BEHAVIOUR & INFORMATION TECHNOLOGY, 1999, VOL. 18, NO. 2, 68 – 82

The use of icons and labels in an end user application program: an empirical study of learning and retention

SUSAN WIEDENBECK

Faculty of Computer Science, Dalhousie University, Halifax, Nova Scotia, Canada B3J 2X4

Abstract. This research compared the learning of an application program whose interface was implemented using buttons with text labels, icons, or a fully redundant combination of icons and text labels. The objective was to: 1) evaluate the success of novice computer users in initially learning to use the application and in later use in a delayed session and 2) measure users' attitudes toward the application. Each session was divided into four blocks, and performance in the blocks was measured in terms of correctness of the tasks performed, time to perform tasks, and number of times the help facility was accessed. In addition, at the end of each session the participants' perceptions of the ease of use and usefulness of the software were measured. The results showed that in the first session performance was best on the label-only and icon–label interfaces. Performance on the icon-only interface was much poorer in session 1, particularly in terms of time and help references, but improved in session 2 to the point where it approached the performance on the other interfaces. Retention of skill between the initial and the delayed session was worse for the icon-only interface, but the effect was short-lived. Perceptions of ease of use were consistently better for the icon-label interface than for the other two interfaces. Perceptions of usefulness were higher for the icon-only and icon–label interfaces than for the label-only interface in the first session. Perceptions of usefulness became more positive for the icon-only group in the delayed session, but did not change for the other groups.

1. Introduction

During the past decade graphical user interfaces have become a common method of interaction between the user and the computer in end user application programs, such as word processors, spreadsheets, databases, and electronic mail. Graphical user interfaces present information to the user by icons rather than textual descriptions. Icons are interface objects that represent in a simplified pictorial fashion larger, more complex, and harder-to-grasp system objects. In the everyday useage of term, which we adopt here, any graphic on a manipulable button is referred to as an icon, regardless of whether it represents some larger entity. Icons can be used to represent objects known to the system, such as files and folders, and to represent actions that a user may carry out, such as print a file or create a folder. The user chooses icons with a mouse or other pointing device to initiate actions.

Icons have been used on a limited basis since the early days of computer graphics. The popularization of iconic representation in the interface dates to the work of David Canfield Smith and his colleagues who developed the interface of the Xerox Star workstation (Smith *et al.* 1982; Johnson *et al.* 1989). Icons are used in the interface because they are presumed to facilitate the human–computer interaction. It is easy to find enumerations of the supposed advantages of icons in popular literature. Some claims that have been made are the following (Horton 1994): icons improve the productivity and reliability of work; icons are better than words for representing subtle visual and spatial concepts; well-designed icons save space; icons speed search; icons lead to immediate recognition; icons lead to better recall, icons reduce the necessity of reading; and icons make interfaces more international. All of these claims about icons implicitly compare icons to text in the interface. Many of the claims are psychological or perceptual in nature: that icons are easier to process than text.

Today application programs make heavy use of icons in place of conventional menus. Items that formerly were represented by text choices in pull down menus are now available on toolbars, which contain icons on which the user clicks to carry out actions. Thus, the menus have migrated onto the screen and been iconified. These toolbars are under the control of the user, who can hide them or show them on the screen as desired.

Behaviour & Information Technology ISSN 0144-929X print/ISSN 1362-3001 online © 1999 Taylor & Francis Ltd
http://www.tandf.co.uk/JNLS/bit.htm
http://www.taylorandfrancis.com/JNLS/bit.htm

Over time, the use of icons has been exploited more and more in the design of end user applications, and today the user is likely to have the choice of toolbars for basic as well as more specialized operations. In complex modern software, the person who uses only the more basic of the toolbars can quickly have 25 to 50 icons visible on the screen, each perhaps 1 cm or less in size. In software that has only a few toolbars or in which toolbars are not expected to be displayed simultaneously, the icons may be larger and then fixed textual labels can be attached to the icons. Usually there will be an option to display or hide these labels. However, in more complex software with small icons, it becomes impossible to display fixed labels for all the icons. Then pop up labels may be available, which can be turned on or off by the user. However, some users do not like to use the pop up labels because they hid other icons when they pop up, and they fail to pop up when the pointer is very close to the icon but not directly over it.

Despite the prevalence of icons in the interface and claims made for their superiority, there have been relatively few empirical studies of the use of icons in end user application programs of realistic complexity. In this paper we report on an empirical study comparing the use of pictorial icons with text labels in the interface of an application program. Our objective was to investigate the learning and retention of skill in an end user application, when the interface was implemented with icons, text, or a fully redundant combination of icons and text. Using a carefully designed interface of realistic complexity, we hoped to determine whether the presumed advantages of icons can be verified in practice. Both learning and retention were chosen for study because the relative advantages of icons and text may differ depending on which one studies.

The remainder of this paper is organized as follows. Section 2 describes iconic interfaces and the empirical work that has been done concerning them. Section 3 presents the research questions. Section 4 presents our methodology. Section 5 describes the results of the experiment. Section 6 discusses the experiment, and Section 7 is the conclusion.

2. Icons and their use in the interface

2.1. *Background to the research*

A key conern in the design of iconic interfaces is effectively depicting the underlying referent of the icon. Potentially an icon can represent not just the referent but its attributes, associations, and state (Gittens 1986). Icons are sometimes classified by the relationship, or mapping, between the icon and its referent. Familant and Detweiler (1993) make the point that all iconic depictions are abstractions that represent only a subset of the features of the referent. However, icons differ in the degree of abstraction. Several icon classifications exist in the literature (e.g. Blattner *et al.* 1989, Familant and Detweiler 1993, Gittens 1986, Rogers 1989). A simple classification that is sufficient for our purposes is that of Blattner *et al.* (1989), which classifies icons as representational, abstract, and semi-abstract.

Representational icons are simplified images of familiar objects or operations, for example, an image of an eraser to indicate a selective deletion operation. Their concreteness is meant to make the mapping between icon and referent obvious. However, often there is no obvious pictorial representation or none that can reasonably be implemented. Then *abstract* icons may be used, which employ geometric shapes or graphic symbols instead of concrete images, for example, an exclamation point to represent the operation of executing a program. According to Blattner *et al.* (1989), *semi-abstract* icons combine a representational pictorial element with an abstract symbol, for example, a folder with an arrow that indicates placing items in it.

There are several reasons why the use of icons might have advantages over text in terms of human–computer interaction. These advantages, as well as some possible disadvantages of iconic representations in computer systems, are summarized by Gittens (1986), Rogers (1989), Shneiderman (1997) and Benbasat and Todd (1993). Benbasat and Todd caution that the discussion of icons often fails to distinguish between fundamental properties of icons and implementation issues. Thus, some advantages claimed for icons over text may not be inherent in icons but may be the result of the way that icons and text labels are usually implemented in computer systems. While the distinction between inherent properties and implementation may not be as clear as Benbasat and Todd suggest, their point is well taken. In this discussion we concentrate on inherent properties of icons which may distinguish them from text.

First, there is the possibility that icons lead to faster recognition. Shepard (1967) showed that recognition of visual images is superior to either recognition of words or sentences. Standing (1973) went on to show that humans have an almost unlimited ability to recognize images that they have seen before. Although advantages have been found for visual images over text in simple recognition tasks, it is not certain whether the advantages are practically significant in human–computer interaction tasks of realistic complexity.

Second, the use of icons may allow users to devote more cognitive resources to the primary task they are trying to accomplish (Benbasat and Tood 1993). This argument is based on a resource pool model of attention

(Navon 1984), which asserts that there are distinct resources available for different activities, such as perception and cognition. If the primary task is a problem solving task which requires cognitive resources, then the use of icons, which draw on a perceptual resource pool, may make more resources available for the primary task. A text-based interface, on the other hand, might compete with the primary task for cognitive resources. Again, it is not clear to what extent this proposed advantage would be significant in a practical settting of human–computer interaction.

Third, icons may aid the learning of a system. According to assimilation theory (Ausubel 1968), meaningful learning occurs when a person works with material to draw substantive connections between new information and relevant anchoring concepts in long-term memory. As a result of working with the material, a deep understanding of basic concepts of the material is achieved. After meaningful learning, a person should be equipped to combine the information in novel ways and to extend it to new situations. In terms of assimilation theory, an interface that represents objects in a concrete, familiar form may aid the learner in assimilating new computer concepts to a related base of concepts in memory. This assimilation is possible because of the analogies of the new situation to known situations. Extended analogies, or metaphors, have been argued to be important in computer learning because they aid users in assimilating new knowledge to related existing knowledge (Carroll and Mack 1985).

The literature on icons identifies their ability to support an interface metaphor as an important property (Familant and Detweiler 1993, Gittens 1986). Benbasat and Todd (1993) point out that useful analogies could potentially be evoked by text as well as icons, so this characteristic might be considered a matter of implementation rather than a fundamental property of icons. However, others aruge that an ensemble of icons has the potential to quickly and effectively evoke a concrete interface metaphor (Davis and Bostrom 1993, Davis and Wiedenbeck, 1998). This evocative power of a group of icons comes from the ability to represent the referent of the individual icon, as well as its key attributes and associations, and to suggest relationships of the icons to each other in a small space (Gittens 1986), something which is difficult to achieve with text labels consisting of only a word or two. Similarly, in their discussion of visual representations in problem solving, Larkin and Simon (1987) argue that there may be fundamental reasons why visual representations are superior to verbal descriptions: 1) they group together information that describes an individual element and that is used together; and 2) they support the making of a large number of perceptual inferences. Thus, icons do have arguable advantages in an interface metaphor. However, in order to support recognition of individual icons and take advantage of their role as part of an interface metaphor, icons must be immediately recognizable to the targeted user population and use graphic conventions familiar to that group (Familant and Detweiler 1993). Hutchins *et al*. (1985) define the *articulatory distance* of an icon and its referent as the difficulty of inferring the referent from the icon. Icons which have greater representational content have a more direct mapping between form and function and, in general, are expected to have smaller articulatory distance. Clearly, the meaning of abstract icons must be learned by beginners, since abstract icons do not have a direct mapping between the pictorial representation and the referent.

Icons might also ease learning and increase acceptance of a system if they give an immediate impression that the system is easy to use to the first time user (Rogers 1989). Attitudes toward usage represent the user's affect about a computer system. The 'Technology acceptance model' of Davis (Davis 1989, Davis *et al*. 1989), argues that learners' perceptions of a computer system play a large role in the formation of their intention to use the system and in its ultimate acceptance and use. According to Davis, two kinds of perceptions are important in the acceptance of a system: perceived usefulness and perceived ease of use. Perceived usefulness (PU) is defined as the user's subjective perception of the extent to which the system or software will aid in job performance. Perceived ease of use (PEU) is defined as the extent to which the user expects a system or software to require low effort to learn and use. The relationship of attitudes to the behavioral intention to use a system is that, all other things being equal, people will form an intention to use a system about which they have positive attitudes. Davis argues that perceptions about ease of use are strongly affected by surface features of the system, such as the presence of a mouse, the use of icons in the interface, color, information presentation formats, etc. On the other hand, perceptions about usefulness of a system are based on two factors: 1) the underlying functionality of the system itself and, in particular, users' judgments of the value of that functionality in the attainment of their own goals; and 2) the learner's perception of how easy it will be to accomplish goals in the system. This suggests that perceived ease of use influences system acceptance both directly by beliefs about the amount of effort required to use a system and indirectly by modifying the user's perceptions of usefulness of the system. As a salient interface feature, icons may be important to acceptance if they do indeed, as suggested by Davis, give a positive early impression and thus support the formation of positive perceptions of ease of use and usefulness.

2.2. *Empirical studies of the use of icons in the interface*

Several experimental studies have been done on the use of icons in the interface, although there is less empirical work on the subject than one might expect. For our purposes we divide the studies into two general classes: 1) the design of icons; and 2) learning and performance studies comparing icons to text. Our interest in the design of icons stems from our desire to create a system with well-designed icons for use in our experimental comparisons. Experimental work on the design of icons has found that:

1. Icons depicting concrete objects tend to be most effective (Blankenberger and Hahn 1991, Rogers 1989). This is consistent with Hutchins *et al.*'s (1985) argument that a small articulatory distance aids recognition.
2. Icons embodying an isolated analogy (as opposed to a complete interface metaphor) are often difficult for computer users to interpret, even concrete analogies, e.g. a wheelbarrow loaded with bricks to represent a move operation (Rogers 1989).
3. People are able to learn the arbitrary relationships of abstract icons to their underlying referent when only a few are used but not when they were embedded in a whole set of abstract icons (Rogers 1989).
4. Icon sets with more visual variety are easier to locate in a display (Bewley *et al.* 1983). This issue of discriminability may explain some cases in which abstract icons have proven to be more successful than concrete icons (e.g. Arend *et al.* 1987).
5. Visually simple icons work better than complex icons in icon search (Byrne 1993).
6. Positional consistency in presentation of icons on the screen has a strong effect on usability and, once learned, helps compensate for initial differences in recognizability (Green and Barnard 1996, Lansdale 1988).

Several studies specifically focusing on the value of icons as opposed to text in information retrieval from databases have compared searching category hierarchies implemented with text, icons, or some combination thereof. A study by Muter and Mayson (1986) of question answering in a videotex system compared menus with text only to menus with text plus simple, concrete graphics representing information categories in the system by exemplars (e.g. a line drawing of a coat to represent the category 'clothing'). The result showed no difference in time to respond to the questions, but an advantage to the text plus graphics condition in accuracy. Muter and Mayson speculate that the graphics provided extra information by providing the exemplar to supplement the category name. They did not include a graphics-only condition in their study. Egidio and Patterson (1988) compared menus using icons and text in a database searching task including text-only, icon-only, and text plus icon representations. Their specific task was retrieval of pictures from a hierarchically organized catalog of animals. Like Muter and Mayson, the icons were exemplars that showed typical examples of categories of objects in the catalog, using simple line drawings. The text labels were one word labels, such as 'vertebrate'. Egidio and Patterson found that the icon plus text condition was best in terms of proportion of trials completed and extra decisions made, the icon-only condition was intermediate, and the text-only condition was poorest. The authors believe that these results reflect the disambiguating effects of pictures in catalog searching. However, in terms of total time to complete a search, there were no differences among the three conditions, and in terms of think time the label-only condition actually tended to be faster than the conditions involving pictorial representations. The authors speculate that the participants in the two icon conditions may have spent longer because they enjoyed looking at the pictures. In other related work which supports the finding of Egidio and Patterson, Bewley *et al.* (1983) showed that icons with text labels were easier for participants to learn to recognize correctly but subsequently took more time for participants to choose from a display than unlabeled icons. However, it was also found that, once learned, there was little difference among icons in recognizability. Similar results on labeling of icons are reported by Byrne (1993), who found that search for file icons was faster when the participant was explicity given the file name.

Two studies comparing text-only to icon-only conditions in non-hierarchical applications domains also failed to find an unambiguous advantage for icons. Rohr and Keppel (1984) found no advantage to buttons labeled with icons over buttons labeled with text in a simple word processing simulation. In another study involving a simple electronic mail simulation, Benbasat and Todd (1993) compared performance on interfaces implemented with buttons using icons or text labels. They found no difference between the iconic version of the interface and the text version. They suggest that most of the advantages claimed for icons can probably also be realized in text.

While there is a substantial body of research on performance issues related to the use of icons in the interface, few studies have investigated the effect of icons vs. text on learners' perceptions of computer systems. Davis *et al.* (1989) suggest that icons and other interface features that make systems appear more accessible to the new user may have the effect of increasing the perceived ease of use of a system and the user's corresponding behavioral intention to learn and use it. With positive perceptions about a system, users are more willing to persevere in learning it in the face of any initial difficulties. Some studies have found greater perceived ease of use or perceived usefulness in systems using iconic representation (Davis and Bostrom 1993, Wiedenbeck and Davis 1997). However, these studies involved systems which differed in other ways in addition to the use of icons vs. text, so the results are not wholly comparable to the present situation which concerns the use of icons vs. text in systems which are otherwise identical.

The research reported here makes several contributions to the understanding of the use of icons in the interface. First, it compares three interface alternatives; icons, text labels, and icons plus text labels. Most past research has not included all three of these alternatives. Second, the study is carried out using a common end-user application program that is of realistic complexity. Past research has often used more specialized applications or very simplified 'toy' systems with only a small number of icons. Third, learning and retention are measured over two sessions and multiple measures are used, including performance time, errors, and use of help. Finally, we address the issue of user perceptions of interfaces implemented with icons only, text labels only, and icon plus text labels. This is a largely neglected question that is, nevertheless, important for user acceptance of interfaces.

3. Research questions

Our first research question centers on the effect of interfaces implemented with icons, text labels, or a combination of icons and text labels in initial learning of an application program in an unknown domain. Effects on performance may be seen in three areas: correctness of tasks, time to complete tasks, and use of help. In terms of task completion, we expect that the icon–label interface will be most successful in initial learning because of the redundancy of providing both icons and text labels. However, the differences should be fairly small if the interfaces are well designed and tested. On the other hand, we expect to find greater differences in time to complete tasks and the use of help. As summarized above, several arguments suggest that pictorial representations may lead to an advantage in speed of processing. However, in initial learning, any such advantage is likely to be masked by other factors: 1) learning of the meaning of the icons, which even with concrete representational icons may slow the learner compared to text labels; and 2) the visual appeal of the icons, which may lead the user to spend more time viewing the icons compared to text labels. Thus, we expect that in initial learning the label-only interface will be faster than the icon-only interface. The icon–label interface is expected to be similar in speed to the label-only interface. Icons with text labels are redundant and would normally be expected to slow the user. However, the combination of icons with text has the possibility of disambiguating the meaning of the menu choices available in the interface. We hypothesize that in initial learning, where the user is not familiar with the representation of the choices, this redundancy will be an advantage that speeds performance. We believe that the aid in disambiguation is likely to make up for any loss of speed associated with the visual appeal of the icons. In terms of use of help, we expect that the icon–label interface will lead to the lowest use of help, because of its redundancy. We expect that the label-only and icon-only interfaces will require more help, particularly the icon-only interface, which in initial learning requires interpreting the pictorial symbol as well as mapping it to a desired operation in the system. By contrast, in the label-only interface the meaning of the label (in its everyday, natural language sense) will be clear, so the only interpretive problem is mapping to the desired object or operation in the system.

Our second research question concerns performance in the subsequent use of interfaces with icons, text labels, and a combination of icons and text labels after a time lapse. Thus, it has to do with retention and continued learning. As in initial learning, we expect the differences between interfaces in correctness of tasks to be small. In subsequent use, we also expect that the three interface forms will be equal in terms of time. At this point the redundancy of the icon–label interface will have lost its advantage. However, we still do not expect a disadvantage from the redundancy because: 1) any visual appeal will have worn off through familiarity; and 2) users come to take advantage of the positional consistency of items in an interface, which makes the nature of the representation of the item less important. We expect use of help to be dramatically reduced in all conditions, but that it will remain higher for the icon-only interface than for the other two because of the need to refresh knowledge of the meaning of the icons.

Our third research question concerns the perceptions of learners using the different interface forms both in initial learning and in a later test of retention. The 'Technology acceptance model' predicts a positive influence on perceptions of ease of use of several features associated with graphical user interfaces, including the use of icons (Davis 1989, Davis *et al.* 1989). Based on the theory and previous experimental work (Davis and Bostrom 1993), we expect that both interface forms incorporating icons will be perceived as easier to use than the label-only form in initial learning. The theory is not clear about the persistence of effects of interface features in later use, but there is some experimental indication of persistence of patterns of response to the PEU in similar interfaces over time (Wiedenbeck and Davis 1997). Thus, we tentatively propose that for all interface forms PEU will improve with further exposure to the system, but the relative differences between the interfaces will remain stable. With respect to perceptions of usefulness, theory argues that differences could come from two sources, the user's perception of the value of the functionality of the software itself and also the user's perception of how difficult it will be to achieve goals with the software (Davis *et al*. 1989). In our research, the functionality of the three systems is identical, so any differences should come from the modifying effects of perceived ease of use on perceived usefulness. Thus, we expect that the same pattern of results will be observed for PU and PEU, i.e. interface forms using icons will have higher perceived usefulness than the label-only interface in initial use. In subsequent use, we expect improved perceptions of usefulness for all groups but that the pattern of differences between groups will remain stable.

4. Methods

4.1. *Participants*

Sixty volunteers took part in the study. The participants were recruited from three undergraduate courses in business administration. Participation in the experiment was offered as an alternative to a homework assignment in one part of the course. Participants received credit for participating but did not receive a grade. The mean age of the participants was 21.5 years old. Sixty-two per cent of the participants were female and 38 per cent male. The average self-reported grade point was 3.08 on a scale ranging from 0 to 4. Participants filled out a questionnaire about their previous computer experience. This showed that on average they had taken one college course on computer literacy. They had also taken an average of 2.6 courses in high school or college in which computers were used as a tool (e.g. use of word processors or spreadsheets in class assignments). None of the participants had experience with electronic mail, which was the applications area of the experiment. On average, the participants' first use of computers was in junior high school. The mean reported previous Macintosh use was 2.0 on a scale from 0 to 4. This corresponded to using the Macintosh 'sometimes'. Three point scales were used for self-ratings of mouse and typing ability. For frequency of mouse use the mean rating was 1.5 (0= 'never', 1= 'a little', and 2= 'a lot'). For typing ability participants rated themselves on average as 1.1, corresponding to '30 words per minute error free'.

4.2. *Materials*

A simulated electronic mail program was used in the study. Three versions of the program were created to represent the three interfaces under study: label-only, icon-only, and icon–label. The three variations were superimposed in the same program so functionality and response times were the same. Clickable buttons represented the objects and actions in the system. The placement of buttons on the screen and their size were the same in all three versions. The only difference was whether the choices on the screen were represented by icons, text, or a combination of the two. There were 32 icon or text buttons used in the interfaces.

The conceptualization of the electronic mail system was similar to that of many existing e-mail systems. They system consisted of a storage area for incoming mail and mailboxes for storing mail that the user wished to retain on a more permanent basis. There was also a scratch area for composing meassages to be sent. The electronic mail program allowed the user to view a list of incoming mail, read incoming mail, print it, save it in a mailbox, delete it, reply to it, and forward it. The user could also view a list of messages previously stored in a mailbox, read a previously stored message, print it, or delete it. An option also existed to delete all messages stored in a mailbox and transfer mail from one location to another. The user could send mail by specifying an address and typing a message, and could send copies of the message to any number of others. The user could edit messages as often as desired before sending them, and could cancel a message any time before pressing the send button.

The icons were designed following principles in the literature on icons, as summarized in Section 2.2. Concrete representational icons were used to reduce the articulatory distance, for example, a drawing of a printer. The use of abstract icons was limited to two, and there were chosen to be as interpretable as possible based on conventions likely to be known to users, e.g. a circle with a diagonal line through it to mean cancel. Overall, the ensemble of icons were designed to suggest a metaphor of sending and receiving mail manually in an office setting. The icons in the set were designed to have visual variety to enhance discriminability. They were visually simple, not incorporating great detail. The labels consisted of one to three words, e.g. 'Delete' or 'Send copy'. In the icon-only interface the icons took up most of the space on the buttons, which were about 2.5 to 3 cm^2. In the label-only interface, the text label was written in large bold type and also took up most of the space on the buttons. In the icon–label interface, the icon was the same size as in the icon-only interface and the label was printed in small bold type below it. Examples of icon-only, label-only, and icon–label buttons are given in Figure 1. The different versions of the interface were pre-tested. Tests of icon recognizability and of matching icons to functions were carried out, as suggested in previous research (Bewley *et al.* 1983, Nolan 1989). The labels were tested through a test matching the label to its function. Based on these results, icons and labels were redesigned to improve those with which the pilot participants had difficulties.

The functions were arranged on the screen to suggest sets of actions which could be carried out on an object and thus a model of the electronic mail task. For example, the reply function was grouped with Read, Save, Print, Forward and Delete, since these are all actions that can be carried out on an incoming message. It was not placed with the Write function, which is physically similar to it in the sense that the user types a message, but conceptually different because the user is initiating the communication rather than processing a message received. All interaction was carried out by use of the mouse, except for the typing of the message text and address of the receiver. To activate a function the user made a single click on the appropriate button. In addition to the buttons representing the functionality of the system, there were navigational buttons for moving around the system, e.g. from the opening screen, to the area containing incoming mail, to the mailboxes for stored mail. The buttons remained in fixed positions on the screen throughout the study.

When an action was carried out, feedback was given. For navigational commands this consisted in a change to a new screen, for example a change from the opening screen to the screen showing the list of incoming mail. Each screen was labeled with a heading, such as 'New Mail', to aid the user in keeping track of the current location. The feedback for the command to read a message consisted of the appearance of the message on the screen and for the command to compose a message it consisted of the appearance of a blank typing area. For most other functional commands the feedback consisted

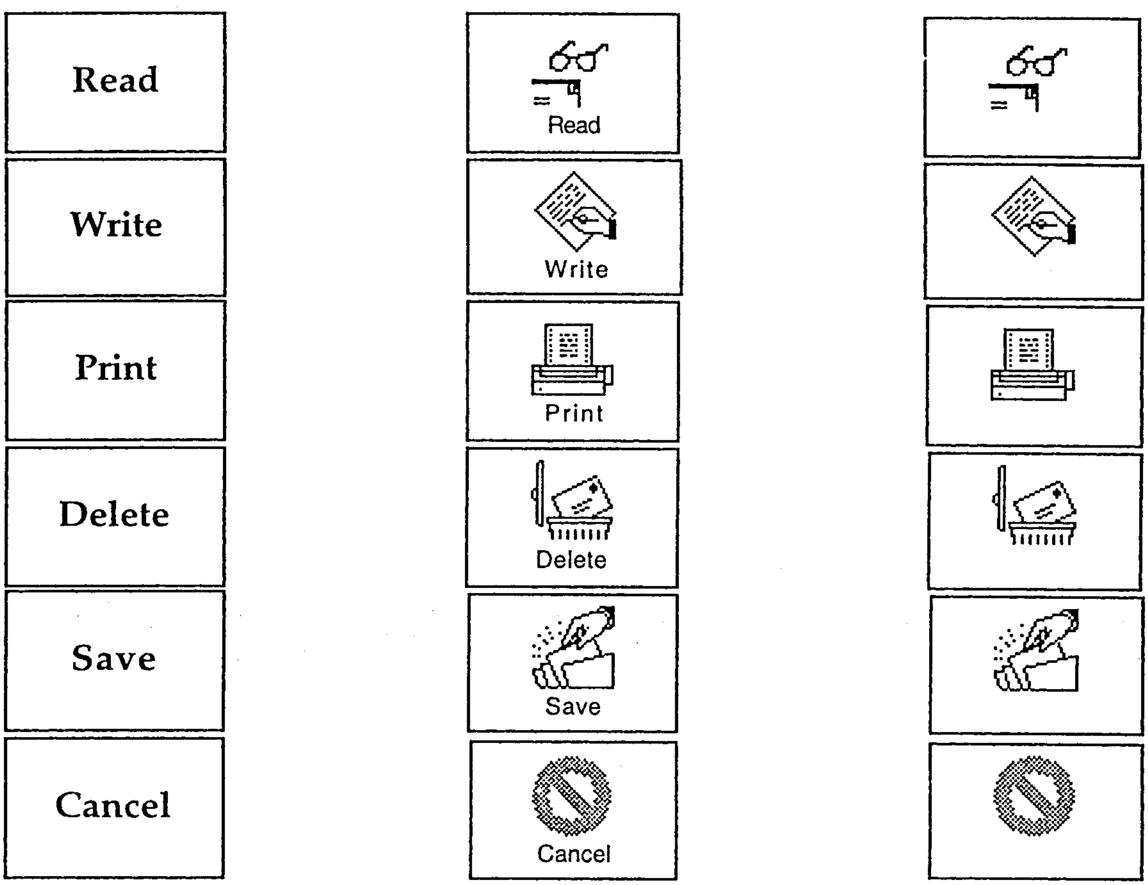

Figure 1. Examples of buttons used in the label-only, icon–label, and icon-only interfaces.

of a confirmatory message displayed in a fixed area of the screen, for example 'Message from Foster now being printed'. Error messages were given in the same area. Because users were guided by the available choices, they could not make many syntactic errors. However, some errors were still possible, e.g. attempting to send a message without specifying a recipient.

For each interface help was provided. The help consisted of on-line step-wise instructions explaining how to carry out tasks. Help was provided for all of the activities supported by the application. The wording of the help was almost identical for the different version of the interface. An index was provided to help the user find the desired functionality quickly. Help was accessed through a help button.

4.3. *Method*

A $3 \times 2 \times 4$ design was used with three interfaces, two sessions, and four blocks of tasks per session. Interface was a between subjects factor, while session and block were within subjects factors.

Participants were run individually and were randomly assigned to one of the three interfaces. They participated in two sessions spaced one week apart, eash lasting about one and a half hours. In the first session, participants filled out a background questionnaire on computer experience. They were then given a hands-on tutorial to learn the conventions of their interface. The tutorial software, which had three versions like the experimental software, simulated the control of a video game. Participants practiced using the mouse to navigate the system and to choose operations. They also learned to look for feedback and interpret it.

After the training tutorial, the participant read a half page description of the functions of the e-mail system. The experimenter answered questions, then set the program to the opening screen and gave the participant the task set. The task set was divided into four blocks of ten tasks each. The tasks exercised all of the operations available in the software, read, write, send, forward, reply, save, delete, etc. The four blocks were equivalent in that they involved carrying out the same operations using different arguments (e.g. 'send e-mail to Peter Wilson on the subject of class scheduling' and 'send e-mail to Allen Kent on the subject of vacation shutdown'). The order of tasks within blocks was randomized.

Participants worked through each block of tasks in order. They were allowed to work on a task as long as they wished but could not come back to a previous task once they had left it. At the end of a block the participant took a short break, if desired, before beginning the next block. The time per block and the number of accesses to the help facility were recorded by the software. A videotape was made which was used to score the correctness of the tasks. An experimenter was present at all times. The experimenter did not answer questions but did intervene in the few cases where unexpected events occurred. After the fourth block, perceived ease of use and perceived usefulness were measured using Davis *et al.*'s (1989) validated four-item scales.

In the second session, held one week later, the same procedure was followed except that the training tutorial was not repeated. The task set was the same as in the first session except for changing the arguments of commands. PEU and PU were measured at the end of the session.

5. Results

Correctness of a task was scored as 1 if the task was completely correct or 0 if it contained errors. No partial credit was given. Since there were 10 tasks per block, the maximum correctness score was 10. All tasks were scored by two judges working independently. In the case of disagreement the judges reviewed the tape together and resolved the difference by discussion. The use of help was measuerd by the number of references to the help per block. The time score consisted of the time in seconds to complete a block of 10 tasks. Each block contained two tasks of sending and replying to electronic mail, which required the participant to type the body of a mail message at the keyboard. This was the only significant use of the keyboard, since all other actions except for one-word electronic mail addresses were done by mouse clicks. To compensate for differences in typing speed, the time to type the body of the message text was substracted from the total block time. The PEU and PU were scored by summing the ratings on the four items. Scores ranged from 4 (highested perceived ease of use/usefulness) to 28 (lowest perceived ease of use/usefulness). The reliability of both PEU and PU was high. Cronbach's alpha for the PEU was 0.93 and for the PU was 0.92. This is similar to the reliability reported in other research (Davis 1989, Davis *et al.* 1989, Adams *et al.* 1992, Davis and Bostrom 1993).

For correctness, times, and help references, separate three-way mixed model ANOVAs were run. Interface was the between subjects factor, and session and block were within subjects factors. Follow-up testing was done with Newman-Keul's test, using an alpha level of 0.05. See Table 1 for the means and standard deviations of correctness, time, and help references.

The results for correctness are graphed in Figure 2. Interface was significant ($F(2,57)= 3.59$, $p < 0.05$). Follow-up testing using Newman-Keul's test showed that there were more errors by the icon-only group then by either of the other groups. The other two groups did not differ significantly. Session was also significant (session: $F(1,57) = 54.45$, $p < 0.01$) with session one scores lower than session two scores. In addition, block was significant ($F(3,55) = 9.73$, $p < 0.01$) with significant differences between all blocks except blocks three and four. None of the two-way interactions was significant, nor was the three-way interaction.

The time results are shown in Figure 3. Interface was significant ($F(2,57) = 10.72$, $p < 0.01$), session was significant ($F(1,57) = 311.68$, $p < 0.01$), and block was significant ($F(3,55) = 205.02$, $p < 0.05$). The two-way interactions were significant: interface by session ($F(2,57) = 8.34$, $p < 0.01$), interface by block ($F(6,112) = 7.85$, $p < 0.01$), and session by block ($F(3,55) = 55.42$, $p < 0.01$). The three-way interaction was significant ($F(6,112) = 2.69$, $p < 0.05$). Follow-up testing showed that *within interfaces*,

Table 1. Means and standard deviations (in parentheses) of correctness, help, and time.

		Session 1				Session 2			
		Block 1	Block 2	Block 3	Block 4	Block 1	Block 2	Block 3	Block 4
Icon-only	Correct	8.45	8.60	8.75	9.15	9.20	9.35	9.40	9.70
		(1.32)	(1.31)	(1.45)	(1.14)	(1.11)	(0.99)	(1.05)	(0.73)
	Help	10.05	2.70	1.70	0.45	2.00	0.75	0.50	0.20
		(7.07)	(2.30)	(1.81)	(0.89)	(2.13)	(1.25)	(1.24)	(0.52)
	Time	719.30	357.15	305.95	231.35	333.15	218.10	211.60	183.30
		(252.10)	(132.52)	(109.91)	(65.83)	(131.18)	(72.43)	(61.78)	(29.80)
Label-only	Correct	8.80	9.25	9.50	9.65	9.75	9.75	9.80	9.65
		(1.01)	(1.16)	(1.15)	(0.81)	(0.55)	(0.55)	(0.52)	(0.67)
	Help	0.80	0.25	0.35	0.05	0.05	0.15	0.15	0.05
		(1.70)	(0.79)	(0.88)	(0.22)	(0.22)	(0.50)	(0.22)	(0.22)
	Time	485.00	275.90	261.55	213.60	262.65	178.90	177.50	168.45
		(131.67)	(181.33)	(87.28)	(40.39)	(49.01)	(34.16)	(21.72)	(20.43)
Icon – Label	Correct	9.10	9.30	9.70	9.60	9.60	9.65	9.80	9.90
		(0.85)	(1.17)	(0.66)	(0.75)	(0.60)	(0.75)	(0.41)	(0.31)
	Help	1.95	0.95	0.25	0.15	0.15	0.05	0.15	0.05
		(3.28)	(1.54)	(0.44)	(0.37)	(0.50)	(0.22)	(0.37)	(0.22)
	Time	475.40	263.55	233.55	188.90	246.35	170.50	181.75	165.70
		(178.40)	(57.05)	(45.84)	(30.59)	(39.98)	(62.61)	(46.35)	(33.73)

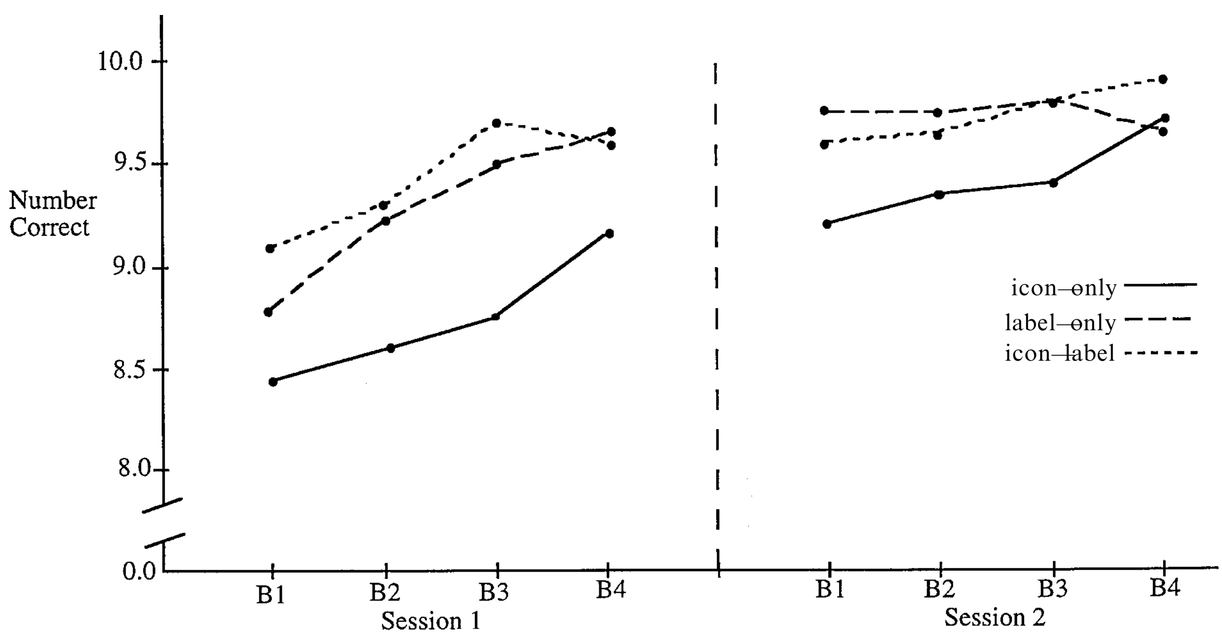

Figure 2. Mean correctness of tasks by session and block (out of a maximum of 10).

in session one, block one was significantly slower than blocks two, three and four for all interfaces, and block two was slower than blocks three and four for all interfaces. Block three was slower than block four only for the icon-only interface. In session two the only difference was that block one was significantly slower than blocks, two, three and four for the icon-only interface. *Between interfaces*, in session one, the icon-only interface was significantly slower than the label-only interface in blocks one and two and slower than the icon–label interface in blocks one, two and three. In session two, the icon-only interface was significantly slower than both the label-only interface and the icon–label interface in block one.

Help references are graphed in Figure 4. Interface ($F(2,57) = 24.50$, $p < 0.01$), session ($F(1,57) = 72.10$, $p < 0.01$), and block ($F(3,55) = 45.59$, $p < 0.01$) were significant. The two-way interactions were all significant: interface by session ($F(2,57) = 24.06$, $p < 0.01$), interface by block ($F(6,112) = 23.45$, $p < 0.01$), and session by block ($F(3,55) = 24.81$, $p < 0.01$). The three-way interaction was also significant ($F(6,112) = 11.39$, $p < 0.01$). Follow-up testing showed that *within interfaces*, in session one, the

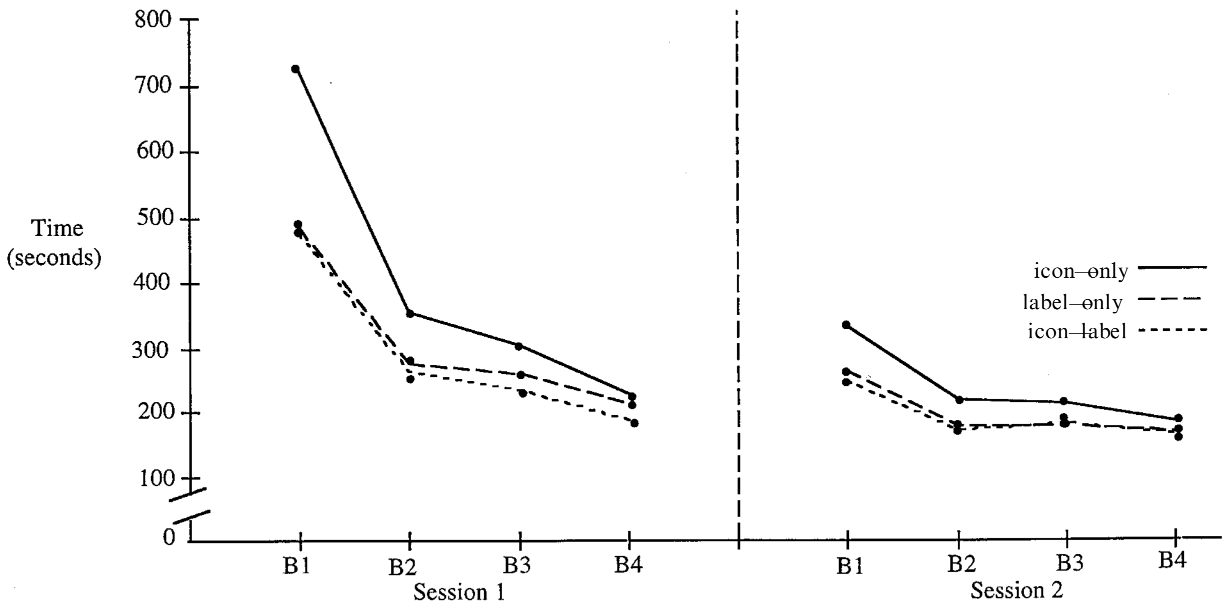

Figure 3. Mean time to complete tasks by session and block.

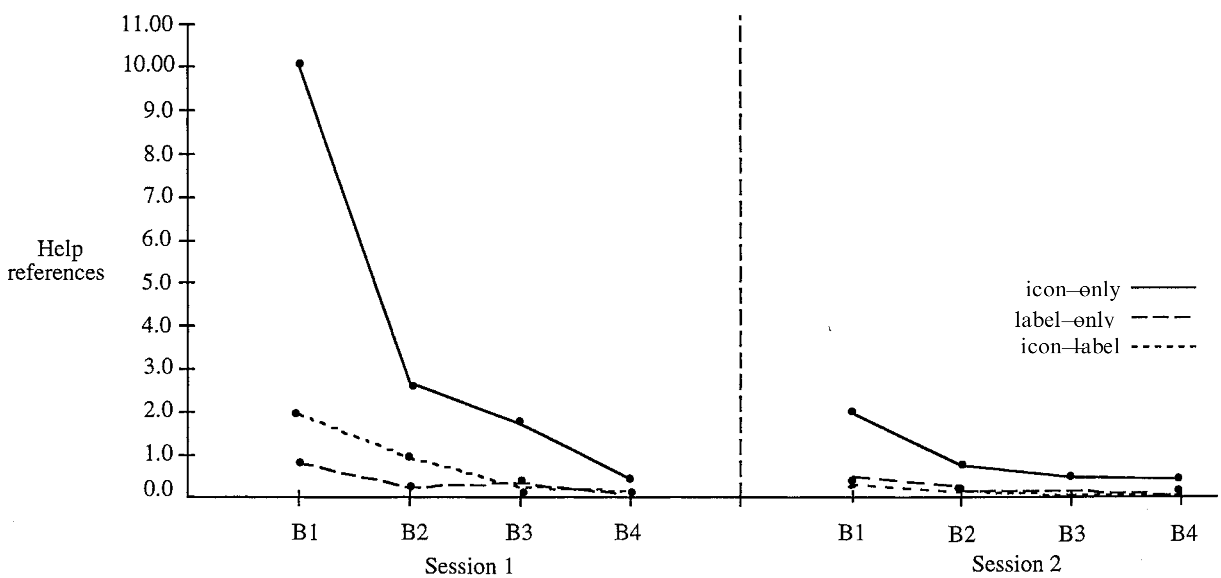

Figure 4. Mean number of references to help by session and block.

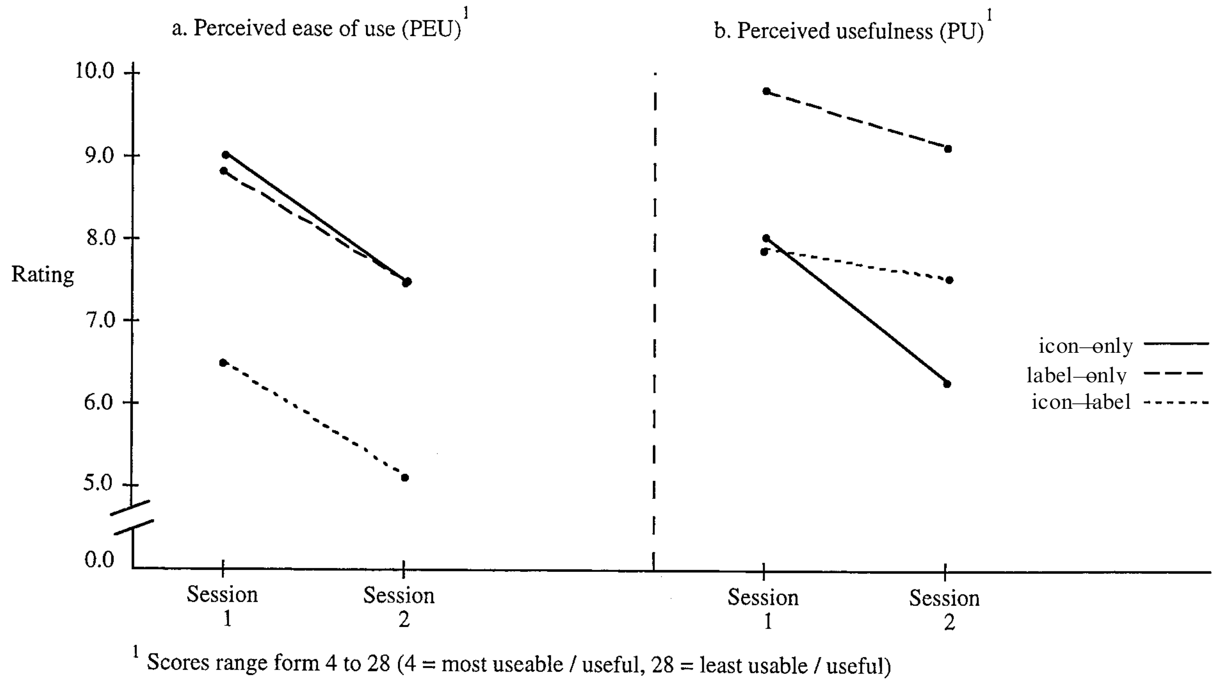

Figure 5. Mean perception ratings by session.

icon-only interface had significantly more help references in block one than in blocks two, three and four, more in blocks two than in blocks three and four, and more in block three than in block four. In session one there were no significant differences in help references within the label-only and icon–label interfaces. In session two the only significant difference within interfaces was that the icon-only interface has significantly more help references in block one than in later blocks. With respect to *between interface* comparisons, in session one, there were significant differences between all three interfaces in block one. In blocks two and three, the icon-only interface had significantly more help references than the other two interfaces, and in block four there were no significant differences. In session two, the icon-only interface had significantly more help references than the other interfaces in block one.

For both PEU and PU a two-way mixed model ANOVA was run with interface as the between subjects factor and session as the within subjects factor. The results of PEU and PU are shown graphically in Figure 5 and the means and standard deviations are shown in Table 2. For PEU the results showed that interface was significant ($F(2,57) = 4.90$, $p < 0.05$). Session was also significant ($F(1,57) = 102.02$, $p < 0.01$), with a more favorable perception of ease of use after session two. The interface by session interaction was not significant. Follow-up testing showed that in both sessions the perception of ease of use was significantly more favorable for the icon–label interface than for either

Table 2. Means and standard deviations (in parentheses) of PEU and PU.

		Session 1	Session 2
Icon-only	PEU	9.04 (4.07)	7.50 (3.58)
	PU	8.05 (5.59)	6.30 (2.76)
Label-only	PEU	8.95 (3.33)	7.50 (3.32)
	PU	9.80 (4.32)	9.15 (3.66)
Icon – Label	PEU	6.50 (2.28)	5.15 (1.57)
	PU	7.90 (3.57)	7.55 (3.54)

the icon-only or the label-only interface. The latter two did not differ from each other. The ANOVA for PU showed that interface was not significant. However, session was significant ($F(1,57) = 24.48$, $p < 0.01$). The interface by session interaction was also significant ($F(2,57) = 5.28$, $p < 0.01$). Follow-up testing showed that in session one the label-only group had significantly less favourable perceptions of usefulness than the other two groups. In session two the label-only group had significantly less favorable perceptions of usefulness than the icon-only group, but no other significant differences were found. The icon-only group changed significantly in its perceptions of usefulness from session one to session two, with more favorable perceptions of usefulness in session two than in session one. By contrast, perceptions of usefulness remained essentially static in the label-only and icon–label groups.

6. Discussion

Our focus was learners who had relatively low experience with computers in general and no experience with software in the target domain. The results indicate a learning advantage for the label-only and icon–label interfaces compared to the icon-only interface for correctness, time and use of help. The differences in time and help references were especially strong in initial learning, indicating that these are the areas in which the most is to be gained by learners who use interfaces with text labels or with labeled icons. On the other hand, the advantage in terms of correctness, although significant, was less exaggerated, suggesting that most participants could do most of the tasks regardless of the interface, simply by continuing to try different solutions until they were successful. The success of the label-only and icon–label groups suggests that in early learning text labels are more informative than icons. Our results generally agree with those of Egidio and Patterson (1988), who found that label-only and icon–label interfaces were faster than icon-only. In our case, the better performance was found on all three performance measures: time, errors and help references. On the other hand, this contrasts with the results of Benbasat and Todd (1993) and Rohr and Keppel (1984) who found no difference between performance using icons only and text only. One possible reason for the different results may be that the system used in this experiment was more complex and full-functioned. Greater functionality leads to more choices and perhaps more subtle differences among the choices. This may have made it more difficult for participants to interpret unlabeled icons. We expect that with a smaller number of icons the performance on our label-only interface would have been better. If the number of icons were greatly increased performance might have been worse than in the present experiment. With a very large number of icons, the problem of icon interpretation is compounded by the problem of having to scan large number of objects on the screen.

The lack of difference between the label-only and icon–label groups in most comparisons suggests that the labels were the element that aided learning. In addition to being close in terms of time and errors, these groups made little use of help, even in their first block of trials. They had little need of the detailed stepwise help instructions, and tooltips alone probably would have been sufficient help. On the other hand, learners in the icon-only group made extensive use of help and were observed by the experimenters to carefully follow the steps of the instructions. From these results, it appears that providing icons in combination with labels added little extra information of use in learning to manipulate the interface beyond what was conveyed by the labels. Another possible interpretation is that the icons did add some useful information, but the redundancy of the icons with labels slowed learners down compared to the label-only group and nullified the advantage of the extra information. However, this explanation does not seem likely. If the icons added useful information, we would expect the icon–label group to have been significantly more correct and have had significantly fewer help references than the label-only group, but this was not the case.

A key question about these results is why the icon-only group performed so poorly. The point has been made in the past that icons, even representational icons, have to be learned. In early learning the meaning of the icons must be interpreted, and the meaning may not be obvious to the beginner from a pictorial representation alone. This is especially true in a complex system with many icons. Of course, learners using text labels must also interpret the labels; but, as we have argued previously, their problem is simpler because they know the natural language meaning of the labels already and must only connect them with their function in the system. Icon-only users, on the other hand, have to first interpret the pictorial symbol *then* interpret its meaning in terms of system functionality. This dual burden is especially heavy in the case of learners who have low experience with both computers and the applications area. These learners have two disadvantages in interpreting icons. First, they do not known much about what the program they are learning is capable of doing. Without this knowledge, it is diffuclt for them to infer functionality from an icon alone. Second, they know little about the conventions of iconic representation. For example, more experienced learners know that a trash can-like object is often used to represent deletions, a file folder-like icon to represent a container for documents, a house icon to represent the home screen etc. Experienced learners can interpret these objects when they learn new interfaces, even though the objects are slightly different in form from those they have previously encountered. Thus, for learners more experienced with computers and the specific application, we would expect better performance with the icon-only interface than was seen in this experiment. While experienced learners will still have to learn unfamiliar icons, there will be fewer such icons and the learners will have a better basis for interpretation because of their past experience.

Although the icon-only group learned more slowly, made more errors, and used more help in early blocks of trials in session one, it should be noted that they closed the gap relatively quickly. Their early performance, although poorest of the three groups, was similar to that of the other groups by the end of the first session. Thus, they were disadvantaged by the lack of labels initially, but they were able to learn the meaning of the icons given hands-on experience with them, amounting to less than half an hour of intensive use. This suggests that the difficulties of interpretation of unlabeled icons can be overcome fairly rapidly.

The results of session two are relevant to the retention of knowledge gained in initial learning and continued improvement with more practice. First, with respect to retention of knowledge, it is clear that, after a time lapse of one-week, participants retained their ability to carry out the tasks correctly. Comparing the correctness on the last block of session one to the first block of session two, we see almost no difference. In terms of time, there was a small increase in time in the first block of session two compared to the last block of session one. This increase in time represents either recall or perhaps trial-and-error attempts to recreate the procedures needed to carry out the given tasks. The increase in time is greatest in the case of the icon-only interace, in which the mean time increased by about 1.75 minutes, compared to about 0.75 minutes in the other two groups. However, by the second block of session two the performance of the icon-only group returned to its level at the end of session one. A similar effect is seen in help references in session two. Here the label-only and icon–label interfaces experienced no increase in help references after a delay of one week, and their help references were very low. However, for the icon-only group help references increased substantially in the first block of session two and were significantly higher than in the last block of session one. Most of the increase disappeared in the block two of session two. Taken together, the results of time and help indicate that the icon-only group had greater retention problems than the other two groups, but the difficulties did not last long. However, it is worth noting explicity that this experiment studied only a retention interval of one week. We do not known what would happen if the retention period were longer, for example one month. One possibility is that the icon-only group would have much greater retention difficulties.

One question was whether the icon-label group would suffer in the retention period because the redundancy was no longer of value, while the visual appeal of the icons remained. We did not find this to be the case. First, we did not find a clear value to redundancy even in the first session, since the icon–label group did not perform better than the label-only group. Second, we did not find evidence that visual interest of the icons interfered with performance. Interference would be seen in the time measure, which did not differ between the label-only and icon–label groups in session two. If icons do evoke greater interest, it may wear off from familiarity, or over time users may rely more on the positional consistency of buttons to guide their choices and thus process the pictorial representation less.

The measures of perception provide some interesting results, particularly since, to our knowledge, no other studies have addressed perceptions in the comparison of iconic and text-based systems in a setting in which other features of the software were identical. We found that the perceived ease of use of the software improved from session one to session two for all groups. At the end of session two participants had a significantly more favorable view of the usability of their software. It makes sense that, as participants through hands-on experience become more proficient at manipulating the software, they come to consider it easier to use. Interestingly, the icon–label interface was always considered easier to use than either of the other interfaces, including the label-only interface. This is a case in which perceptions do not match performance, since the label-only group inconsistently performed as well as the icon-label group. It appears that, as argued by Davis *et al.* (1989), an interface designed with icons suggests ease of use to the learner and is rated easier to use partially independent of performance. This is suggested by the higher ratings for the icon-label interface *vis-à-vis* the label-only interface in this experiment. However, if this is the case, there are certainly other factors involved, as well. This is indicated by the fact that the icon-only interface, which used exactly the same icons, was rated significantly harder to use than the icon–label interface. Our interpretation is that the superiority of the icon–label interface on the PEU comes from two different sources. First, in the case of its superiority over the label-only interface, it is attributable to the surface look of the interface created by the use of icons. Second, in the case of its superiority over the icon-only interface, it is attributable to the greater success in performance in the icon–label condition. If this is a correct interpretation, it appears that in the earliest stage of learning users' perceptions may be formed both by the surface look at the interface, including preconceptions attached to a certain look, *and* by the result of hands-on experience with the interface. Davis's model posits that factors such as the look of the interface may play a role in the formation of perceptions, but it does not explicitly factor in the effect of the initial experience in learning the interface. This could be done in future research by first measuring the PEU after the learner has seen a demonstration of the system and again after the initial hands-on session.

Perceived usefulness followed a different pattern. Theory argues that PU is based on two factors: 1) the functionality of the software and users' judgements of how much that functionality will aid them in their work; and 2) perceived ease of use, since a tool that is difficult to use will require more effort on the part of the user, which correspondingly reduces it usefulness. In session one, the PU of the label-only interface was poorer than that of the other two interfaces. Since the functionality was identical, the difference should be explained by the influence of PEU on PU. This makes sense in interpreting the label-only vs. icon–label difference, since the icon–label version did have a significantly more positive PEU. However, this does not satisfactorily explain why the PU of the icon-only group equalled that of the icon–label group, given that the PEU of the icon-only group was poorer. Further experimental work is needed to replicate and explain this result. For the moment, we can only say that other unidentified factors in addition to functionality and PEU may influence the formation of PU. In subsequent use in session two, the PU of the label-only interface and the icon–label interface remained nearly static. However, the PU of the icon-only interface improved substantially. An explanation of this pattern of results might be that performance on the label-only and icon–label interfaces was considerably better in terms of correctness, time, and help use in the initial session. These two groups had an easier time getting the interface to do what they wanted in session one and, as a result, probably gained a better understanding of system functionality initially than did the icon-only group. So their perceptions of usefulness were formed at that early point and remained fairly stable as they gained more incremental skill in session two. Their small and non-significant improvement in PU, in spite of a significant improvement in PEU, suggests that PEU is less improtant to the formation of PU than are perceptions about the value of system functionality. On the other hand, performance on the icon-only interface in terms of correctness, time, and help was poorer in session one but approached that of the other two groups in session two. Thus, in session two the icon-only group had much less trouble using the interface than previously. Their difficulties in session one may have both affected their perceptions of ease of use *and also* obscured their understanding of the system's functionality, both of which could have affected PU. Their improved PU in session two probably represents the joint effect of an improved PEU and of a better understanding of system functionality and how it could be useful to them. We tend to attribute it mostly to better understanding of system functionality, since the other two interfaces also improved in PEU without experiencing a large improvement in PU.

7. Conclusions

The results of this study lead to the conclusion that learners of application programs are aided in initial learning by the use of either icons with text labels or text labels alone. Icons lacking labels performed very poorly in the early stages of learning, although learners of the icon-only interface caught up with the performance of the other two groups by the end of the first session. While the importance of the text labels to early learning might lead to a recommendation to eliminate icons and simply use text labels on buttons, such a recommendation may not be practical or desirable. In terms of practicality, a graphical user interface with icons has become an important element in marketing personal computers, providing a sort of producer 'signature'. Also, text labels or icons incorporating labels can be more space consuming than icons alone. In terms of desirability, the evidence from this study suggests that learners have less favorable perceptions of the usability and usefulness of text-only interfaces. While learners' perceptions may have little effect in a situation of captive use, in discretionary use they are a determining factor in the choice to use a certain software package.

If, as we argue, the use of icons in application programs is likely to continue to be more popular than the use of text labels, the introduction of users to iconic software should be planned to aid them in mastering the icons. Learners who have little knowledge of the functionality of the application program and who have little knowledge of representational conventions often used in iconic interfaces are most in need of aid. These users have little basis for making inferences about the meaning of icons in the lack of text labels. On the basis of our results we recommend the provision of text labels in initial use of iconic interfaces. However, the value of labels is quite short-lived and retention problems are not severe over intervals up to one week, according to our results. This suggests that the text labels play an important role briefly in the early stage of learning, but after that they lose their value. From this, it appears that labels may be suppressed after the users have worked with the program for a relatively short time. Since new users of software today are very likely to learn the software on their own, it may be advisable for the software to be designed to initially show labels, with an easy to locate option to turn them off. Alternatively, online training tutorials could direct the new users to turn on fixed or pop up text labels and show the new user how to turn them off when desired.

Another conclusion is that the interface type strongly affects user perceptions. A strong connection has been shown between perceptions of software and its adoption and use in the case of discretionary use (Davis 1989). Thus, the factors that lead to the formation of perceptions are important. In this study, in which the functionality of the interfaces was identical, it appears that perceptions of ease of use were affected both by the participants' performance and by the surface look of the interface.

Participants using the label-only interface formed poorer perceptions of the interface than their performance would warrant. For both perceived ease of use and perceived usefulness, there was an indication of evolution of perceptions over time, especially in interfaces that were initially more difficult to use. Furthermore, it appeared that the effect of PEU on the formation and evolution of PU was small and that other unidentified factors may be involved beyond those predicted by theory. While a number of studies were carried out in developing and validating the Technology acceptance model of Davis (Davis 1989, Davis *et al*. 1989), few empirical studies have attempted to vary the form of the interface in a controlled setting and measure the effects on perceptions and on performance. The asymmetry between the results of perception and performance and the nature of the evolution of perceptions over time in this study suggest that further research on the effects of various interface factors on perceptions is warranted. The importance of further study of the effect of interface features on perceptions is emphasized by the fact that in our study performance of the different groups converged by the end of the experiment, but perceptions did not.

References

Adams, D. A., Nelson, R. R. and Todd, P. A. 1992, Perceived usefulness, ease of use, and usage of information technology: A replication. *MIS Quarterly,* **15,** 227–247.

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

Ausubel, D. P. 1968, *Educational Psychology: A Cognitive View.* (NY: Holt, Reinhart and Winston).

Benbasat, I. and Todd, P. 1993, An experimental investigation of interface design alternatives: icon vs. text and direct manipulation vs. menus. *International Journal of Man–Machine Studies,* **38,** 369–402.

Bewley, W. L., Roberts, J. L., Schroit, D. and Verplank, W. L., 1983, Human factors testing in the design of Xerox's 8010 'Star' office workstation. *CHI '83 Conference Proceedings.* (NY: ACM), 72–77.

Blankenberger, S. and Hahn, K. 1991, Effects of icon design on human–computer interaction. *International Journal of Man–Machine Studies,* **35,** 363–377.

Blattner, M. A., Sumikawa, D. A. and Greenberg, R. A. 1989, Earcons and icons: their structure and common design principles. *Human–Computer Interaction,* **4**(1), 11–44.

Byrne, M. D. 1993, Using icons to find documents: simplicity is crucial. *INTERCHI'93 Conference Proceedings.* (New York: ACM) 446–463.

Carroll, J. M. and Mack, R. L. 1986, Metaphor, computing systems, and active learning. *International Journal of Man–Machine Studies,* **22,** 39–57

Davis, F. D. 1989, Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly, **13**(3), 319–340.

Davis, F. D., Bagozzi, R. P. and Warshaw, P. R. 1989, User acceptance of computer technology: a comparison of two theoretical methods. *Management Science,* **35**(8), 982–1003.

Davis, S. A. and Bostrom, R. P. 1993, Training end users: an experimental investigation of the roles of the computer interface and training methods. *MIS Quarterly,* **17,** 61–85.

Davis, S. A. and Wiedenbeck, S. 1998, The Effect of Interaction Style and Training Method on End User Learning of Software Packages. *Interacting with Computers.* **11**(2), 147 – 172.

Egido, C. and Patterson, J. 1988, Pictures and category labels as navigational aids for catalog browsing. *CHI'88 Conference Proceedings.* (New York: ACM), 127–132.

Familant, M. E. and Detweiler, M. C. 1993, Iconic reference: evolving perspectives and an organizing framework. *International Journal of Man–Machine Studies,* **39,** 705–728.

Gittens, D. 1986, Icon-based human–computer interaction. *Interaction Journal of Man–Machine Studies,* **24,** 519–543.

Green, A. J. K. and Barnard, P. J. 1990, Iconin interfacing: the role of icon distinctiveness and fixed or variable screen location. *INTERACT '90 Proceedings.*(Amsterdam: North-Holland), 457–462.

Horton, W. 1994, *The Icon Book: Visual Symbols for Computing Systems and Documentation.* (New York: Wiley).

Hutchins, E. L., Hollan, J. D. and Norman, D. A. 1985, Direct manipulation interfaces. in D. A. Norman and S. Draper (eds), *User-centered System Design.* (Hillsdale, NJ: Lawrence Erlbaum), 87–124.

Johnson, J., Roberts, T. L., Verplank, W., Smith, D. C., Irby, C. H. and Beard, M. 1989, The Xerox Star: a retrospective. *IEEE Computer,* **22**(9), 11–26.

Lansdale, M. W. 1988, On the memorability of icons in an information retrieval task. *Behaviour and Information Technology,* **7,** 131–151.

Larkin, J. H. and Simon, H. A. 1987, Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science,* **11,** 65–99.

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

Navon, D. 1984, Resources: a theorectical soupstore?*Psychological Review,* **91,** 216–234.

Nolan, P. R. 1989, Designing screen icons: ranking and matching studies. *Proceedings of the Human Factors Society 33rd Annual Meeting.* (Santa Monica, CA: The Human Factors Society), 380–384.

Rogers, Y. 1989, Icons at the interface: their usefulness. *Interacting With Computers,* **1,** 105–117.

Rohr, G. and Keppel, E. 1984, Iconic interfaces: where to use and how to construct? in H. W. Hendrick, and O. Brown (eds), *Human Factors in Organizational Design and Management* (New York: Elsevier), 269 – 275.

Shepard, R. N. 1967, Recognition memory for words, sentences and pictures. *Journal of Verbal Learning and Verbal Behavior,* **6,** 156–163.

Shneiderman, B. 1997, *Designing the User Interface: Strategies for Effective Human–Computer Interaction,* 3rd ed. (Reading, MA: Addison-Wesley).

Smith, D. C., Irby, C., Kimball, R., Verplank, W. and Harslem, E. 1982, Designing the Star user interface. *Byte,* **7**(4), 242–282.

Standing, L. 1973, Learning 10,000 pictures. *Quarterly Journal of Experimental Psychology,* **25**(2), 207–222.

Wiedenbeck, S. and Davis, S. 1997, The influence of interaction style and experience on user perceptions of software packages. *International Journal of Human–Computer Studies,* **46,** 563–588.