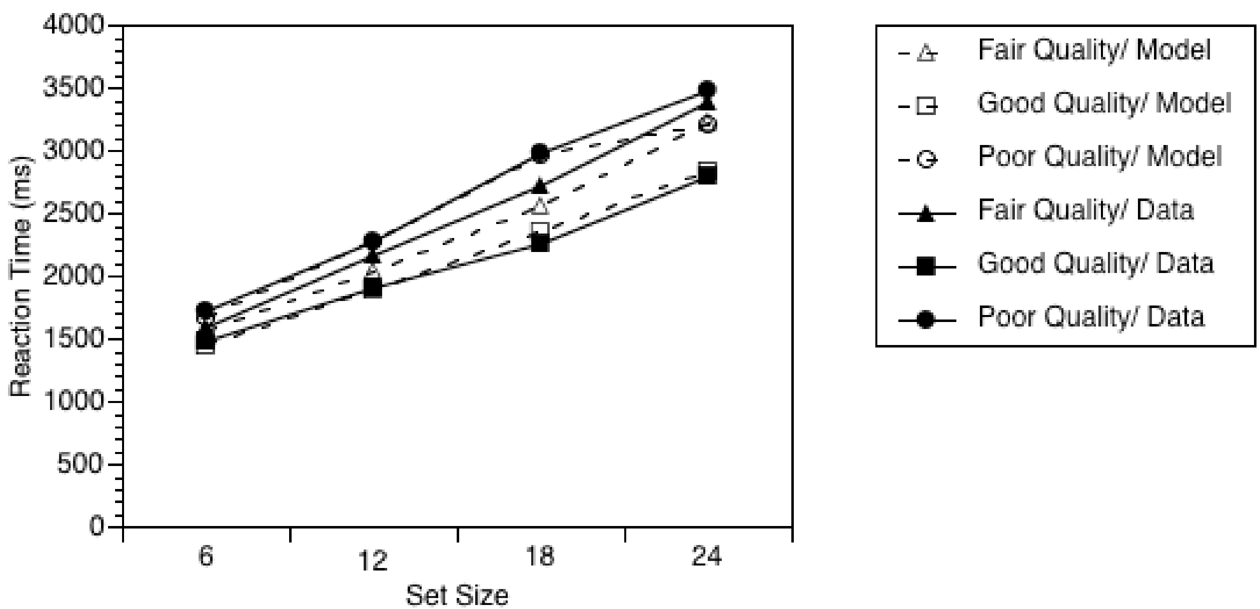

Figure 4.2 Text-Look Model. Reaction Time by Set Size and Icon Quality.

It is clear from the previous charts that the performance of the model is quite similar to that of real subjects. Where the model and the data show some divergence, at the largest set size, this is somewhat to be expected due to the greater amount of variability in reaction times at large set sizes. Most importantly, the model has retained each of the pronounced effects that were seen in the data—those of set size and icon quality. A final quantitative comparison of the two sets of data was obtained by examining the correlation and the percent mean deviation (root-mean-square) between the two sets of mean reaction times.

The R^2 between the data matrices for both the text-look and double-shift models is 0.98. The high correlation between the models and the data suggest that the models both do an excellent job of accounting for the major trends in the data. This provides some measure of validation for the models and the timing parameters incorporated into ACT-R/PM. However, these fits also depended on hand-tweaking of the basic visual features used to represent the icons, which will be discussed in the next section.

The correlation between the data matrices for both the text-look and double-shift models is 0.99. A couple of conclusions can be drawn from such a high correlation. One is that the slopes of the two sets of data are very similar. A more inferential conclusion is that the action times for the motor and vision modules in ACT-R/PM are quite accurate. In this experiment they mimicked the reaction times of actual subjects to a high degree of accuracy.

The root mean square error (RMSE) and percent average absolute error between the model and the data for the two charts shown previously are presented in the following table (Table 4.1). Note that the percent average absolute error in each of the above matrices of data remains in the remarkably low range of three percent—again indicating that the models were quite accurate in their performance in reference to the real subject data.

	Double-Shift Model	Text-Look Model
RMSE	125.41ms	112.28 ms
percent average absolute error	2.95%	3.19%

Table 4.1: Percent average absolute error. Model data as compared to real data.

4.4 Discussion of Model

Several points are about the performance of the model are worth discussing further here. One point that the model makes implicitly is the lack of the use of borders. The model does not use the icon borders in any way in its search strategy in locating the target icon. Despite this fact, the model still mimics the real performance of subjects very accurately. It could thus be inferred that real subjects do not make use of the border information either, or, at least, that the task can be carried out without utilizing the borders and still leaving performance relatively unaffected. Such an inference lends further confirmation to the conclusions regarding users' lack of use of icon borders drawn from the initial two experiments.

Based on the standard fit metrics, the fit of the models to the data is excellent for both models. The difference between the DS and TL models was very small, suggesting the increased number of attention shifts generated by the DS model were, in effect, cancelled out by the revisitation of the TL model. Based on the fits, there is no strong argument for preferring one model over the other, suggesting that strategy variation among users may not play a large role in determining search times. This was highly encouraging. However, there are some caveats. The first is that the feature lists used to represent each icon were generated *post hoc* to provide good fits of model to data. Ideally, we would like to have a principled way of constructing the feature lists on an *a priori* basis to make the models predictive rather than simply explanatory. This is an involved subject for future research.

The second caveat deals with an analytic prediction made by the DS model. It is worth clarifying the means by which the effect of quality was achieved in the model data. As previously mentioned, "quality," as used here in reference to icons alludes to the distinctiveness of the icons from one another. The icons are interpreted, "seen," by ACT-R/PM as a list of features. For example, a red circle will be "seen" as having the feature "circle" and the color "red." In contrast, more complex icons will have a number of features and colors associated with them, gray triangles, white circles, etc. What makes these more complex icons "poor" icons in the experiment is not the number of features the icon has per say, but rather the number of features the icon shares with other icons in the distractor set. For example, many of the icons in the poor quality set have gray triangles and white circles. It is this overlap of features, or "similarity," that makes such icons poor icons in context. On the other side, the good quality icons have no feature overlap with other good quality icons.

The reason this issue has been covered in some depth is because where there is no overlap, in the good quality icons, we can make some empirical predications about the performance of users and the performance of the model. Of particular interest is the performance of the model and the subject using high quality icons at the various set sizes. (This corresponds to the lowest, i.e. fastest, line in Figures 4.1 and 4.2.) One of the things that was meant to be a consistent value in the experiments was the number of icons matching the target icon (more on this subject later—our definition of equality had to change from pictorial equality to include functional equality, as well). In every distractor set exactly one-third of the icons were intended to be identical to the target icon (including the target icon). So for example, in a set size of six, where the target icon is a red circle, there would be two red circles in the display and the user would have to identify the target icon by reading the filenames. Such a system and the fact that we know the times that the model attributes to each of its actions allow us to make some quantitative predictions about the performance of the model. We can calculate how long on average it should take the model to find the target icon for each set size from six icons to twenty-four icons (1500 ms to 2760 ms). Additionally, we can compute the predicted slope of the line for good quality icons across the four set sizes, which, as discussed previously, is 420 ms—i.e. it should take the model an average of an additional 420 ms for every six icons that are added to the set size.

Here again, if the performance of the model matches that of users, this will provide additional evidence that the model's performance is predictive of users' performance. In fact, by making such quantitative predictions, we have become even more demanding in our criteria for what constitutes similar performance. Furthermore, because of the more stringent criteria, if the performance of the model approximates the behavior of real users, then we will be able to make stronger inferences regarding the search strategies of the users.

Alas, our model data matched that of the users quite well, but did not match our predictions. The slope for both the users *and the model* was approximately 460 ms, significantly steeper than the predicted 420 ms. Certainly, we could not assume to predict the performance of the users, but we should certainly be able to predict the performance of our own model. This was a surprise for us, and caused us to carefully re-examine the behavior of the model because it did not match our own analytic prediction. This led us to the discovery of an interesting point in the programming of the experiments. When the experiments were designed and programmed, we originally expected icon borders to play some significant role in the search strategies of users. One factor that we wanted to keep consistent in our design was the number of target-matching icons, icons identical to the target icon except with different filenames (one-third of the icons in the set). Thus, the user would ultimately be forced to identify the target icon based on its filename. However, because we assumed that icons with different borders would be seen as different than the target icon, we allowed icons matching the target icon except for the border to be included in the distractor set. However, as our experiments have shown, borders do not play a role in icon search, and thus, icons with different borders but with the same base pictorial icon may be seen as functionally the same icon. Because the distractor icons were chosen randomly, we occasionally, and randomly, allowed an additional icon (or icons) matching the target icon, beyond the predetermined one-third, to be present in the distractor set. The inclusion of this additional icon drove up reaction times for both users and the model, and was a cause of the steeper than predicted slope.

While this was certainly a flaw in the design of the experiment, it speaks to a strength of the approach employed. The ACT-R/PM models interact with the same experimental software as human subjects, and thus are sensitive to the same factors. It was therefore encouraging to see that the model was affected by this flaw as well, and we certainly would not have noticed this flaw had it not been for the (mis)behavior of the DS model. As a result, in order to examine the model's predictions, and directly compare its results with those of users, we needed to remove this element of randomness from our design, and a follow-up experiment was conducted which did just that.

5. EXPERIMENT 3

In Experiment 3, we were looking to get a clearer picture of the data. By removing the source of unpredictability in our program and, hence, a source of randomness or unexplained variance in our data, Experiment 3 was designed to get a "cleaner" picture of the data to allow for comparisons with a computational model.

5.1 Method

The design, procedures and apparatus of Experiment 3 were nearly identical to those of Experiment 2. The only factor that changed was the number of distractor icons in the distractor set that matched the target icon. In experiment two, distractor icons with the same base pictorial icon, but with a different border, were allowed to be randomly selected for the distractor set. In this experiment, icons with the same base pictorial icon as the target icon were excluded from the distractor set.

5.1.1 Subjects

The users in the experiment were 20 undergraduate students at Rice University who were participating in order to meet a requirement for a psychology course.

5.2 Results

User errors and outliers were removed from the data according to the same procedure as Experiments 1 and 2. In total, less than 5% of the trials were replaced due to errors and outliers. For statistical tests, where response times had been removed as errors or outliers, they were replaced with the user's grand mean.

A summary of the data, collapsed across the icon quality variable, provides results quite similar to those in Experiment 2. (Refer to Figure 5.1.) As in the first experiment, there is a significant effect of set size ,$F(3,57) = 178.56$, $p < 0.001$. Also, there is a reliable linear effect of set size, $F(1,19) = 282.35$, $p < 0.001$. Additionally, as in the first experiment, there is not a main effect of border type, $F(2,38) = 1.10$, $p = 0.34$.

In Figure 5.2, mean response times are presented as a function of set size and icon quality. Here again, as in Experiment 2, it is evident that as icon quality decreases (good to fair to poor), response times increase. This is confirmed by a significant main effect of quality, $F(2,38) = 58.71$, $p < 0.001$. Also, not only are the three qualities significantly different, but the slopes of the lines appear to be different, as confirmed by a reliable quality by set size interaction, $F(6,114) = 6.89$, $p < 0.01$.

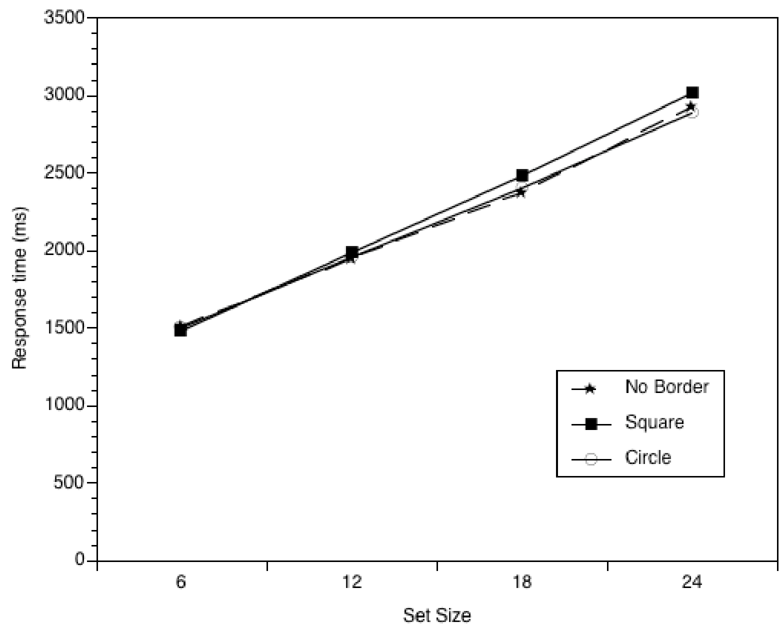

Figure 5.1 Mean reaction times by set size and border type.

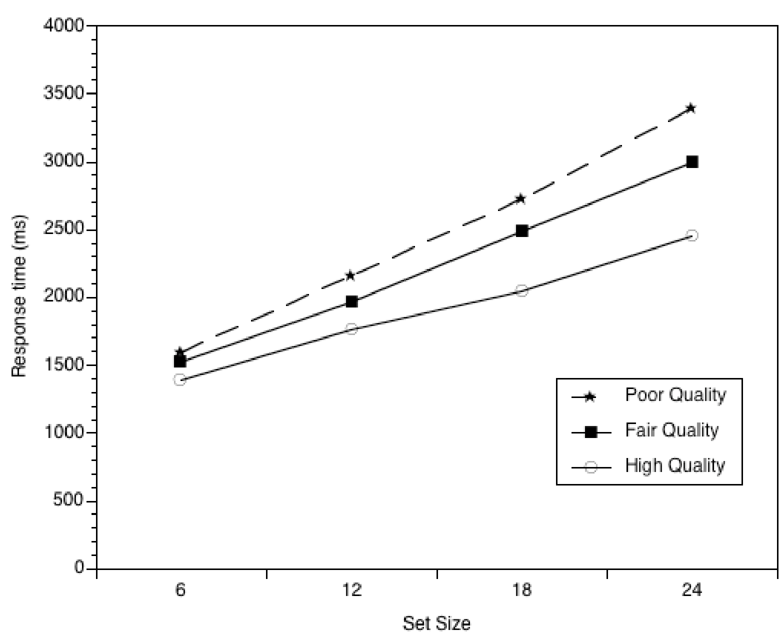

Figure 5.2 Mean response times by set size and icon quality, illustrating a main effect of icon quality.

In Figure 5.3, mean response times are presented across icon quality and target border type. This chart once again displays the main effect of icon quality—as quality decreases, response times increase. Here again, there is no effect of border type, and no interactions involving border type. Also, the difference in response times across the different icon qualities is relatively consistent, leading to a reliable linear effect of icon quality, $F(1,19) = 80.25$, $p < 0.001$.

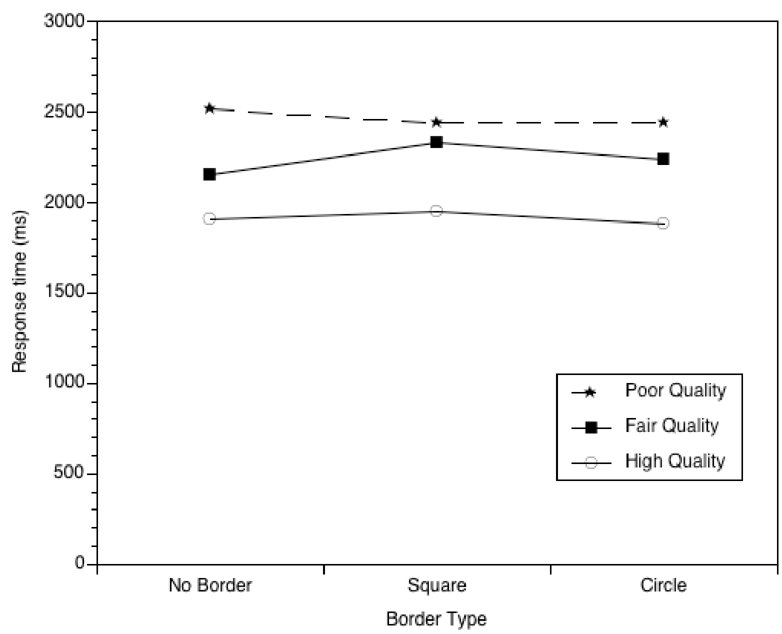

Figure 5.3 Response times by icon quality and border type.

5.3 Discussion of Experiment 3

Clearly, this experiment was able to produce the same general results from Experiment 1 for a third consecutive time. Again the experiment produced the effects of icon quality and set size and a lack of an effect of icon borders. Such reproduction allows us to draw our conclusions with even greater confidence.

The particular aspect of this experiment that we were most interested in was not, however, these broader effects. We were interested in closely examining the effect of set size on the high quality icons. Specifically, we wanted to study the slope of the graphical representation of this subset of data and determine if it approximated 420 ms, the value predicted by our computational model.

At first glance, the slope of the line appears to be shallower than 420 ms. The specific data points that make up the line are presented in Table 5.1 below. The average slope for the four data points was calculated as 355.08, quite a bit lower than 420 ms.

Set Size	6	12	18	24
RT (ms)	1385.97	1765.62	2042.21	2451.19

Table 5.1 Mean Reaction Times for Good Quality Icons across set size.

In order to compare the two different slopes more thoroughly, an analysis using confidence intervals was conducted. The slope for each of the 20 subjects was calculated. Then, the 95% confidence interval for the "good quality slope" based on the slope of the 20 subjects was determined. The upper and lower limits for the confidence interval were 392.06 ms and 316.70 ms, respectively. Clearly, 420 ms does not fall within the confidence interval.

In this instance we have found a segment of the model that does not conform well to the data. Our model, in fact, is too slow. Real subjects can find the icon faster than our model can under optimal conditions of no feature overlap. The solution then is to change the model, but that is certainly easier said than done. There are numerous ways that the strategy of real subjects may differ from the strategy implemented by the model. For example, one possibility stems from this experimenter's intuition. Having done the task many, many times, it seems that I look first at a group of icons that match the target icon and begin my search through the filenames there. In this manner, I do not follow a simple left to right or random strategy of examining icons that match the target icon. Unfortunately, my own intuition is hardly empirical data on which to base a model. It would make sense then to investigate what the real strategies of users are in the icon search task. In order to do this, one would have to actually study users eye movements while engaged in the task, accomplished through the use of an eye tracker.

6. EYE TRACKING AND EYE MOVEMENT STUDIES

The issue we would like to examine is that of the search strategies employed by users during icon search. As mentioned, the logical way to go about studying these strategies is through empirical data obtained through the use of an eye tracker. However, before we go into the specifics of such a study, we want to know that eye tracking will give us the answers that we seek. A brief review of some of the relevant eye tracking and eye movement literature is provided to give a basis for how other findings might relate to our research, and what kind of "answers" we might expect to find using the eye tracking technique.

By tracking eye movements researchers have been able to account for many of the factors that determine where our eyes move. These movements have been analyzed at the level of individual eye saccades.

In one influential eye tracking study, Findlay (1981) directly examined the concepts of "global processing" and a "center of gravity." The term "global processing" refers to the idea that the information being used in the calculation of saccade amplitude is obtained by integrating information over a large spatial "window." In this study, Findlay conducted four experiments in which saccadic eye movements were examined when the eye moved to targets in peripheral vision, which consisted of two discrete stimuli. It was found that under a variety of conditions, the saccade amplitude is such that the saccade lands at an intermediate position between the stimuli. In Findlay's words, "The effect depends in a systematic quantitative manner on the properties of the visual stimuli. This may be loosely described by saying the saccade is directed to the centre of gravity of the target configuration." For example, in cases where one target was large and one small, the saccade landed closer to the large target. (with slightly more "weight" given to the target nearest to the previous fixation). Deubel et al. (1984) extended Findlay's findings by showing that increasing the relative intensity of one of the two stimuli has an effect similar to increasing its size.

In an earlier study, Williams (1966, 1967) used eye tracking to determine the features of objects that allow for efficient search by examining eye movements during search of a set of simple geometric forms which varied in color, shape, and size. He measured search time and recorded the pattern of eye movements in a number of conditions, which differed in the amount of prior specification of the features of the target. For example, the subject might be told that the target was red, or was a circle. He compared these with a baseline condition in which no advance knowledge of the target was given. He found that specification of the color of the target object speeded the search process considerably and moreover the majority of eye fixations fell on non-target items of the specified color. Specification of the size of the object was much less effective; specification of the shape provided almost no advantage. There was a close correspondence between the ability to direct the eyes to the pre-specified items and the speed of the search. The details of Williams' finding appear to be task specific since other studies (Gould and Dill, 1969; Viviani and Swensson, 1982) have demonstrated the ability to use shape information to direct eye movements. However, Williams demonstrated that in some search tasks saccadic eye movements can be made directly to targets whereas in others non-targets are also scanned. This distinction anticipates the current distinction between parallel and serial search.

Findlay's (1981) results are broadly consistent with other ideas on the operation of the human saccadic system. Becker and Jurgens (1979) suggested that two channels are involved in the process of generating a saccade. One of these, the "when" system, determines the instant at which the saccade is initiated; the other, the "where" system, determines the spatial characteristics of the saccade. We might suppose that if the "when" system triggered the saccade too early, then the "where" calculation would be interrupted before the target had been completely perceptually isolated from its surroundings, and the saccade computation would remain influenced by the whole configuration surrounding the stimulus.

Coeffe and O'Regan (1986) were able to examine Becker and Jurgens' hypothesis by showing in a set of experiments that the "global effect" can be attenuated when the "where" calculation is facilitated or when the moment of saccade triggering is sufficiently delayed. They suggest that the exact position where the eye lands can be calculated from a linear combination of two tendencies: a tendency to saccade to the aimed-for target, and a tendency to saccade to a "gaze attraction position," which roughly approximates to a less eccentric version of Findlay's "centre of gravity."

Zelinsky and Sheinberg (1997) did some further work investigating the oculomotor behavior accompanying parallel–serial visual search. Two experiments using a basic visual search paradigm were conducted (one using “O”-like and “Q”-like stimuli and the other using colored and oriented bars) and eye movements were recorded as participants searched for a target in 5-item or 17-item displays. Results indicated the presence of parallel–serial search dichotomies and confirmed that such a dichotomy is reflected in oculomotor behavior.

A slightly different use of eye tracking techniques to understand visual search comes from a number of studies from Motter and Belky. In one study, they analyzed the eye movements of monkeys to investigate the zone of focal attention during active visual search (Motter and Belky, 1998). They examined target detection performance in two monkey subjects during visual search with eye movements as a function of stimulus density and the eccentricity of the target from fixation. As expected, search time and the number of fixations per trial were related to the number of objects in the set. Additionally, they looked at the factors that influence the size of the area surrounding the fixation point, where there was a high probability of detection of the target (the zone of focal attention). The density of stimuli in the visual scene, as measured by average nearest neighbor distances, proved to be one key factor. In general, they found that focal attention operates within a conspicuity area having an effective radius of about twice the average nearest neighbor distance. The size of this area was also found to be largely determined by target eccentricity.

In a second study, Motter and Belky (1998b) investigated the features of objects that guide eye movements during active visual search. The task for the monkeys in this classic conjunction and feature search paradigm (based on color and orientation features) was to find and fixate a target in an array of stimuli. It was found that monkeys used target color, but not orientation, to selectively guide visual search.

7. EYE TRACKING THE ICON SEARCH TASK

A few general principles can be gathered from these eye movement and visual search studies that will likely relate to studies of the icon search process. Clearly, eye tracking allows researchers to make fine distinctions regarding the process used in a visual search task. For example, researchers were able to identify oculomotor distinctions between parallel and serial search processes and even develop a model that accounts for the global effect of the visual environment in making saccades. Also, researchers have used eye tracking to gather information on the features of objects that drive visual search. If these two types of information can be gleaned by implementing our icon search task in an eye tracking paradigm, then such a study would certainly be beneficial to our overarching goal of understanding the icon search process.

As it was probably expected, the use of eye tracking has made its way into studies of human-computer interaction and as a potentially applied procedure in the computer industry. On the applied side, a number of researchers have investigated and are continually developing "gaze-based" interfaces in which a user controls the computer using his/her eye movements (Sibert and Jacob, 2000; Salvucci and Anderson, 2000). On a more academic level, it has been used as a means of understanding the processes underlying the behavior of computer users (Byrne, 2001). An understanding of the icon search process at the saccadic level would be beneficial in both of these areas.

The next step in our research process was an eye-tracking study of the icon search task. The study was designed to provide some insight into the processes underlying icon search, i.e. the sequence of steps a subject goes through when searching for and identifying a target icon. For example, does the subject look at the icon and then its corresponding filename (as the Double-Shift model does), or does he/she even need to directly examine the icon?

7.1 Methods

7.1.1 Design

The design of the experiment was nearly identical to that carried out in Experiment 3 previously discussed in this paper with the additional use of an eye tracker to record subjects' eye movements.

7.1.2 Apparatus/Materials

The eye tracker used was an ISCAN RK726/RK520 HighRes Pupil/CR tracker with a Polhemus FASTRACK head tracker. Head-mounted optics and a sampling rate of 60 Hz were used in the experiment. This system, like many other laboratory eye trackers, works by shining an infrared light on the eye and taking a video image of the eye. From that image, it is possible to determine the pupil center and the point on the cornea closest to the camera (the corneal reflection) and take the vector between them. This vector changes as the eye orients to different positions on the screen, and with calibration to known points, it is possible to compute visual point of regard (POR, also referred to as "point of gaze"). The magnetic polhemus is used to compensate for head movements. POR reports by the eye-tracking equipment are typically accurate to within one-half degree of visual angle.

POR and mouse position were recorded approximately every 16.7 ms by the experimental software. Stimulus and POR/mouse data for each trial were recorded so that all individual trials could be "replayed" at various speeds. An experimenter monitored each experimental trial and recalibrated the eye tracker if there appeared to be a sizable disparity between reasonable expectations about where users were looking (in particular, users needed to look at the target on each trial) and the position reported by the tracker.

7.1.3 Analysis Technique

From the raw data it is be possible to compute where and when fixations occur. This can be done either by assuming that any eye position within a given region for more than some threshold number of milliseconds is a fixation (dwell-based) or assuming that any period of time showing relatively low velocity is a fixation (velocity-based). For the data set, both methods were initially used and examined to verify that they both yield approximately the same result. Mouse data were treated similarly; that is, post-processing analysis was used to identify the number and location of mouse "fixations" for each trial.

7.2 Results

The average number of fixations per trial are plotted as a function of icon quality and set size in Figure 7.1. Patterns in the fixation data were similar to those found in the reaction time data from the previous experiments—i.e. as set size increases and icon quality decreases, the average number of fixations increases (as does response time). There were reliable main effects of set size, $F(3, 27) = 77.08$, $p < 0.001$, and icon quality, $F(2, 18) = 56.60$, $p < 0.001$. As indicated by these results, the fixation data was highly correlated with the response time data from Experiment 3, $r = 0.73$, $p < 0.01$.

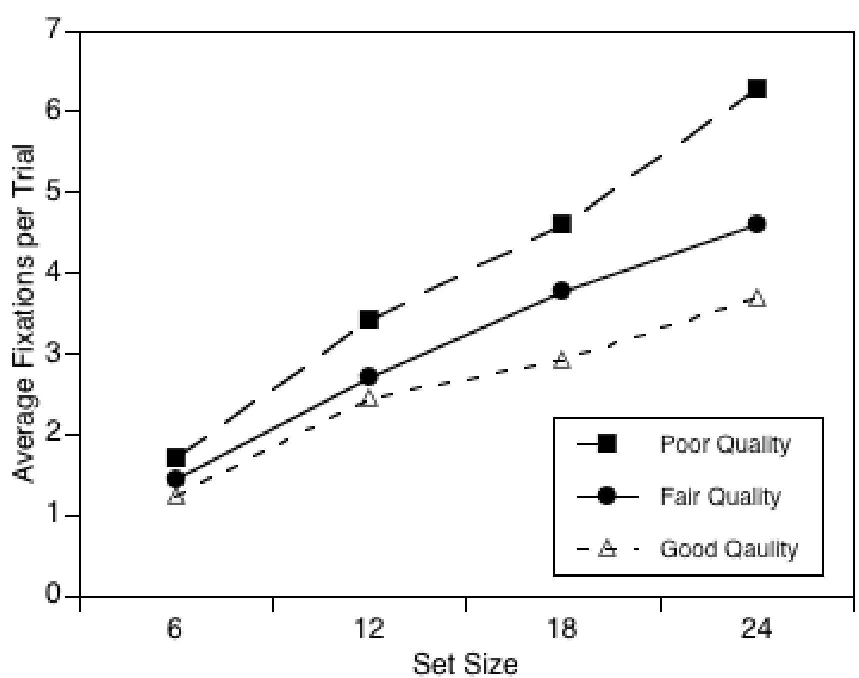

Figure 7.1. Average number of fixations per trial by Icon Quality and Set Size.

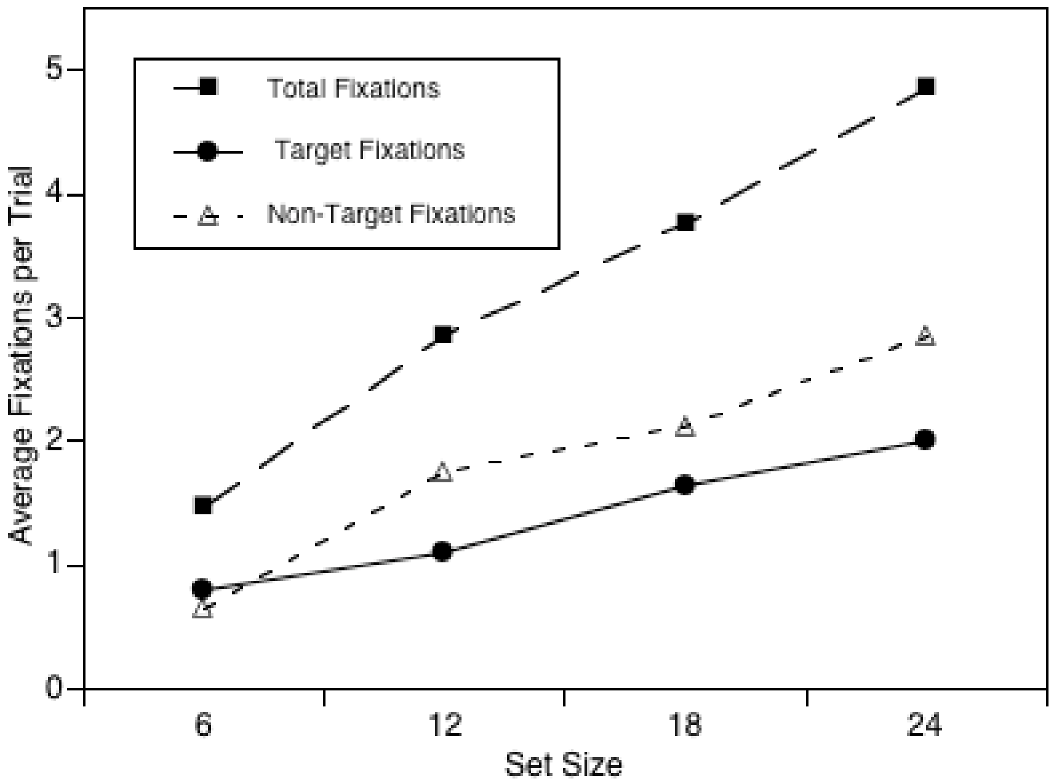

Figure 7.2 Average number of fixations per trial by type of fixation and set size. Target fixations are those fixations to an icon exactly matching the target icon, which was one-third of the icons in each distractor set.

In Figure 7.2, the average number of fixations is presented as a function of type of fixation and set size. “Target” fixations are those fixations to an icon exactly matching the target icon, which was one-third of the icons in each distractor set. “Non-target” fixations are fixations to any other icon in the distractor set. Fixations to areas outside of the distractor set of icons were excluded from the analysis. An ANOVA was conducted to examine the relationship between the target fixations and the non-target fixations. Here again, a reliable effect of set size is evident in the data, $F(3, 16.87) = 85.02$, $p < 0.001$ according to the Greenhouse-Geisser correction. Also, there is a reliable effect of type of fixation (target v. non-target), $F(1, 9) = 172.31$, $p < 0.001$. A reliable interaction between set size and type of fixation was also found, $F(3, 27) = 74.00$, $p < 0.001$, indicating that participants made a higher proportion of target fixations at the smallest set size. If participants made randomly directed fixations, then we would expect one-third of their fixations to land on icons matching the target icons, which compose one-third of the distractor set. A t-test was conducted to examine if the proportion of target fixations differed from the "expected" proportion of one-third. Participants made a higher proportion of fixations to target-matching icons than would be expected if fixations were randomly directed, $t(3) = 7.33$, $p < 0.01$.

In Figure 7.3, the proportion of target fixations to total fixations is presented as a function of icon quality and set size. Participants had a higher proportion of target fixations relative to non-target fixations with better quality icons, $F(2, 18) = 7.87$, $p < 0.01$, according to the Huynh-Feldt correction.

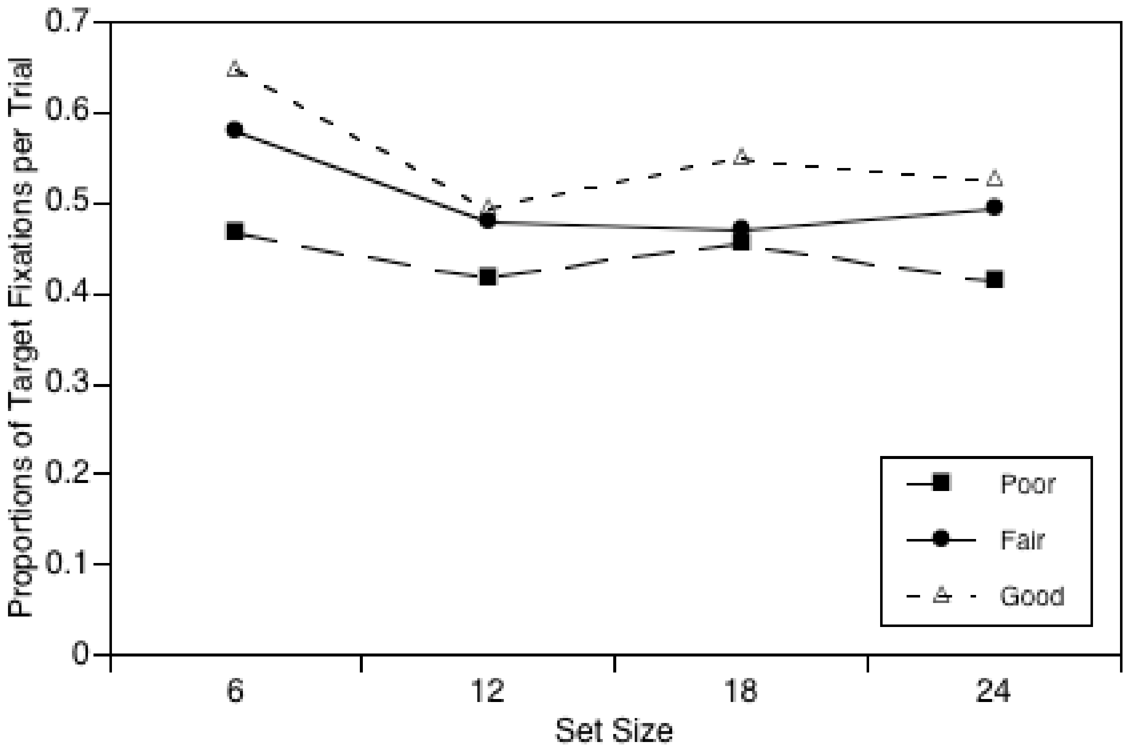

Figure 7.3. Proportion of target fixations to total fixations by icon quality and set size.

Because the models make several predictions regarding the reaction time of participants in the good icon quality condition, data pertaining to this condition are examined separately and in greater depth (See Figure 7.4). Three different “types” of fixations were examined. Target fixations are fixations on an icon exactly matching the target icon. Color fixations are those fixations on an icon of the same color as the target icon, regardless of shape. Shape fixations are fixations landing on an icon of the same shape as the target icon, regardless of color. An ANOVA was conducted to examine the proportion of target fixations relative to color fixations, $F(1, 9) = 67.53$, $p < 0.001$, indicating that there is a reliable difference here. This effect would be expected even if the non-target fixations were randomly placed because some of the non-target icons were icons of the same color as the target icon.

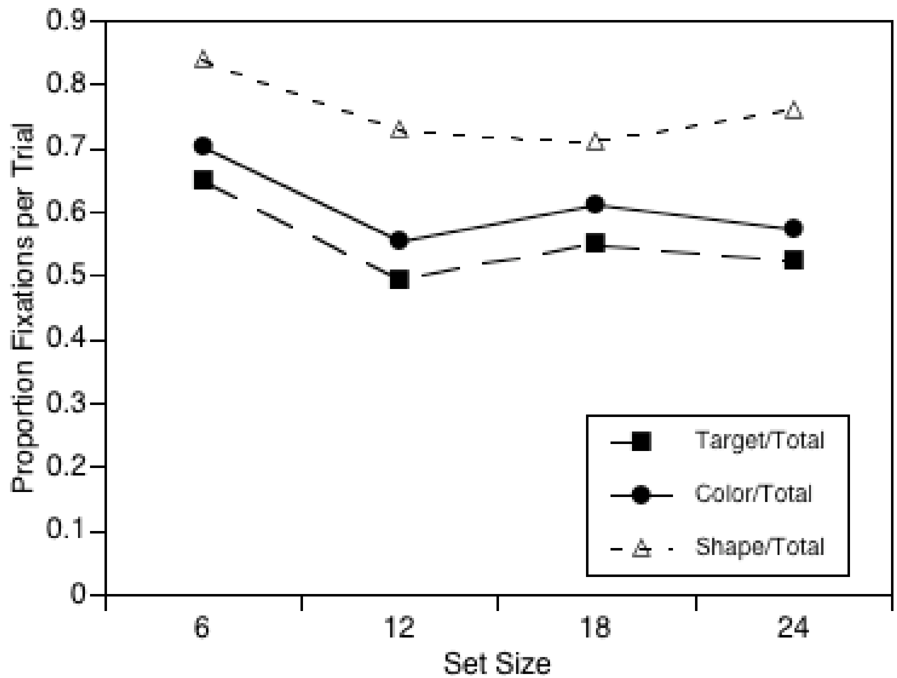

Figure 7.4. Proportion of specific fixations to total fixations. Target fixations are fixations on an icon exactly matching the target icon. Color fixations are those fixations on an icon of the same color as the target icon, regardless of shape. Shape fixations are fixations landing on an icon of the same shape as the target icon, regardless of color.

It would be more informative to examine whether the proportion of non-target fixations that were directed at an icon of the same color (or shape) as the target was reliably different than the proportion that could be expected if the fixations were random (Figure 7.5). The proportion of non-target color fixations that would be expected with random fixations is 1/11 or 0.09. (Each non-target icon was randomly selected from the entire corpus of icons of the same level of quality excluding the target icon with the three different borders. 3/33 = 1/11.) With shape, if non-target fixations are randomly directed, we would expect the proportion of non-target fixations directed at icons of the same shape to total non-target fixations to be five to eleven or 45%. A matched samples t-test was applied to examine the difference between this expected proportion of fixations (if the fixations are randomly directed) and the observed proportion of fixations. For color fixations there is an indication that the search strategies of participants were at least in part driven by the color of the icon, $t(3) = 3.26$, $p < 0.05$. For shape fixations, there is no such indication, $t(3) = 0.20$, $p = 0.86$.

Several qualitative patterns emerged in the data, which are difficult to quantify, but are nonetheless informative. For one, it seemed that participants in the experiment used different search strategies depending on the level of quality of the icons. For instance, in the good quality condition (Figure 7.6), the search strategy used by participants was often directed specifically at the icons matching the target icon. In this case, the saccades were nearly all directed to a target icon or fell in the area between two groups of target icons, leaving whole areas of the distractor set unexamined. Further, this "directed" strategy often began with the largest group of matching icons and proceeded to smaller groups of matching icons until the target was identified. In contrast, search strategies in the poor quality condition were not directed at icons matching the target icons and might cover the whole set of icons, possibly in a circular or zigzag pattern (Figure 7.7).

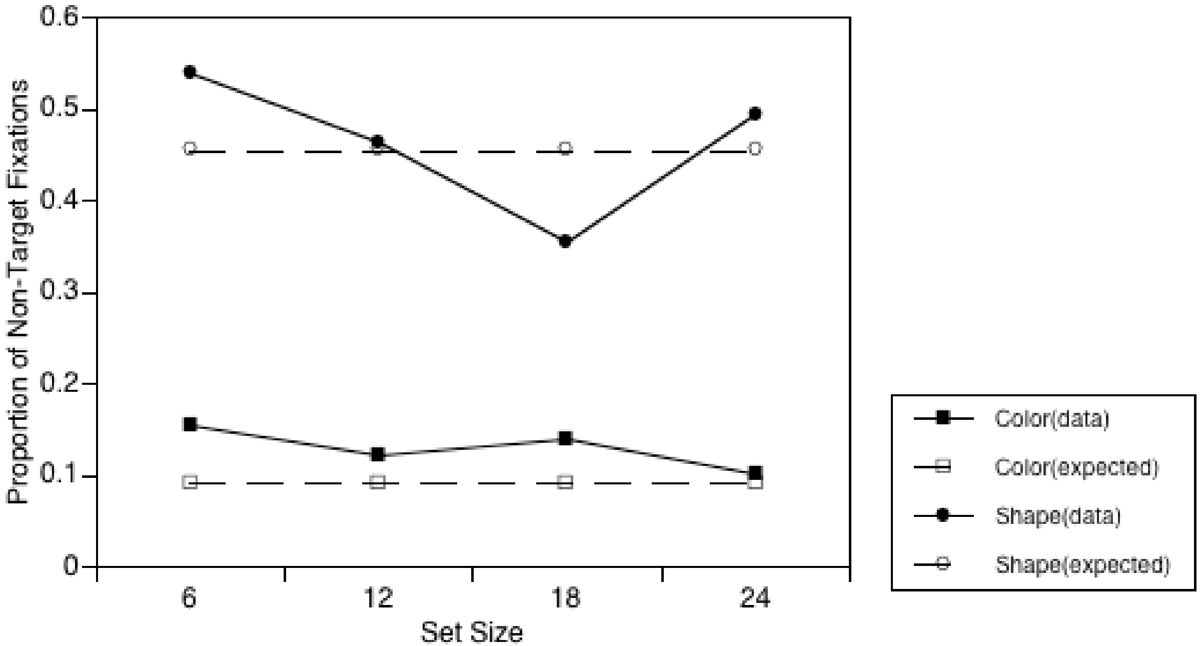

Figure 7.5 Proportion of non-target fixations directed at icons matching the color or shape of the target icon to the total number of non-target fixations. Also presented are the expected proportion of color and shape fixations to total non-target fixations if non-target fixations are randomly directed.

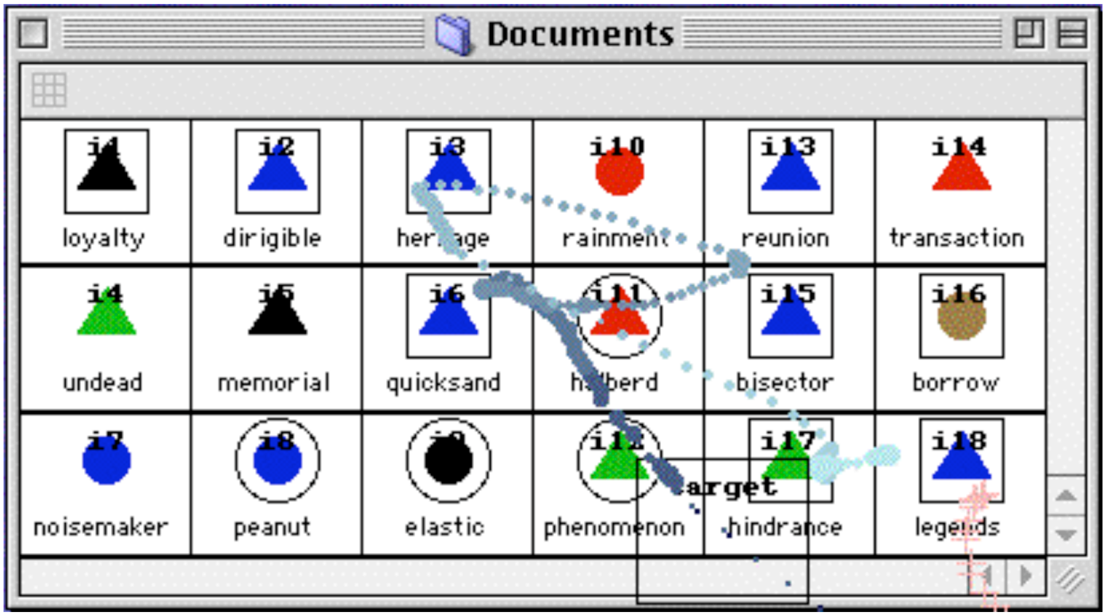

Figure 7.6 Example of a directed search with good quality icons. The round dots indicate point of regard, going from darker to lighter with time. The cross-hairs (in the lower right) indicate the position of the mouse. Note that the participant only examines icons matching the target icon, beginning with the largest group of matching icons and eventually proceeding to the single matching icon (which is the target in this case) in the lower right.

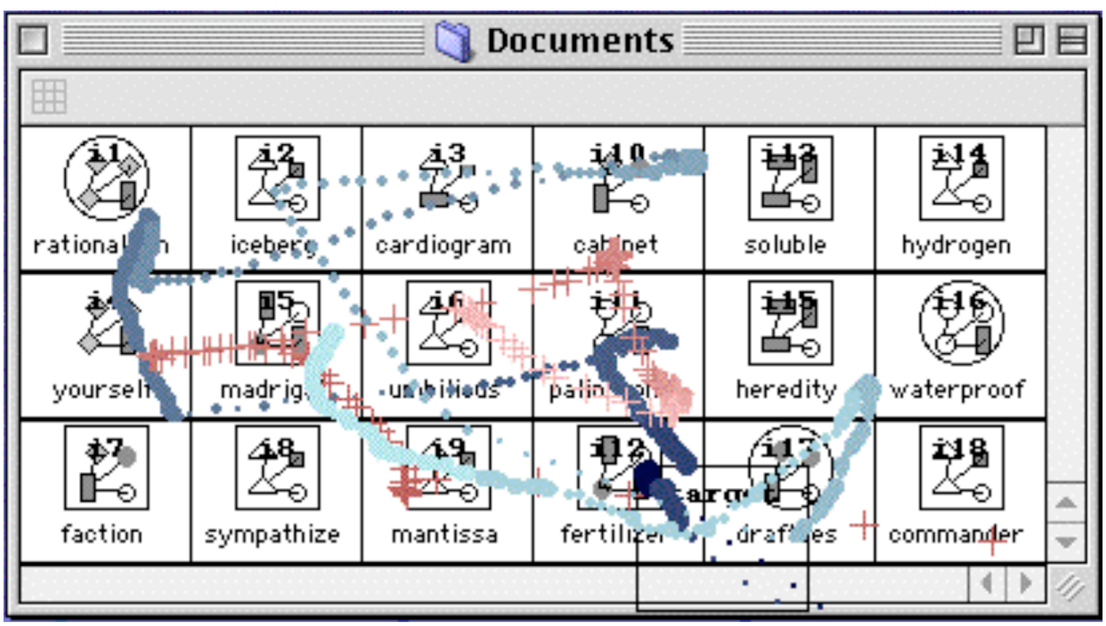

Figure 7.7. Example of an undirected search with poor quality icons. Following the dots, which indicate point of regard, from dark to light (with time) indicates that the subject examined nearly the entire set of icons in a zigzag manner.

7.3 Discussion of Eye Tracking

The eye tracking data revealed a strong correspondence with the reaction time data from the previous experiments. As we saw increases in reaction time with increases in set size and decreases in icon quality, we saw corresponding increases in the number of fixations per trial under the same conditions.

As for the accuracy of the fixations, the data revealed that participants were more accurate in their search with better quality icons. This effect was manifested in the proportion of target fixations to total fixations, which increased with each level of improvement in icon quality. There was also some evidence for this effect at a qualitative level, manifested in the “directed” search strategies in the good quality icons and the “undirected” search strategies seen with poor quality icons.

The eye tracking data also provided us with some data as to which features of the good quality icons were used by participants to guide their “directed” search. Participants made a higher proportion of fixations to non-target icons of the same color as the target icon than would be expected if non-target fixations were randomly directed, indicating that color is a feature guiding search in the good quality condition. Participants did not make a higher proportion of non-target fixations to icons of the same shape, i.e. circle or triangle, as the target icons. Hence, there is no evidence that shape is a guiding feature of the participants’ icon search strategy in the good quality condition.

8. REVISING THE MODEL

Our original impetus for the eye tracking study was to examine the icon search strategies of computer users in some detail. Our previous ACT-R/PM models were less efficient, in terms of response time to find and select a target icon, than participants in our experiments. Our goal in the eye tracking study was to examine where our model strategies fell short of those of users in terms of search efficiency. This section outlines some of the potential revisions to the models suggested by the results of the eye tracking study.

8.1 Double-Shift vs. Text-Look (Accuracy of Fixations)

One of the questions we turned to the eye tracking study to answer was whether the Double-Shift or the Text-Look model was a better approximation of the strategy employed by participants. It was clear from looking at the search patterns of participants (Figures 7.6 and 7.7) that they did not need to look directly at an icon to determine whether it was the target filename. This is evidence in favor of the Text-Look model, which does not shift visual attention to each icon, but only to the filename below it. However, even the Text-Look model falls short of the strategies of the experiment participants, who often do not even look directly at a filename to determine whether it is the target filename. (The only instance where participants did consistently foveate an icon was when they were moving the cursor to select it.)

This data alone would suggest that future iterations of our modeling efforts be focused on the Text-Look model or other implementations that examine both an icon and its filename in a single shift of visual attention.

8.2 Number of Fixations

The inaccuracy of the location of where the models shift visual attention relative to the actual search patterns of participants may be tied to another area where the performance of our models does not match that of real computer users—the total number of fixations. It is clear from Figure 8.1 that both of the models make far too many shifts of visual attention. Again, the Text-Look model is a closer approximation of the participants' performance; however even the TL model makes substantially more overt shifts of visual attention than participants.

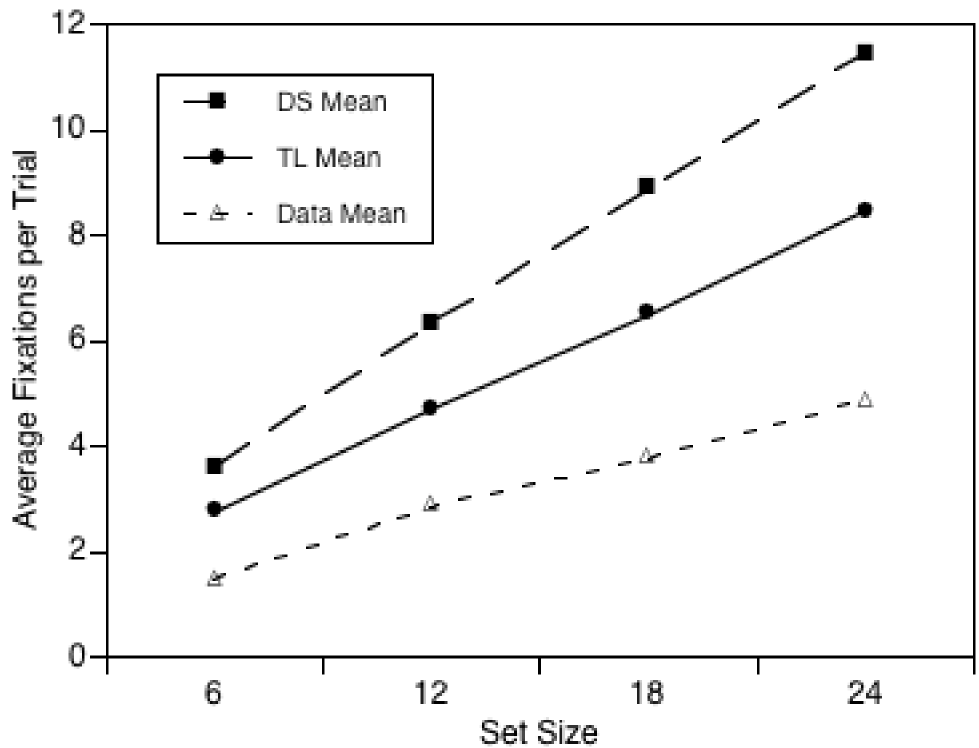

Figure 8.1 Mean number of shifts of visual attention made by the Double-Shift (DS) model, the Text-Look (TL) model relative to the number of fixations made by participants in the eye tracking study (Data) as a function of set size.

This leads us to consider an issue in the underlying cognitive architecture of ACT-R/PM that other authors have discussed previously (Salvucci, 2000). ACT-R/PM currently assumes a direct correspondence between unobservable attention shifts and observable eye movements; that is, people fixate the target of attention. Such an assumption holds in some cases, but it is agreed upon in the research community that it does not hold in general (Henderson, 1992; Rayner, 1995). The experiments modeled here may provide an example of one such case where this does not hold. Therefore, in order to model the experiment accurately in ACT-R/PM, some of the underlying assumptions of the Vision Module need to be improved upon. Fortunately, there already exist computational models that serve as a bridge between observable eye movements and the unobservable cognitive processes and shifts of attention that produce them, such as EMMA (Eye Movements and Movements of Attention) developed by Salvucci (2000).

8.3 Eye Movements and Movements of Attention (EMMA)

EMMA serves as a bridge between observable eye movements and the unobservable cognitive processes and shifts of attention that produce them. Concerning eye movements, the model describes whether or not eye movements occur, when they occur, and where they land with respect to their targets. Concerning visual encoding, the model describes how peripheral viewing and object frequency affect the time needed to encode a visual object into an internal representation.

With respect to visual encoding, EMMA uses a “spotlight” metaphor of visual attention that selects a single region of the visual field for processing. When cognition requests a shift of attention to a new visual object, EMMA encodes the visual object into an internal representation.

The time T_{enc} needed to encode object i is computed as follows:

$$T_{enc} = K\,[-\log f_i]\, e^{k\varepsilon_i}$$

The parameter f_I represents the frequency of the object encoded, specified as a normalized value in the range (0,1). The parameter ε_i represents the eccentricity of the object, measured as the distance from the current eye position to the object in units of visual angle. Thus, encoding time increases as object eccentricity increases and as object frequency decreases. The constants K and k scale the encoding time and exponent, respectively. To reflect variability in the system, the model assumes that encoding time is distributed as a gamma distribution with mean T_{enc} and standard deviation equal to one-third the mean encoding time. The computation of T_{enc} is based on the E-Z Reader model of eye-movement control in reading (Rayner, Reichle, & Pollatsek, 1998; Reichle et al., 1998).

In EMMA, a shift of attention initiates an eye-movement program to the attended object. The eye movement runs through two stages, preparation and execution (cf. Byrne & Anderson, 1998; Kieras & Meyer, 1997). Preparation represents the retractable, or "labile," stage of the eye-movement program—i.e. it can be canceled if cognition requests another preparation before the first terminates. Execution includes both non-retractable programming and the eye movement itself. The motivation for a separation of labile and non-labile programming comes from results showing that eye movements can be canceled only within a certain time threshold after the initiation of saccadic programming (Becker & Jurgens, 1979). The completion of a saccade initiates preparation for a new saccade to the same visual object in the event that encoding has not yet completed.

EMMA describes both the temporal and spatial characteristics of eye movements. With respect to temporal characteristics, the model includes two parameters T_{prep} and T_{exec} that describe the time required for preparation and execution, respectively. Preparation time T_{prep} is estimated based on the direction and distance of the saccade relative to the previous saccade (if the previous saccade was in the same direction and of the same distance, then T_{prep} decreases correspondingly). For our purposes, T_{prep} will generally fall in the range of approximately 100 ms. Execution time T_{exec} includes 50 ms for non-labile programming (cf. Becker & Jurgens, 1979), 20 ms for saccade execution, plus an additional 2 ms for each degree of visual angle subtended by the saccade (Fuchs, 1971). The total time to prepare and execute a saccade closely resembles saccade latencies of approximately 200 ms cited in many previous studies (e.g. Anderson, Matessa, & Lebiere, 1997; Fuchs, 1971; Russo, 1978). Again, the model adds variability by assuming that T_{prep} and T_{exec} are distrubuted as a gamma distribution with a standard deviation equal to one-third the mean.

With respect to spatial characteristics, EMMA provides a simple formalization of where eye movements land with respect to a desired destination. Given a saccade to a particular object, the model assumes that the landing point follows a Gaussian distribution around the center of the object. The distribution is given a standard deviation of 0.1 times the distance from saccade origin to intended destination as has been estimated empirically (Salvucci, 2000).

The control flow of the EMMA model describes how cognition, visual encoding, and eye movements interact as interdependent processes. When cognition requests an attention shift to a new visual object (such as a new icon in our paradigm), EMMA begins encoding the object while an eye movement is prepared and (possibly) executed. Depending on the order in which the processes complete, various scenarios arise in their interaction.

In the simplest case, encoding requires the same amount of time as an eye movement. In this case, the visual-encoding module works on encoding the object while the eye –movement module runs through each of its two stages.

Another two cases arise when encoding completes and cognition requests a subsequent shift of attention before the original eye movement has completed. If the attention shift occurs during eye-movement preparation, the eye movement is canceled and a new eye movement is begun. If the attention shift occurs during eye-movement execution, execution continues to run to completion while preparation for a new eye movement is begun.

If the eye movement completes before encoding completes, encoding continues and a new eye-movement is prepared. However, because the eye movement has (presumably) brought the fovea nearer to the visual object, encoding speed increases accordingly. This aspect of the model helps to account for behavior when objects are distant from the fovea and encoding time is very long, since the model can refixate objects to decrease eccentricity and facilitate encoding (Salvucci, 2000).

<u>8.3.1 Incorporating EMMA</u>

Where the modeling of the icon search process is concerned, we incorporated EMMA to improve the performance of the models in those areas that the eye tracking study indicated were ripe for improvement, specifically the number and location of fixations and the timing of movements of visual attention.

As noted previously, our models made too many shifts of visual attention relative to the number of fixations made by participants in the eye tracking study. With EMMA, the number of shifts of visual attention will not decrease; however, because eye movements do not directly correspond with shits of attention in EMMA, we could expect the number of eye movements, or shifts of POR, to decrease. When the encoding time for a visual object is less than the time to make the eye movement to the object, then the eye movement is not made, even though the object has been examined.

In the Text-Look model, we expected to see less of a drop in shifts of POR than in the Double-Shift model. Shifts of visual attention only occur in the TL model from one filename to another filename, whereas shifts in the DS model also occur from an icon to the filename below it. The distance from an icon to a filename is consistently less than the distance from icon to icon. As mentioned, the encoding time calculated in EMMA is a function of the eccentricity of the target object from the current location. When this eccentricity, or distance to the target object, is smaller, which will occur more often in the DS model, we expected to have more opportunities for the encoding time to be less than the execution time. In each instance where the encoding time is less than the labile portion of the execution time, the shift of POR will not be made, resulting in one less fixation.

In addition to seeing a decrease in the number of shifts of attention made by the models, we also expected see increasingly similar patterns of the location of shifts of attention relative to the fixations of participants. Although visual attention will be focused on an icon selected by the model, the actual point of regard (POR) calculated by EMMA will be based on a Gaussian distribution around the center of the object. Thus, the model will not always shift its POR directly to the center of an object. However, when visual attention remains focused on an object for an extended period of time, this will allow the model to make successive shifts in its POR, each one presumably more accurate than the last (i.e. focused on the center of the target object). This will occur due to longer encoding times for the object or because cognition has not requested an additional shift of visual attention, as when the model is making a mouse movement to the object. As noted previously, one of the instances where participants consistently foveated the target icon was when selecting it with the mouse; this behavior is predicted by EMMA.

We also hoped that the incorporation of EMMA into our modeling efforts would provide greater predictive power of our models regarding the timing of fixations. Our previous models used a fixed estimate of time to make a shift of visual attention of 135 ms, the default value in ACT-R/PM. However, there is a large body of evidence that suggests that the time to make a saccade is a function of a number of different factors, one of the most influential of which is the eccentricity of the target object (e.g. Fuchs, 1971; Russo, 1978). Because the eccentricity of the object is taken into account when calculating the time to make a shift of the POR in EMMA, the incorporation of EMMA will allow us to make predictions as to the relative efficiency of various icon search strategies based on the average length of shifts of visual attention. Specifically, strategies that make shorter shifts of visual attention can be expected to be more efficient.

8.4 Improving the Model Search Strategies

Our goal was to incorporate some changes into the search strategies used by the model based on the results of the eye tracking study. For example, there was some qualitative evidence for a "grouping strategy" in the eye tracking study, particularly with good quality icons. As outlined previously (refer to Figure 7.6), when a group (or groups) of icons matching the target icon were present in the distractor set, participants would often begin their search for the target icon within the largest group, then proceed to subsequently smaller groups. Such a strategy is difficult to implement in ACT-R/PM largely due to the fact that it is difficult to determine on an a priori basis what constitutes a "group" in a display as complex visually as the distractor set. As a result, we chose to implement a simpler strategy in our new models. The model would simply select the target-matching icon (an icon sharing the pertinent feature with the target icon, functionally "seen" be the model as the same type of icon as the target icon) nearest to the icon that is the current focus of visual attention. Thus, if examining an icon in a group of target-matching icons, the model will examine all of the icons in the group before moving on to the next group. However, such a strategy does not guarantee that the largest group will be examined first, the second largest next, and so on. Such a strategy also ties in with the predictions made by EMMA regarding a decrease in saccade time with a decrease in the eccentricity of the subsequent shift of visual attention. Specifically, a strategy that makes the shortest possible shift will be the most efficient strategy.

Using a strategy such as the “nearest first” strategy just described is only one of many possible improvements to the models inspired by the eye tracking study. The inclusion of this strategy, EMMA, and other modeling changes may improve our models to resemble the behavior of participants more accurately, in terms of reaction time and eye movement patterns. We can now turn our discussion to the implementation of these improvements in the following section.

9: THE REVISED MODELS

The basis for our new models was the Text-Look and Double-Shift models we worked with previously (See Section 4.1 for a description). Although the eye tracking data provided slightly more evidence in favor of a strategy approximating the TL model rather than the DS model, we wanted to continue to look at the DS model for a several of reasons. For one, we believed that EMMA may have a greater impact on the DS model, because of the larger number of short saccades, which will result in a model that is functionally similar to the TL model in terms of the number of overt shifts of visual attention. Also, one aspect of the DS model we found very useful in the previous iterations was the fact that we could make analytic predictions regarding its performance, and we wanted to retain that capability.

9.1 Changes to the Models

Several changes were made to the models from the previous studies, and where possible, the effect of each of these changes on the models will be analyzed and discussed separately. Some discussion of how the changes were implemented will be provided here.

As mentioned, a major change to the modeling arena was the addition of EMMA. Fortunately, the actual implementation of EMMA within the ACT-R/PM framework was fairly simple to accomplish. EMMA was originally written to interact with the vision module of ACT-R/PM, so implementing EMMA with our models simply involved adding this code to our current modeling set-up.

Implementing the strategy change where the models shifted visual attention to the nearest icon matching the target icon was also relatively easy to implement. The capability to reference visual locations that are "nearest" to the current visual location was already built in to ACT-R/PM. Adding this capability to our models simply involved adding another criteria to the left-hand side of a production that was responsible for shifting visual attention to a new location. The left-handed side of the relevant production (named "Focus-Icon" in Appendix B) previously matched to any icon on the screen that had a feature (which was stored in the goal) in common with the target icon. The "nearest" criteria added the requirement that the new visual location not only contain a target-matching icon but that the new icon also is the target-matching icon nearest to the current visual location, which has not yet been attended.

An additional change that was made to the modeling involved the test of the module state in many productions. When a production involves the use of a perceptual/motor module, such as the vision module in the icon search productions, a check on the status of the module is performed in the left-hand side of the production, the logic being that the production should not fire if the module is still "busy" executing a previous command. However, "busy" can mean several things. In the first versions of the icon search modules, the status check of the vision module was "modality free," which matched when the module had entirely completed the previous command sent to it from cognition. In many of the productions in the new models the status requirement is changed to “processor free,” which is a less stringent test in the sense that it will match under more conditions. Specifically with respect to the vision module and EMMA, the “processor free” condition will be met once the encoding of a visual object is complete. This allows the model to take advantage of EMMA and make new shifts of visual attention before others are completely carried out. This change should manifest itself in quicker response times for the models.

Another change that was made to the modeling environment involves what we termed “feature synthesis”. In the previous models, each icon was “seen” by ACT-R/PM as a list of “characteristics” or “features.” The models would store one of these characteristics of the target icon in the goal, which was later used to identify icons matching the target icon. For example, each of the good quality icons was made up of only one characteristic, such as “red circle.” (As noted before, these “characteristics” or “features” do not directly correspond to “features” in the sense of low level visual features.) When the model focused visual attention on an icon containing a “red circle,” it would mark the “red circle” feature of the icon as having been attended. This worked fine with good quality icons, which were only made up of one feature. However, the medium and poor quality icons were composed of multiple features, and only one of these would be marked as attended when the icon was examined. Generally, only marking one feature of the icon as attended was functionally indistinguishable from marking all of the features as having been attended because the model was only “interested” in the one feature of the icon. However, there are a couple of problems with such a strategy. One issue arises in the rare instance when an icon has two or more of the same features (such as a poor quality icon with two “gray rectangles”). In this instance, it would be possible for the model to revisit an icon multiple times. Also, just marking one aspect of an icon as having been attended does not seem cognitively plausible given what we know about human visual perception. The change that was made to the second iteration of models was to have ACT-R/PM mark all of the features that compose an icon as having been attended when any one of them is attended.

Although, this change in the modeling environment alone did not have a substantial impact on this version of the models, we felt it made our models more generalizable to other situations, and it allowed us to introduce some changes to the icon search strategy of the models. Specifically, it allowed us to introduce a more complex strategy for the Double-Shift model. The model still shifts attention to an icon based on the presence of a single characteristic (such as “gray triangle”) in an icon. However, once the model has shifted attention to the icon, if the icon is not the same icon as the target icon, but only shares a feature in common with the target icon, it will not shift attention down to the filename. This “smarter” strategy is only effective for the medium quality icons; the good quality icons are only composed of a single feature, and we did not implement the feature synthesis of the poor quality icons to aid in their distinctiveness.

One of the aspects of the Text-Look model that was changed in the new models was the model’s behavior of revisiting icons that it had already “examined.” Certainly, this revisitation of the model accounted at least in part for the similar performance of the two models in terms of response time. While the DS model made more shifts of visual attention than the TL model, the cost in response time of the additional shifts was compensated in the TL model by its propensity to “revisit” icons. This behavior was due to the fact that the TL model only marked filenames as having been attended, but the icons above each filename were not marked as attended. This is because the ACT-R/PM architecture only marks as attended visual objects where visual attention has been shifted. Because visual attention went directly to the filename below the icon in the TL model, there was never any opportunity for the model to mark the icon as having been attended. Although, this aspect of the model’s performance certainly proved to be interesting, we could think of no cognitively plausible explanation for this random revisitation strategy. As a result, in the new models, we made some changes that allowed us to mark specific objects at a location as having been attended even when visual attention had not explicitly been directed there. This change was manifested in a substantial reduction in response time for the TL model.

One final change to the model involves the use of preparing motor movements in ACT-R/PM. One aspect of the experiment participants’ behaviour that was apparent was their use of the mouse. Often participants “trailed” their eyes with the mouse (See Figure 8.2). This strategy presumably resulted in quicker response times for subjects; when a participant finally located the target icon, the mouse movement needed to select the icon was much shorter than if the mouse had been left in its original position on the distractor screen (where the “ready” button was on the previous screen). Despite the clear advantage of this “trailing” strategy, our models initially used the less efficient strategy of making one long mouse movement once the target icon had been selected. We realized the inadequacy of this strategy early on, but to dynamically model eye movements and corresponding motor movements was beyond the scope of our project and possibly even beyond the current capabilities of ACT-R/PM.

In the first iteration of the DS model presented previously, we attempted to approximate the trailing strategy by preparing the motor movement of the mouse. As with movements of visual attention, motor movements have a preparation stage in ACT-R/PM. By preparing the mouse movement in the general direction of the distractor set (towards the center of the set), the models could become slightly more efficient. However, preparing the motor movement requires time, which would not be an issue unless another motor movement is requested before preparation is complete. This request during preparation never occurred in the DS model, but could rarely (and randomly) occur in the TL model. If the target icon happened to be the first icon examined by the TL model in the distractor set, then a mouse movement would be requested to select the target icon before the preparation had been completed. In this case, the system would have to wait for the preparation to complete before the mouse movement could be made. Hence, in this instance, the preparation of the mouse movement had an increased cost in response time, rather than the anticipated savings. Further, the random examination of the target icon on the first shift of visual attention was more likely to occur at the smaller set sizes (simply because there were less items to be randomly selected from). Because of this unequal effect across set sizes and because we were initially very interested in examining the slope of the reaction times across the set sizes, we did not prepare the motor movement with the TL model. The preparation was used with the DS model, however. In order to complete the analysis, we have once again implemented the motor movement preparation in a version of the revised TL model.

9.2 Modeling Results

The model data was gathered by running the models for 80 blocks of trials, which approximates the number of real blocks of subject data involved.

As in the previous comparisons of the modeling data with the experiment data, in each of the comparisons made here, several empirical metrics of comparison are used, including the root mean squared error (RMSE), the percent average absolute error (PAAE), and the proportion of variance explained (R^2).

In figures 9.1 and 9.2, both models are presented along with the experiment data. All of the “new” models incorporate the feature synthesis change and the “nearest” strategy. The only other changes made to the model at this point are the introduction of EMMA to the architecture and the use of the “processor free” criteria in the model productions. With respect to this version of the Text-Look model, the PAAE was 26.59%; the RMSE was 628 ms, and the R^2 was 0.97. On the basis of response time alone, relative to our previous models, the introduction EMMA into the TL model severely hurt its performance. The Double-Shift model faired better; the PAAE was 15.33%; the RMSE was 322 ms, and the R^2 0.99. However, this still does not match the performance of our previous models in this domain.

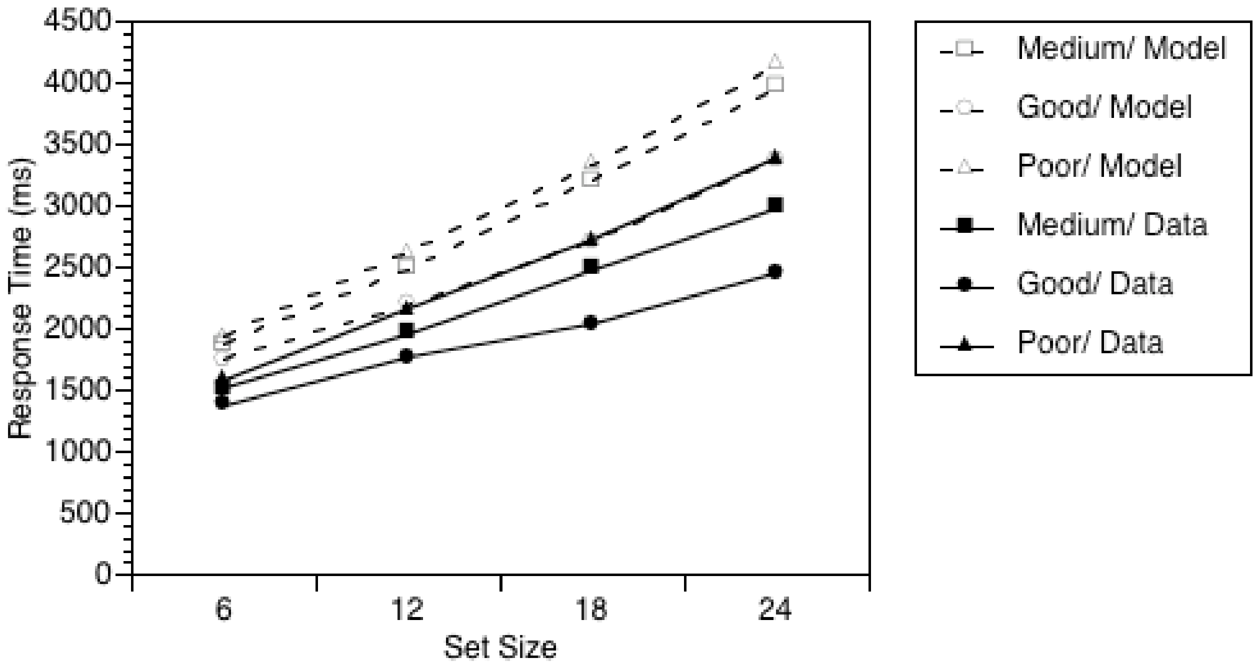

Figure 9.1. Text-Look model and experiment mean response times by set size and icon quality. In this version of the model, new incorporations to the model include EMMA (with the "processor free" criteria), feature synthesis, and the nearest strategy.

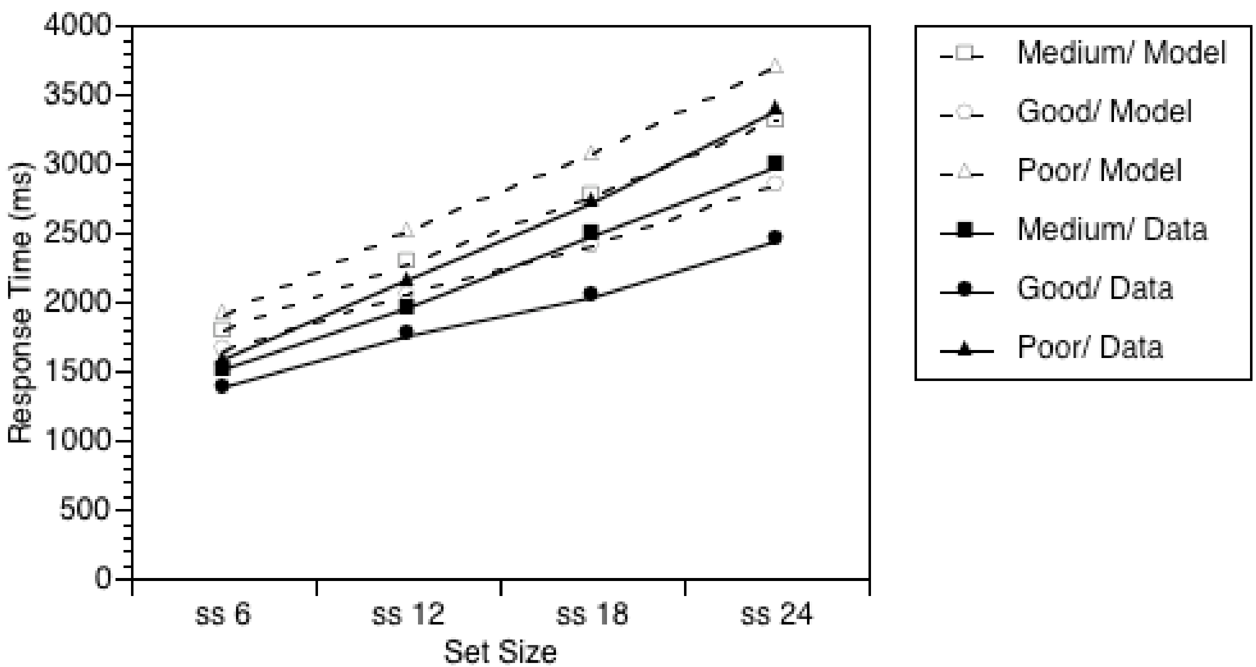

Figure 9.2. Double-Shift model (with incorporation of EMMA and "processor free") and experiment mean response times by set size and icon quality.

In figure 9.3, a version of the TL model is presented with the experiment data. The model data presented here include the incorporation of marking the icons as attended when the model examines the filename below the icon as well as all changes made in previous iterations of the model. The change to the TL model resulted in much improved performance; the RMSE was 216 ms; the PAAE was 10.58%, and the R^2 was 0.99.

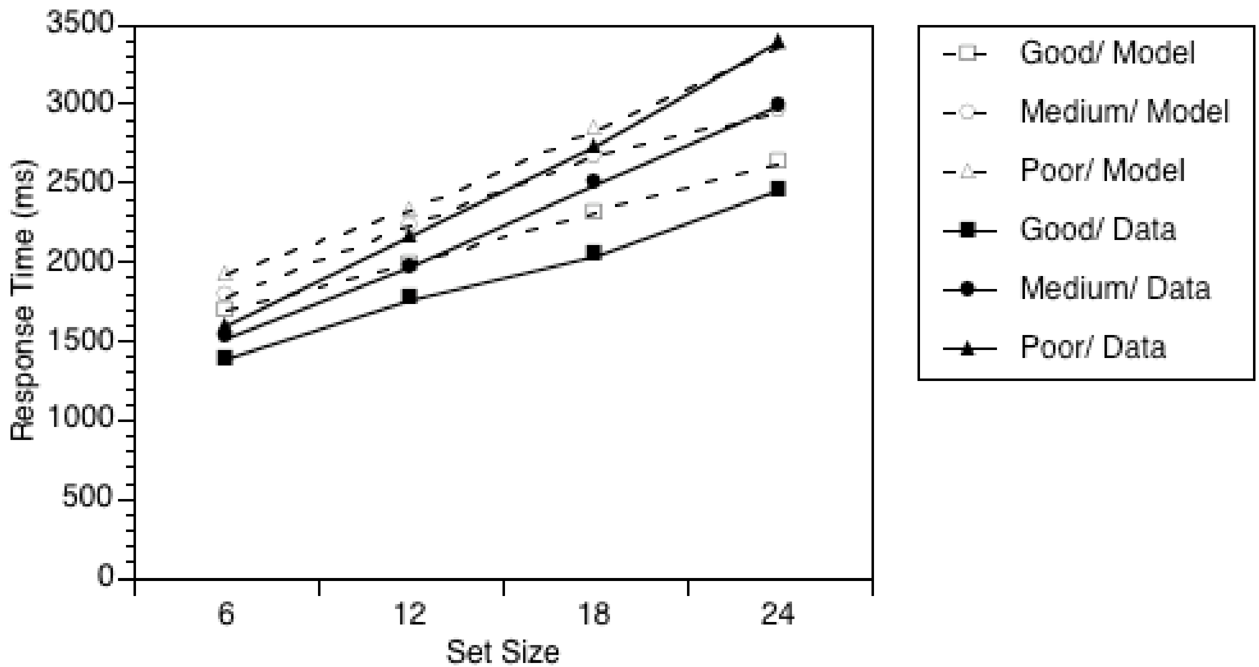

Figure 9.3 Response time by set size and icon quality for the Text-Look model and the experiment data. The model data is based on the Text-Look model with the incorporation of EMMA (with the “processor free” criteria), marking icons as attended, using the “nearest” search strategy, and feature synthesis.

Figure 9.4 shows the TL model with the incorporation the preparing the motor movement to the approximate center of the distractor screen. The model now approaches the comparison metric values of the earlier iteration of the TL model; the RMSE was 129 ms; the PAAE was 5.89%, and the R^2 was 0.99.

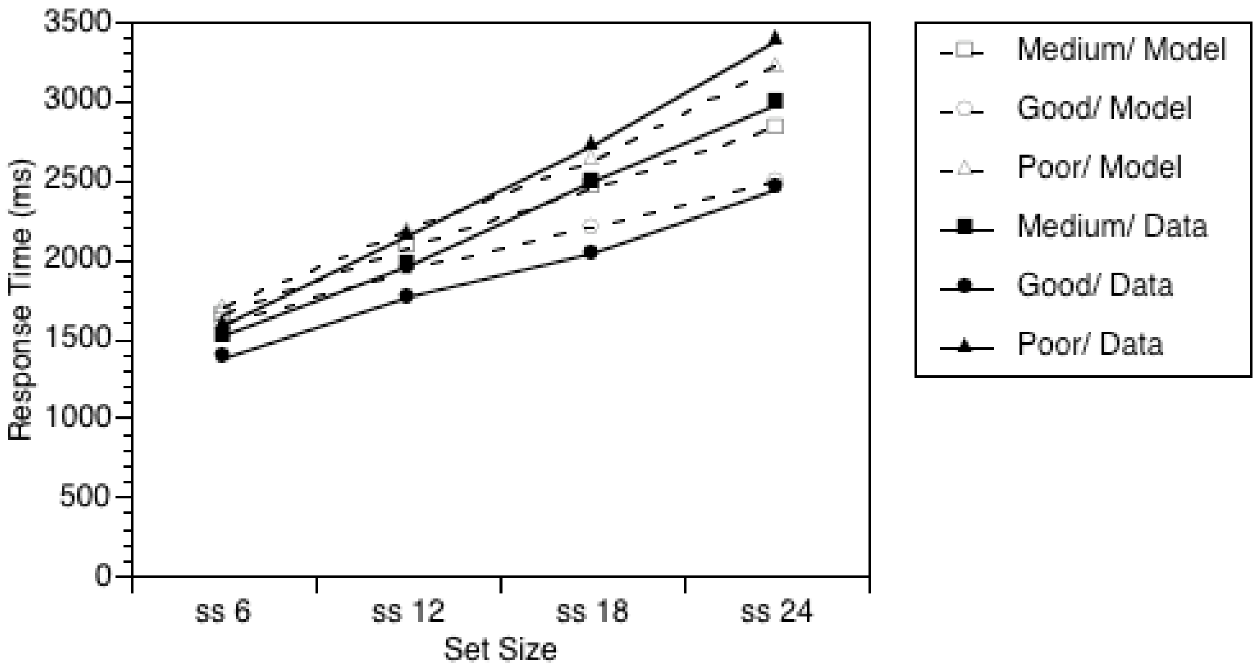

Figure 9.4 Response time by set size and icon quality for the Text-Look model and the experiment data. The model data is based on the Text-Look model with the incorporation of EMMA (with the "processor free" criteria), marking icons as attended, using the "nearest" search strategy, feature synthesis, and preparing the motor movement.

One other metric by which we can compare the models to human performance is that of the eye fixation data relative to the mean number of shifts of visual attention in the model. The fixation data cannot be compared with the point of regard data generated by EMMA. Each shift in POR made by the model represents a predicted eye movement but does not necessarily correspond to a fixation as they as were measured in our analysis.

As with eye movements relative to fixations, there are a much larger number of shifts of POR for each shift of visual attention in the model.

A graph comparing the eye fixation data to the mean number of shifts of visual attention in the model is presented for the both the previous and the new Text-Look models in Figure 9.5. (No changes were made to the DS model that would result in a change in the number of shifts of visual attention made by the model.) Relative to the previous TL model, the current model, with the incorporation of marking the icons as attended when the filename is examined fares much better although the model makes slightly more overt shifts than subjects. Using the same metrics for comparing the new model visual shift data to the experiment fixation data as for the response time data compared previously, the RMSE was 0.58 fixations; the PAAE was 15.79%, and the R^2 was 0.99.

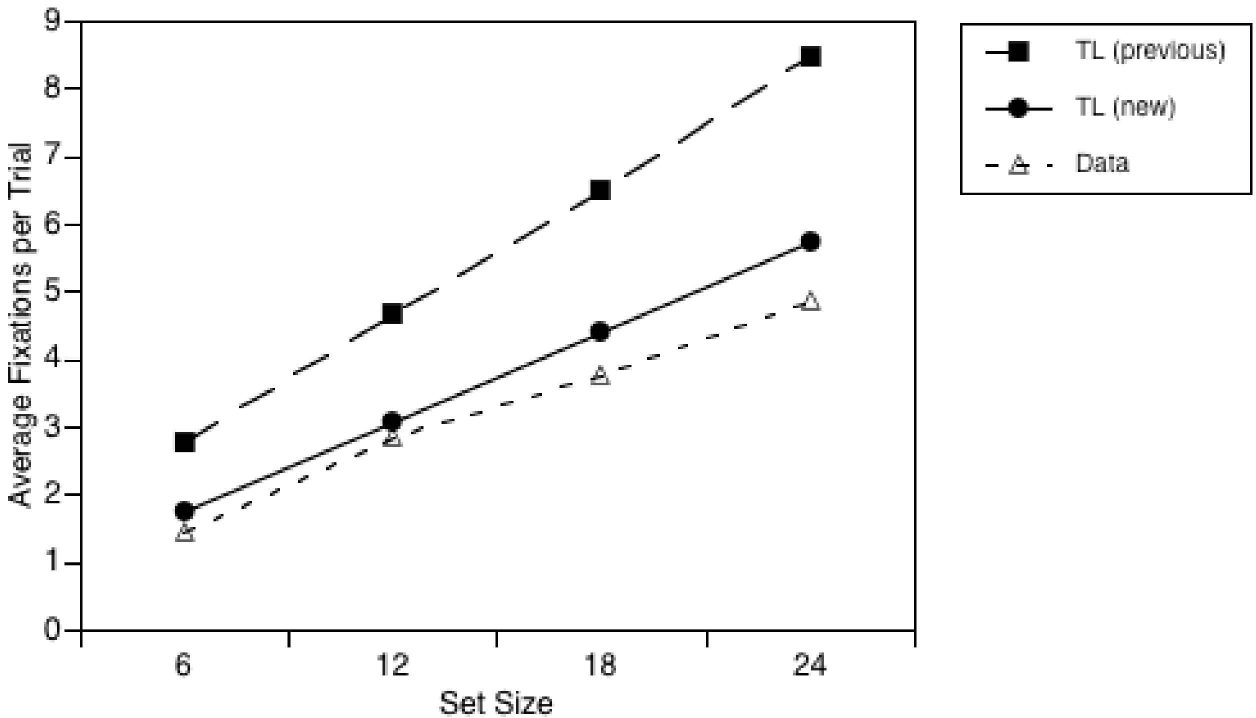

Figure 9.5 Mean number of fixations (data) or shifts of visual attention (models) by set size for both the previous and the new TL models as well as the eye tracking data. The new TL model used was the final model, which included the motor movement preparation.

One interesting trend to note in the eye tracking data is that the experiment participants averaged approximately one fixation for each target icon that would theoretically be examined in the distractor screen. For example, with a set size of six icons, there are two icons exactly matching the target icon in the set (one-third of the set). In the best case scenario, the first and only icon that the participant looks at is the target icon with the matching filename. If participants only made fixations to target-matching icons and could ignore the rest of the distractor set, then the worst case scenario would have participants only looking at both of the target-matching icons in the distractor set—i.e. the second icon they looked at would be the icon with the matching filename. On average, then, we would expect these efficient searches (where only icons matching the target icon are examined) to look at 1.5 icons in a set size of six icons. Interestingly, the average number of fixations made by participants in the eye tracking experiment conforms to this theoretical average for each of the different set sizes quite closely. This "best case" search is exactly the method of search employed by the model with the good quality icons—i.e. only icons matching the target are examined. The experiment data, along with the TL model averages, are presented in Table 9.1. The data are not quite so clean when examined by set size and icon quality. As expected, participants make more fixations when searching poor quality icons and less fixations when searching through good quality icons (Figure 9.6). The fact that the participants made consistently fewer fixations than the model (by approximately 0.5 fixations per trial) suggests that the models should be slightly more efficient in this respect. EMMA may provide this increase in efficiency, but as mentioned before, we cannot compare the number of shifts of visual attention made by EMMA with the number of fixations made by participants.

Set Size	Eye Tracking Data	New TL Model	TL model with good icons	Theoretical "best case" average
6	1.46	1.74	1.53	1.5
12	2.86	3.10	2.64	2.5
18	3.76	4.43	3.63	3.5
24	4.86	5.74	4.55	4.5

Table 9.1 Average number of fixations / shifts of visual attention for the experiment data, the overall average for the TL model, the TL model with good quality icons, and a theoretical efficient search.

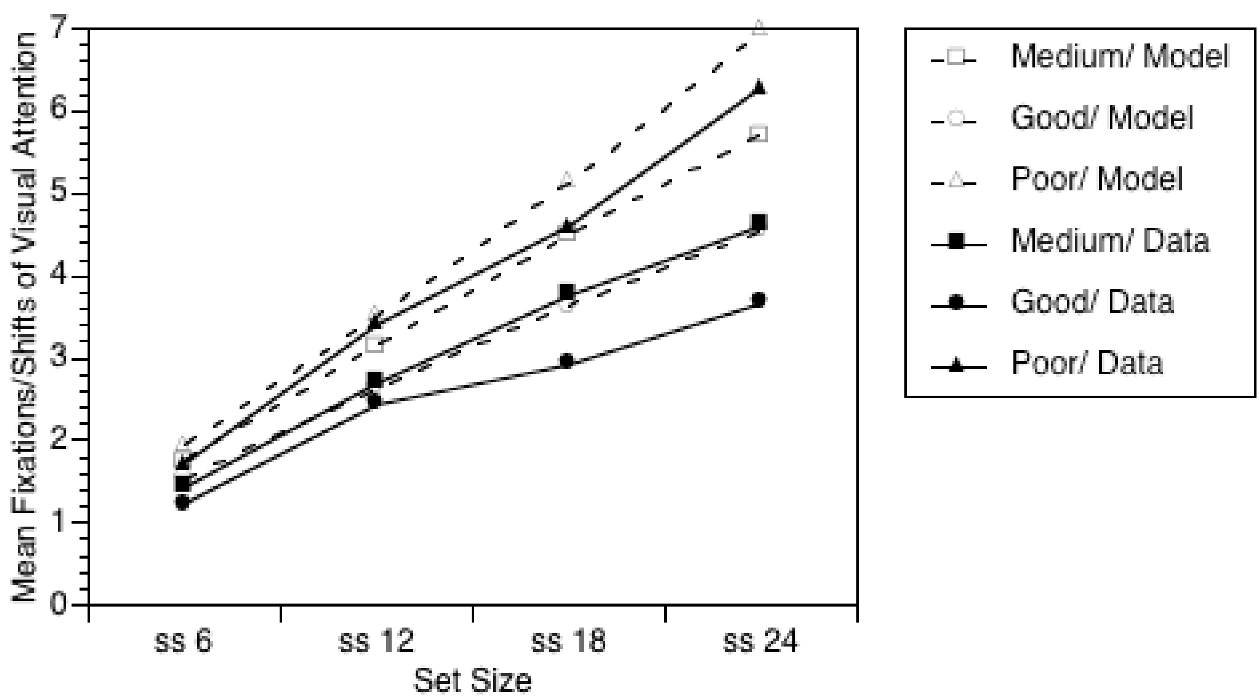

Figure 9.6 Mean fixations/shifts of visual attention by set size and icon quality for the final version of the TL model and the eye tracking data.

Although we cannot evaluate certain aspects of the models quantitatively, the qualitative performance of the model is quite improved in many respects. One aspect of the eye tracking study that we discussed was the general search patterns of participants. We noted that participants employed a “directed” strategy that was quite efficient in terms of only examining target-matching icons (at least with the good quality icons). There was also some evidence for a grouping strategy, whereby the icons in a group of target-matching icons were examined before moving on to another group of target matching icons. An example of a trial where these strategies were employed was given, which is reproduced in Figure 9.7. The new versions of the model were able to reproduce these qualitative aspects of the data quite well. As an example of the capability of the models, the exact same trial as was presented to the user in Figure 9.7 was run with the model. The blue line shows the resulting trace of the POR data of the newest TL model. The model begins its search from the “Ready” button and enters the depicted portion of the trial from the lower-left corner. The model proceeds in a fashion quite similar to that of the human participant, first examining the largest group of icons before moving on to the nearest group and finally to the target icon in the lower-right corner of the window. Certainly, the model would not follow this exact search pattern every time (the first icon that the model selects for search is randomly chosen from the target-matching icons presented), but the capability of the model to mimic human performance in this respect is encouraging.

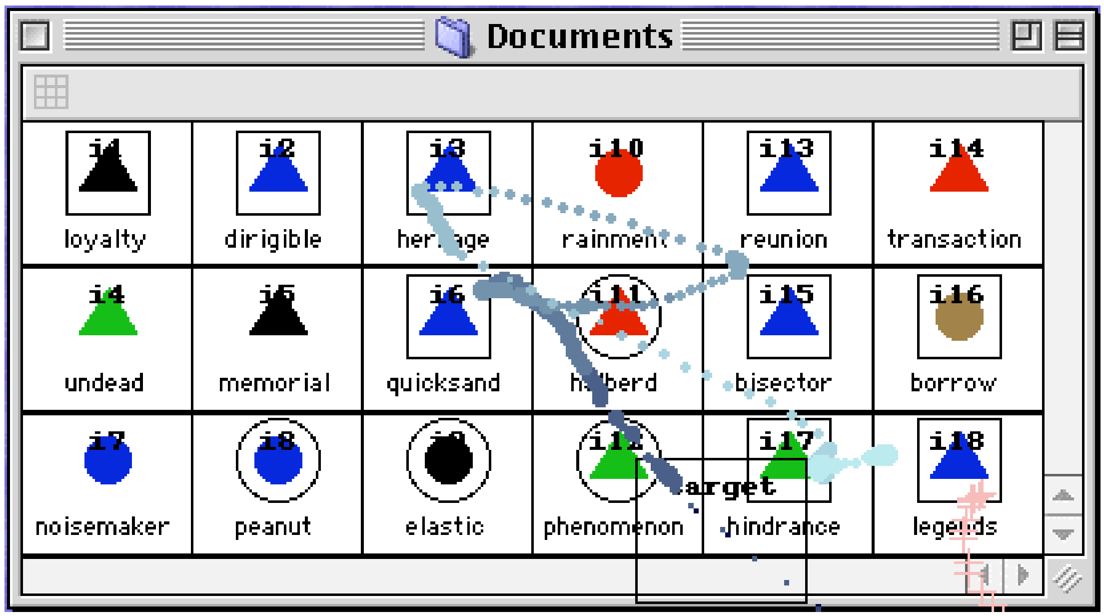

Figure 9.7. Reproduction of Figure 7.6. Eye tracking example of a directed search with good quality icons.

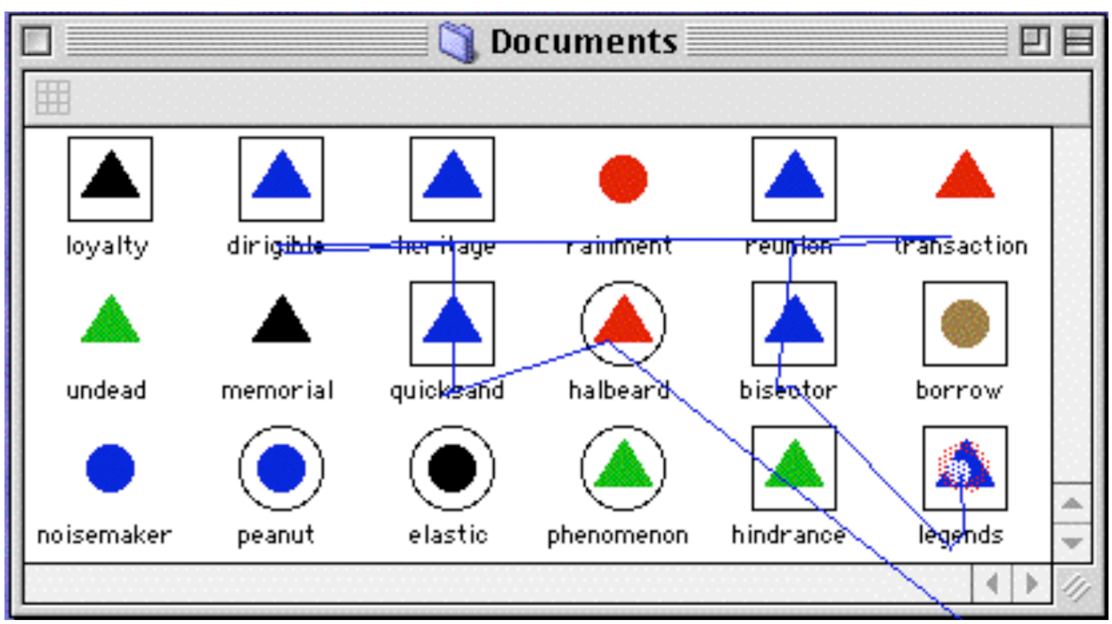

Figure 9.8. Example of the Text-Look model running an identical trial to that presented in Figure 9.7. The blue line indicates the POR path of the model. The model POR data begins at the “Ready” button, which out of view in the lower-left corner and finishes by selecting the icons above the filename “legends.”

9.3 Discussion of Modeling Revisions

There are several questions we can ask where the performance of the models is concerned. At a macro level, we need to address the issue as to whether the new versions of the models are improved over the previous versions. Before we get to that level of analysis, we'll take a look at each of the "improvements" individually.

Adding the preparation of the motor movement towards the center of the distractor set in the TL model was one change that quantitatively improved the match of the model data to the participant data. However, it altered one of the specific performance elements that we were interested in, the relative increase in response times with good quality icons across the four levels of set size. The slope of this line, which is effectively the time it takes the user to examine an additional icon, was one of the primary metrics by which we were examining the efficiency of the models. Preparing the motor movement reduced the overall response time across the four set sizes, but because this reduction was disproportionately greater at the larger set sizes, the efficiency of the model moved beyond that of participants in the experiment as examined in the aforementioned line slope. Our model slope was now below 300 ms, while the participant slope was over 350 ms.

The point here is not necessarily that our model slope did not exactly match that of the participants, but rather that we were unable to model one aspect of human performance in the icon search task. As mentioned before, the motor preparation was really just an very loose approximation of a more complex strategy employed by the participants. Often, the participants would trail their eye movements with the mouse, so that when they visually located the target icon, the mouse movement to select it was shorter. The altered slope in the model's performance when the motor preparation was added to the model was just a further indication that the motor preparation is a poor approximation of the true behavior of users. However, because the addition of the motor preparation did improve the quantitative fit of the model data to the experiment data, it suggests that there is at least some benefit to including the motor movement in the model.

Although there is room for improvement here, creating a model that accurately mimics the performance of users eye movements and mouse movements when engaged in a task such as icon search was well beyond the scope of this project. Modeling the dynamic interplay between the two aspects of behavior at a low level of perceptual and motor performance is a daunting task and would likely strain the current capabilities of ACT-R/PM.

One of the larger changes that was made to the Text-Look model was marking the icons above each filename as having been attended when the model examines the filename. This had a large effect in the RT data of the model because the model no longer revisited icons that it had previously examined. Additionally, marking the icons as attended had a large effect on the number of shifts of visual attention that the model made. The visual shift data for the new model closely approximated the fixation data from the eye tracking experiment; the model was off by a relatively constant value of approximately 0.5 fixations across the four set sizes.

Although we cannot directly compare the POR data from the EMMA modeling runs, we can examine the effect EMMA had on the response time of the model. The clear effect of incorporating EMMA into the models was an overall increase in response time. The previous models used a constant parameter of 135 ms for each shift of visual attention. EMMA uses a set of algorithms to compute the encoding and saccade time based primarily on the eccentricity of the target object from the current POR and the frequency of the target object. Because the average response time of the models increased with EMMA, one obvious conclusion is that the values computed by EMMA for making a shift of visual attention to each new target sum to a value greater than 135 ms. A closer examination of this attribute of EMMA revealed that longer saccades, such as those from one side of the distractor set to the other, took an estimated time much greater than 135 ms, and were thus responsible for much of the increase in average saccade time.

One way to work with the concept that longer saccades result in much longer response times is to reduce the number of lengthy shifts of visual attention made by the model. In our discussion of the original models, we noted that the strategy for finding the target icon worked by matching any icon on the distractor screen that contained the target characteristic that had been stored in the goal. The icon that was examined in each case was randomly selected from all the potential matching icons. At that time, we considered implementing a strategy that searched from left to right, top to bottom, etc. However, since the location of the target icon was random, whether we searched randomly or in some more orderly fashion made no difference in the overall response times of the model. With the incorporation of EMMA, there is a clear cost any strategy that causes the model to make longer shifts of visual attention. A left-to-right or top-to-bottom strategy would reduce this cost in shift time, but this would not necessarily minimize it. An optimal search strategy, which is what we eventually implemented, would examine the nearest icon to the current point of regard, thereby minimizing the time to make the shift.

This “nearest” strategy has further implications as well. For one, it adds some credence to the “grouping” strategy that we were only able to verify at a qualitative level. This grouping strategy involved initially looking at the largest group of icons matching the target icon in the distractor set, then proceeding to the subsequently smaller groups until the target icon was located. Such a strategy would generally result in a very short average saccade distance, which would have implications in terms of response time.

The nearest strategy also has implications well beyond the realm of icon search. Tullis (1983, p. 510) discusses the grouping of information in the realm of screen design issues and techniques. “The ways in which the elements are grouped plays an important role in both the ease with which the users can extract the information and the interpretations that they assign to it.” Other researchers have made similar distinctions. For example, “grouping similar items together in a display format improves their readability and can highlight relationships between different groups of data.” (Cakir, Hart, and Stewart, 1980, p. 114). From this perspective, the organization of information on the screen has value to the user by giving them some additional categorical information regarding what is presented on the screen as well as improving the general “readability” of the information. Not to diminish the value of this categorical information, but from the perspective of our modeling effort, there is also an added value of grouping the information on the screen that is reflected at a much lower level in the cognitive system—in the visual search strategies employed by users. Grouping information will reduce the number and average distance of saccades made by the user while searching for a desired piece of information. As we've noted, shorter saccades and less of them will result in finding the desired information more quickly.

Despite all of the "improvements" that were made to the new versions of the models over the previous versions, the fit of the model data to the experiment data did not improve much according to our quantitative metrics. Figures 9.9 and 9.10 show the previous version of the TL model and the new version relative to the experiment data. A comparison of the metrics of fit of the two models is provided in Table 9.2.

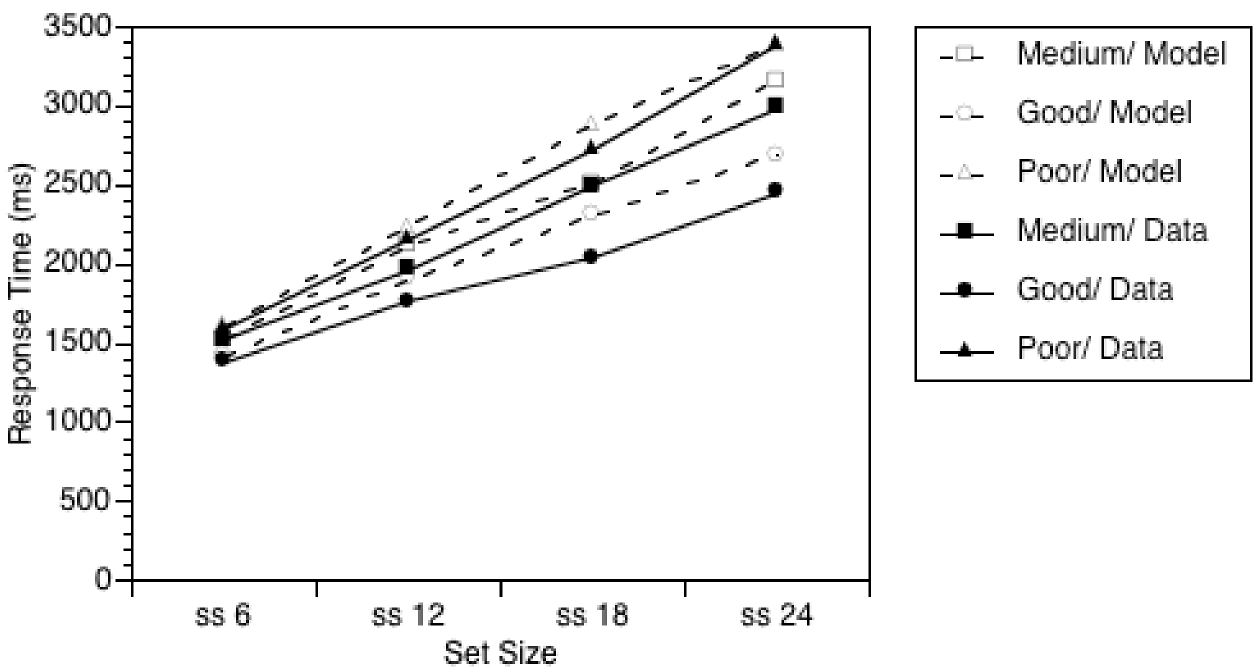

Figure 9.9 The original Text-Look model and presented with the data from Experiment 3.

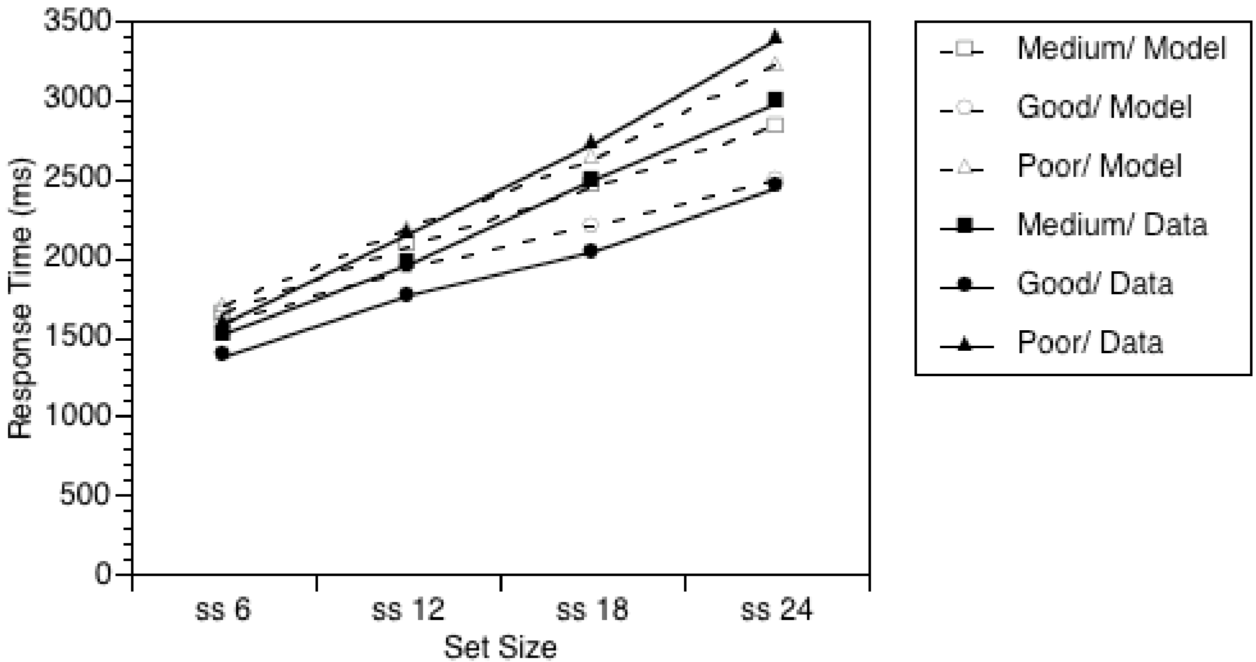

Figure 9.10. The final version of the Text-Look model presented with the data from Experiment 3.

	Original Text-Look Model	Final Text-Look Model
RMSE	137.61	129.45
PAAE	4.75%	5.89%
R^2	0.99	0.99

Tab le 9.2 Quantitative comparison of metrics of model to data fit for the original and the final version of the Text-Look model.

Examining the graphical presentations and the quantitative metrics of fit of the model and experiment data, it is clear that if there is any improvement in the performance of the models based on these presentations, it is slight. However, the eye tracking study has given us another perspective from which to view the fit of the model and the data. In previous discussions, we noted that the new version of the TL model is a much better fit of the fixation data than the old version (See Figure 9.5). Hence, part of the reason that we do not see a strong improvement in the fit of the reaction time data is that we are now concerned with matching multiple sets of data with the model.

There have also been a number of “improvements” to the model that have done little to improve the RT fit of the model but have had an effect on the performance of the model as a whole. For example, the feature synthesis incorporation had virtually no effect on the response times of the model but was implemented in order to improve upon the plausibility of the model. Also, the visual search patterns of the models using the “nearest” strategy versus a random search strategy had some effect on the RTs of the model but also went towards matching the qualitative patterns we had seen in the eye tracking data, such as a the tendency for participants to search within groups of target-matching icons. Again, judging the performance of the model from multiple perspectives, rather than just that of the response time data, we see that the new versions of the models are indeed improved.

Perhaps the most significant reason that little improvement was seen in the metrics of the model to data fit is due to the construction of the feature lists that comprise how ACT-R/PM “sees” each icon. In the first iteration of the modeling process, great care was taken in constructing a list of features that resulted in the best match of the model to data fit. Much of this was done in a post hoc manner. For example, a set of features was constructed and the model the run for a number of trials. Based on the performance of the model in that series of trials, the set of features was altered to create “improved” model performance. The alterations consisted of changing the amount of feature overlap within the different levels of icon quality. In essence, the feature list was treated as a free parameter in ACT-R/PM that could be adjusted so the models’ performance would most closely match the data. It is important to note that because there was never any feature overlap in the good quality icons, only the performance of the model with poor quality and medium quality icons was altered. However, the metrics of fit that we have examined take into account all three levels of icon quality. (The icon quality effect was one aspect of the data that we were interested in modeling). Because the same effort of altering the feature sets was not undertaken with the new models, we should not expect the overall metrics of fit to meet the same level as before.

The fact that construction of the feature sets plays such an important role in the predictive power of the model indicates a clear weakness in modeling a visual process in ACT-R/PM. In fact, this issue goes beyond ACT-R/PM; to our knowledge, no one has developed a method for feature decomposition of displays such as those used here. Much of the predictive power of modeling in general is lost when some of it has to be done in a post hoc manner. The development of a computational model that is capable of making specific predictions regarding a mental representation of a complex visual scene is needed for the modeling of tasks that rely on a visual representation of the world to continue to improve. Fortunately, there are a number of researchers working to gain some insight into this issue. EMMA represents one step in this direction. Jeremy Wolfe and his colleagues have made some headway in this arena with their “Guided Search” models (1994). Kieras and Meyer (1997) have done some work implementing a system of vision that takes into account a number of “strong” visual features, such as color and shape, within a larger cognitive modeling architecture. Despite the progress made, a comprehensive account of a relatively complex visual scene remains a long ways off.

However, without this work, much of modeling the visual world will functionally remain a "free parameter" in any modeling effort.

10. GENERAL DISCUSSION AND CONCLUSIONS

Overall I was pleased with the performance of the models. Each of the major trends in the data were well captured, the effect of set size, the effect of icon quality, no effect of icon borders, and the general icon search strategies employed by users. Further, our understanding of icon search was expanded, particularly in each of these areas.

In each of the experiments and in each of the models, there was a linear increase in icon search time with an increase in set size. This increase was in the range of 350 milliseconds to 500 milliseconds for each additional icon added to the display depending on the level of quality of the icons. Clearly, the number of icons in the display is a powerful predictor of icon search performance. The accuracy of the models in matching this trend in the data speaks to the accuracy of the strategies in the models and of the timing parameters in ACT-R/PM. The timing of the mouse movements based on Fitts' law and the eye movements based on the algorithms in EMMA all proved to be very accurate in modeling a relatively complex task.

The level of icon quality also proved to be a powerful predictor of performance. We defined icon quality generally as the more distinctive an icon is from a specified set of icons, the higher the quality in that context. We based our construction of high quality icons on principles derived from the visual search literature, specifically on the "pop out" features of objects. The generation of the icon quality effect within the model work provided us with several insights into the realm of icon quality. We created the effect with what we termed “feature overlap,” where icons shared multiple characteristics. In essence, our feature overlap was a measure of similarity between icons, a factor that is likely important in the effectiveness of icon search by computer users. One issue with the characteristics that we used to identify our icons in ACT-R/PM is that they were created by the experimenters and tweaked in a post hoc fashion. This highlights the need for a modeling architecture that is capable of generating this list of identifying features on its own, or at least the development of a systematic method for doing so.

Capturing the lack of an effect of icon borders on the performance of the model is obviously a very trivial exercise, simply don’t include any reference or use of the borders in the functioning of the model. The fact that the performance of the model fits the data quite well without the inclusion of icon borders in the process adds some support, albeit tenuous, that borders play no part in the search strategies of computer users. However, this is only half of the issue that we are concerned with. We would also like to know if borders can be used by computer users to augment and improve performance in icon search tasks. Further research to examine whether borders may be able to play a useful role in icon search is under way.

The strategies that were implemented in the models also provided some insight into the strategies of human computer users. For one, the strategy of the model was a surprisingly robust parameter. A number of changes were made to the multiple strategies in the modeling effort without drastically different results. This suggests that icon search is a cognitive process that is driven from the “bottom-up” more than “top-down”—i.e. variation in the characteristics of the display, such as the quality and number of the icons, have a greater impact on the search times of users than does variation in the strategies of users.

Our original goal in this set of studies was to gain some understanding of the icon search process. This set of studies sheds some illumination on the factors that contribute to the icon search process. Further, we have developed a model that makes quantitative predictions regarding the performance of users in displays employing icons. There is no reason that this line of research should end here; there are numerous opportunities for further development of the model and greater understanding of the icon search process.

The ultimate goal of this line of research is to develop an engineering level model of the task that can aid in the design of real world systems. The models we have developed here make a lot of headway in that direction and have served to point out as to where future cognitive models and modeling architectures (ACT-R/PM in this case) need to be improved. More importantly, the models developed here serve to make some initial predictions regarding the performance of users in icon based systems, which moves toward our overarching objective of aiding the designers of real world systems.

REFERENCES

Anderson, J. R., and Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Lawrence Erlbaum Associates.

Anderson, J. R., Matessa, M., & Lebiere, C. (1997). ACT-R: A theory of higher level cognition and its relation of visual attention. *Human-Computer Interaction*, 12, 439-462.

Arend, V., Muthig, K-P. and Wandmacher, J. (1987). Evidence for global feature superiority in menu selection by icons, *Behaviour & Information Technology*, 6, 411-426.

Becker, W. and Jurgens, R. (1979). An analysis of the saccadic system by means of double step Stimuli. *Vision Research,* 19, 967-983.

Bilsky, A.A., Wolfe, J.M., and Friedman-Hill, S.F. (1994). Part-whole information is useful in size x size but not in orientation x orientation conjunction searches. *Investigative Opthalmology and Visual Science*, 35(4), 1622.

Bloomfield, J. R. (1970). Visual search, Ph.D. Thesis, University of Nottingham.

Byrne, M. D. (1993). Using icons to find documents: simplicity is critical, *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, 446-453.

Byrne, M. D. (2001). ACT-R/PM and menu selection: applying a cognitive architecture to HCI. *International Journal of Human-Computer Studies*.

Byrne, M.D. & Anderson, J.A. (1998). Perception and action. In J.R. Anderson & C. Lebiere (Eds.), *The Atomic Components of Thought*, 167-200. Hillsdale, NJ: Lawrence Erlbaum Associates.

Cakir, A., Hart, D. J., & Stewart, T. F. M. (1980). *Visual display terminals: A manual covering ergonomics, workplace design, health and safety, task organization.* England: Wiley.

Coeffe, C., and Regan, J.K. (1987). Reducing the influence of non-target stimuli on saccade accuracy: Predictability and latency effects. *Vision Research*, 27(2), 227-250.

Dick, M., Ullman, S., and Sagi, D. (1987). Parallel and serial processes in motion detection. *Science*, 237, 400-402.

Enns, J. T., and Rensink, R.A. (1990). Sensitivity to three-dimensional orientation in visual search. *Psychological Science*, 1(5), 323-326.

Fahle, M. (1991). Parallel perception of vernier offsets, curvature, and chevrons in humans. *Vision Research*, 31(12), 2149-2184.

Findlay, J.M. (1982). Global visual processing for saccadic eye movements. *Vision Research*, 22(8), 1033-1045.

Findlay, J.M. (1997) Saccade target selection in visual search. *Vision Research*, 37, 617-631.

Fleetwood, M. D. (2000). The effect of simple icon borders on icon search. Unpublished manuscript.

Fuchs, A.F. (1971). The saccadic system. In P. Bach-y-Rita, C.C. Collins, & J.E. Hyde (Eds.), *The Control of Eye Movements*, 343-362. New York: Academic Press.

Gould, J. and Dill, A. (1969). Eye-movement parameters and pattern discrimination. *Perception and Psychophysics*, 6, 311-320.

Henderson, J. M. (1992). Visual attention and eye movement control during reading and picture viewing. In K. Rayner (Ed.), *Eye Movements and Visual cognition: Scene Perception and Reading*. New Your: Springer-Verlag.

Houde, S. and Salomon, G. (1993). Working towards rich and flexible file representations. *Proceedings of the CHI '94 conference companion on Human factors in computing systems*, 9-10.

Jacko, J. Dixon, M. Rosa R. H. Jr., Scott, I. U. and Pappas, C. J. (1999). Visual profiles: A critical component of universal access, *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, 1, 330-337.

Kacmar, C. and Carey, J. (1991). Assessing the usability of icons in user interfaces, *Behaviour and Information Technology*, 6, 443-457.

Kieras, D.E., & Meyer, D.E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic Mechanisms. *Psychological Review*, 104, 3-65.

Mohr, W. (1984). Visuelle Wahrnehmung und Zeichenfunktion, (Roderer, Regensburg); cited in Scott 1993.

Motter, B.C., and Belky, E.J. (1998) The zone of focal attention during active visual search. *Vision Research*, 38(7), 1007-1022.

Motter, B.C., and Belky, E.J. (1998b). The guidance of eye movements during active visual search. *Vision Research*, 38(12), 1805-1815.

Mounts, J.R. and Melara, R.D. (1999). Attentional selection of objects or features: Evidence from a modified search task. *Perception & Psychophysics*, 61(2), 322-341.

Muter, P. and Mayson, C. (1986). The role of graphics in item selection from menus, *Behaviour & Information Technology*, 5, 89-95.

Nagy, A.L. and Sanchez R.R., (1990). Critical color differences determined with a visual search task. *Journal of the Optical Society of America A*, 7(7), 1209-1217.

Rayner, K. (1995). Eye movements and cognitive processes in reading, visual search, and scene perception. In J. M. Findlay, R. Walker, & R. W. Kentridge (Eds.) *Eye Movement Research: Mechanisms, Processes, and Applications*. New York: Elsevier Science Publishing.

Rayner, K., Reichle, E. D., & Pollatsek, A., (1998). Eye movement control in reading: An overview and model. In G. Underwood (Ed.), *Eye Guidance in Reading and Scene Perception* (pp. 243-268). Oxford, England: Elsevier.

Reichle, E. D., Pollatsek, A., Fisher, D. L., & Rayner, K. (1998). Toward a model of eye movement control in reading. *Psychological Review, 105*, 125-157.

Salvucci, D. D. (2000). *An integrated model of eye movements and visual encoding*. Submitted manuscript.

Salvucci, D. D. and Anderson, J. R., (2000). Intelligent gaze-added interfaces. *Proceedings of ACM CHI 00 Conference on Human Factors in Computing Systems*, 1, 273-280.

Scott, D. (1993). Visual search in modern human-computer interfaces, *Behaviour & Information Technology*, 12, 174-189.

Sibert, L. E. and Jacob, R.J.K., (2000). Evaluation of eye gaze interaction. *Proceedings of ACM CHI '00 Conference on Human Factors in Computing Systems*, 1, 281-288.

Smallman, H.S., and Boynton, R.M. (1990). Segregation of basic color in an information display. *Journal of the Optical Society of America A*, 7(10), 1985-1994.

Triesman, A. M., and Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, 12, 107-141.

Treisman, A.M. and Gormican, S. (1990). Feature analysis in early vision: Evidence from search asymmetries. *Psychological Review*, 95, 15-48.

Tullis, T. S. (1983). The formatting of alphanumeric displays: a review and analysis, *Human Factors*, 25, 657-682.

Tullis, T. S. (1997). Screen Design. In M. Helander, T. K. Landauer, & P. Prabhu (Eds.) *Handbook of Human-Computer Interaction*, 2nd Edition. New York: Elsevier Science Publishing.

Viviani, P. and Swensson, R. G. (1982). Saccadic eye movements to peripherally discriminated targets. *Journal of Experimental Psychology, Human Perception and Performance*, 8, 133-126.

Wandmacher, J. and Muller, U. (1987). On the usability of verbal and iconiccommand representations, *Zeitschrift fur Psychologie*; cited in Arend et al. 1987.

Williams, L.G. (1966). The effects of target specification on objects fixated during visual search. *Perception and Psychopysics*, 1, 315-318. Amsterdam: North-Holland.

Williams, L.G. (1967). The effect of target specification on objects fixated during visual search. *Attention and Performance 1*, 355-360.

Wolfe, J. M., Cave, K.R., and Franzel, S. L. (1989). Guided search: An alternative to the feature integration model for visual search. *Journal of Experimental Psychology: Human Perception and Performance*, 15, 419-433.

Wolfe, J.M. (1994). Guided search 2.0 A revised model of visual search.*Psychonomic Bulletin & Review*, 1(2), 202-238.

Wolfe, J.M. (1998). Visual search. In H. Pashler, *Attention* (pp. 13-74). East Sussex, UK: Psychology Press.

Wolfe, J.M., Klempen, N.L., and Shulman, E.P. (1999). Which end is up? Two representations of orientation in visual search. *Vision Research*, 39, 2075-2086.

Zelinsky, G.J., and Sheinberg, D.L. (1997) Eye movements during parallel-serial visual search. *Journal of Experimental Psychology: Human Perception and Performance*, 23(1), 244-262.

<u>Appendix #1 ACT-R/PM Productions</u>

Section 1: English language descriptions of productions

1.1 <u>Final Double-Shift model</u>

FOCUS-TARGET:
Finds the target icon on the screen and attends to it.

IF the goal is icon-search and the goal-state is “start”
And there is a visual location on the screen with an object of the kind file-icon,
And the vision module is free
THEN move visual attention to the location
And change the goal-state to “targ-state”.
And store the a feature and color of the target icon in the goal

EXAMINE-FEATURE
Randomly selects a value (feature) of the target icon and the feature’s corresponding color and stores the value and color in the goal to be matched against later. (Finesses remembering the icon.)

IF the goal is icon-search and the goal-state is “targ-state”
And there is an file-icon with a particular feature and color
THEN store that feature and color in the goal.

FOCUS-TEXT
Find the filename below the attended icon and attend to it.

IF the goal is icon-search and the goal-state is “targ-state”
and there is a visual location of the kind text, which has not been attended
and the vision module is free
THEN move visual-attention to the location.

EXAMINE-TEXT
Put text of filename into the goal slot "filename" to be tested against later. (Finesses remembering the file name.)

IF the goal is icon-search and the goal-state is “targ-state”
And there is an visual-object of the type text
THEN store that text in the goal as the filename
And change the goal-state to “ready”.

READY-BUTTON
Next, find the ready button that's on the screen and move the mouse over to it.

IF the goal is icon-search and the goal-state is “ready”
And there is a visual location of the value oval which has not been attended
And the vision module is free
And the motor module is free
THEN move visual attention to the location
And move the cursor to the location.

CLICK-READY
Once the mouse move is complete then we can click the mouse.

IF the goal is icon-search and the goal-state is “ready”
And there is a visual-object of the value oval at a specific location
And there is a visual-location, with a kind cursor
And the motor module is free
THEN click the mouse button
And change the goal-state to “distractor1”.

FIND-ICON-1
Now in distractor screen, move attention to a specific icon that has a feature matching that of the target icon stored in the goal, and also prepare the hand to move the mouse in the towards the center of the distractor set. (This production will only be called once.)

IF the goal is icon-search and the goal-state is “distractor1”
And there is a visual-location that has not been attended and whose feature and color matches that stored in the goal.
And the vision module is free
And the motor module is free
THEN move visual attention to the location.
And prepare the motor movement of the right hand
And change the goal-state to “distractor2”.

FIND-ICON-2
Now in distractor screen, move attention to a specific icon that has a feature matching that of the target icon stored in the goal. (This production is just like Focus-icon-1 except the hand movement is not prepared, and the “nearest” criteria is used for the new visual location.)
IF the goal is icon-search and the goal-state is “distractor1”
And there is a visual-location that has not been attended and whose feature and color matches that stored in the goal. (Select the nearest of those meeting the criteria.)
And the vision module is free
THEN move visual attention to the location.
And change the goal-state to “down”.

LOOK-TEXT
Find the filename below the attended icon and attend to it.

IF the goal is icon-search and the goal-state is "down"
And there is a visual-object at a specific screen position
And there is a visual location of the kind text, which has not been attended, and which is nearest to the visual-object
and the vision module is free
THEN move visual-attention to the visual location
And change the goal-state to "name".

EXAMINE-TEXT-N
Check to see if filename matches target filename. If doesn't, change state back to find an icon (Focus-Icon).

IF the goal is icon-search and the goal-state is "name"
And there is a visual-object whose value does *not* match the filename stored in the goal
THEN change the goal-state to "distractor".

EXAMINE-TEXT-Y
Check to see if filename matches target filename. If it does, change state to attend to icon.

IF the goal is icon-search and the goal-state is "name"
And there is a visual-object whose value does match the filename stored in the goal
THEN change the goal-state to "move-up".

BACK-UP
Moves attention and cursor back to the icon; so it can be clicked on.

IF the goal is icon-search and the goal-state is "move-up"
And there is a visual-object at a specific location
And there is a visual-location that has been attended with a feature and color matching those stored in the goal and nearest to the visual-object
And the vision module if free
THEN move visual attention to the visual-location
And make the motor movement to move the cursor to the visual-location.
And change the goal-state to "click-target".

CLICK-TARGET
Once the mouse move is complete then we can click the mouse.

IF the goal is icon-search and the goal-state is "click-target"
And there is a visual object with a specific location

And there is a visual-location, with a kind cursor
And the motor module is free
THEN click the mouse button
And change the goal-state to “done”.

FINISHED
When everything is done, pop the top goal.

IF the goal is icon-search and the goal-state is “done”
And the motor module is free
THEN pop the goal.

<u>Text-look model</u>

The productions for the text-look model are very similar. In fact, they are identical up to the Find-Icon productions and from the Examine-Text-N production on. However, the Find-Icon (1 and 2) and the Look-Text productions have been replaced with the following three productions.

FIND-ICON-1
Now in distractor screen, move attention to a specific icon that has a feature matching that of the target icon stored in the goal, and also prepare the hand to move the mouse in the towards the center of the distractor set. (This production will only be called once.)

IF the goal is icon-search and the goal-state is “distractor1”
And there is a visual-location that has not been attended and whose feature and color matches that stored in the goal.
And the vision module is free
And the motor module is free
THEN note the visual location in the goal.
And prepare the motor movement of the right hand
And change the goal-state to “distractor2”.

FIND-ICON-2
Now in distractor screen, move attention to a specific icon that has a feature matching that of the target icon stored in the goal. (This production is just like Focus-icon-1 except the hand movement is not prepared, and the “nearest” criteria is used for the new visual location.)
IF the goal is icon-search and the goal-state is “distractor1”
And there is a visual-location that has not been attended and whose feature and color matches that stored in the goal. (Select the nearest of those meeting the criteria.)
And the vision module is free
THEN note the visual attention in the goal.
And change the goal-state to “down”.

TEXT-FIRST
Moves visual attention to the filename corresponding to the icon located in Check-Icon.

IF the goal is icon-search and the goal-state is "distractor2"
And there is a visual location of the kind text, which has not been attended, and which is nearest to the location stored in the goal
and the vision module is free
THEN move visual-attention to the visual location
And change the goal-state to "name".

Section 2: ACT-R/PM Code of Productions

Double-Shift Model

```
(p focus-target
   =goal>
     isa icon-search
     state start
   =loc>
     isa visual-location
     time now
     attended nil
     screen-x lowest
     kind     file-icon
   =state>
     isa module-state
     module :vision
     modality free
   =state2>
       isa module-state
       module :motor
       modality free
==>
   !send-command! :vision move-attention :location =loc
   =goal>
     state targ-state)

(p examine-feature
   =goal>
     isa icon-search
     state targ-state
     feature1 nil
     color nil
   =obj>
     isa file-icon
     time now
     value =feature
     color =color
```

```
   =state>
     isa module-state
     module :vision
     modality free
==>
   =goal>
     feature1 =feature
     color  =color
)

(p focus-text
   =goal>
     isa icon-search
     state targ-state
     feature1 =thefeature
     color =thecolor
     filename nil
   =loc>
     isa visual-location
     time now
     attended nil
     screen-x highest
       kind text
   =state>
     isa module-state
     module :vision
     modality free
==>
   !send-command! :vision move-attention :location =loc :scale word)

(p examine-text
   =goal>
     isa icon-search
     state targ-state
   =obj>
     isa visual-object
     time now
       value  =text
   =state>
     isa module-state
     module :vision
     modality free
==>
   =goal>
       filename =text
       state ready)
```

```
(p ready-button
   =goal>
     isa icon-search
     state ready
   =loc>
     isa visual-location
     time now
     screen-x highest
     attended nil
     kind text
   =state>
     isa module-state
     module :vision
     modality free
   =state2>
     isa module-state
     module :motor
     modality free
==>
   !send-command! :vision move-attention :location =loc
   !send-command! :motor move-cursor :loc =loc)

(p click-ready
   =goal>
     isa icon-search
     state ready
   =btn>
      isa VISUAL-OBJECT
      value "ready"
      time now
      screen-pos =loc
   =loc>
      isa VISUAL-LOCATION
      kind CURSOR
      time now
   =state>
     isa module-state
     module :motor
     modality free
==>
   !send-command! :motor click-mouse
   !send-command! :vision clear
   =goal>
     state distractor1
)

(p focus-this-1
   =goal>
     isa icon-search
     state distractor1
     feature1 =thefeature
```

```
      color =thecolor
    =loc>
      isa visual-location
      time now
      attended nil
      value =thefeature
      color =thecolor
    =state>
      isa module-state
      module :vision
      modality free
    =state2>
       isa module-state
       module :motor
       modality free
==>
   !send-command! :vision move-attention :location =loc
   !send-command! :motor prepare ply :hand right :r 26.0 :theta -2.55
   =goal>
     state down
)

(p spin
   =goal>
   isa icon-search
   state distractor
==>
)
(spp spin :r 0.8)


(p focus-this-2
   =goal>
     isa icon-search
     state distractor
     feature1 =thefeature
     color =thecolor
   =loc>
     isa visual-location
     time now
     attended nil
     value =thefeature
     color =thecolor
     nearest  current
   =state>
     isa module-state
     module :vision
     processor free
==>
   !send-command! :vision move-attention :location =loc
   =goal>
     state down
)
```

```
(p look-text
   =goal>
     isa icon-search
     state down
     feature1 =thefeature
     color =thecolor
   =obj>
     isa visual-object
     screen-pos =loc
     time now
  =loc2>
     isa visual-location
     time now
     attended nil
     kind text
     nearest =loc
     screen-y greater-than-current
   =state>
     isa module-state
     module :vision
     processor free
==>
!send-command! :vision move-attention :location =loc2
=goal>
     state name)

(p examine-text-y
   =goal>
     isa icon-search
     state name
       filename =thefile
   =obj>
     isa visual-object
     time now
    value  =thefile
   =state>
     isa module-state
     module :vision
     modality free
   =state2>
       isa module-state
       module :motor
       modality free
==>
   =goal>
       state move-up)
```

```
(p examine-text-n
   =goal>
     isa icon-search
     state name
       filename =thefile
   =obj>
     isa visual-object
     time now
   - value  =thefile
   =state>
     isa module-state
     module :vision
     modality free
   =state2>
       isa module-state
       module :motor
       modality free
==>
   =goal>
       state distractor)

(p back-up
   =goal>
     isa icon-search
     state move-up
     feature1 =theicon
     color =thecolor
   =txt>
     isa visual-object
     screen-pos =loc2
     time now
  =loc>
     isa visual-location
     attended t
     value =theicon
     color =thecolor
     nearest =loc2
     screen-y less-than-current
     time now
   =state>
     isa module-state
     module :vision
     modality free
==>
!send-command! :vision move-attention :location =loc
   !send-command! :motor move-cursor :loc =loc
   =goal>
     state click-target)
```

```
(p click-target
   =goal>
     isa icon-search
     state click-target
   =obj>
      isa VISUAL-OBJECT
      time now
      screen-pos =loc
   =loc>
      isa VISUAL-LOCATION
      kind CURSOR
      time now
   =state>
     isa module-state
     module :motor
     modality free
==>
   !send-command! :motor click-mouse
   !send-command! :vision clear
   =goal>
     state done)

(p finished
   =goal>
     isa icon-search
     state done
   =state>
       isa module-state
       module :motor
       modality free
==>
   !pop!)
```

Productions Unique to the Text-Look Model

```
(p find-icon-1
   =goal>
     isa icon-search
     state distractor1
     feature1 =thefeature
     color =thecolor
     location =loc
   =loc2>
     isa visual-location
     time now
     attended nil
     value =thefeature
     color =thecolor
   =state>
     isa module-state
     module :vision
     processor free
   =state2>
     isa module-state
     module :motor
     modality free
==>
   =goal>
     state distractor2
     location =loc2
     !send-command! :motor prepare ply :hand right :r 26.0 :theta -2.55 )

(p find-icon-2
   =goal>
     isa icon-search
     state distractor
     feature1 =thefeature
     color =thecolor
     location =loc
   =loc2>
     isa visual-location
     time now
     attended nil
     value =thefeature
     color =thecolor
     nearest current
   =state>
     isa module-state
     module :vision
     processor free
==>
   =goal>
     state distractor2
     location =loc2)
```

```
(p text-first
   =goal>
     isa icon-search
     state distractor2
     feature1 =thefeature
     color =thecolor
     location =icon-loc
   =text-loc>
     isa visual-location
     time now
     attended nil
     kind text
     nearest =icon-loc
   =state>
     isa module-state
     module :vision
     processor free
==>
   !eval! (mark-as-attended (wme-to-xy =icon-loc))
   !send-command! :vision move-attention :location =text-loc
   =goal>
   state name)
```