Natural Dialogue in Modes Other Than Natural Language

Robert J.K. Jacob

Human-Computer Interaction Lab
Naval Research Laboratory
Washington, D.C., U.S.A.

ABSTRACT

Each command or transaction in a modern graphical user interface exists as a nearly independent utterance, unconnected to previous and future ones from the same user. This is unlike real human communication, where each utterance draws on previous ones for its meaning. While some natural language human-computer interfaces attempt to include such characteristics of human dialogue, they have been notably absent from graphical user interfaces. Our goal is to connect these properties of human dialogue to direct manipulation or graphical styles of user-computer interaction. This chapter describes our approach to incorporating some of the properties of natural dialogue (such as conversational flow, discourse, focus) into "other" (that is, not natural language) modes of human-computer communication. It also describes our research on building a natural user-computer dialogue based on a user's eye movements. That work shows that starting from natural, rather than trained, eye movements results in a more natural dialogue with the user.

Introduction

In a direct manipulation or graphical interface, each command or brief transaction exists as a nearly independent utterance, unconnected to previous and future ones from the same user. Real human communication rarely consists of such individual, unconnected utterances, but rather each utterance can draw on previous ones for its meaning. It may do so implicitly, embodied in a conversational focus, state, or mode, or explicitly (''Do the same sorting operation you did before, but on these new data'').

Our goal is to connect these properties of human dialogue to direct manipulation or graphical interaction styles. While some natural language human-computer interfaces attempt to exploit these characteristics of human dialogue, they have been notably absent from graphical interfaces. Some of the properties of dialogue seem tightly connected to natural language and hence may not apply to a graphical interface, but some reflect deeper structure of human discourse than can usefully be applied to other modes of communication, such as graphics, pointing, dragging, gesturing, looking, and form fill-in. Natural dialogue is by no means restricted to natural language. Most research on the processes needed to conduct such dialogues has concentrated on natural language, but some of them can be applied to any human-computer dialogue conducted in any language. A direct manipulation dialogue is conducted in a rich graphical language using powerful and natural input and output modalities. The user's side of the dialogue may consist almost entirely of pointing, gesturing, and pressing buttons, and the computer's, of animated pictorial analogues of real-world objects. A dialogue in such a

language could nevertheless exhibit useful dialogue properties, such as following focus.

Direct Manipulation

Direct manipulation graphical interfaces provide many significant advantages in human-computer communication (Shneiderman, 1983). They exploit the high bandwidth of the human visual system through the use of graphics to communicate information. They provide inter-referential input-output, which allows the user to communicate back to the computer in the same graphical mode by referring to objects on display. A direct manipulation user interface typically presents its user a set of objects on a display and a standard repertoire of manipulations that can be performed on any of them. This means that the user has no command language to remember beyond the standard set of manipulations and few cognitive changes of mode. The displayed objects are active in the sense that they are affected by each command issued. Whenever the user requests output, the objects shown in the resulting display on the screen are acceptable inputs to subsequent commands. They provide a continuous, implicit reminder of the available objects and their states. Output is thus not a fixed, passive display, but a collection of dynamic, manipulable objects, each usable as an input to subsequent commands. A typical user command or input is synthesized from objects already on display, the outputs of previous commands.

Recent work has carried the user's illusion of manipulating real objects still further. By coupling a the motion of the user's head to changes in the images presented on a head-mounted display, the illusion of being surrounded by a world of computer-generated images or a virtual environment is created. Hand-mounted sensors allow the user to interact with these images as if they were real objects located in space surrounding him or her (Foley, 1987). Much of the advantage of such an interfaces derives from the fact that the user seems to operate directly *on* the objects in the computer rather than carrying on a dialogue *about* them. Instead of using a command language to describe operations on objects, the user ''manipulates'' objects visible to him or her.

In cognitive terms, the properties of direct manipulation interfaces can be decomposed into direct engagement and reduced semantic distance (Hutchins, 1986). Direct engagement is the sense of manipulating objects directly on a screen rather than conversing *about* them. ''There is a feeling of involvement directly with a world of objects rather than of communicating with an intermediary. The interactions are much like interacting with objects in the physical world. Actions apply to the objects, observations are made directly upon those objects, and the interface and the computer become invisible.'' (Hutchins, 1986) The other property is a reduction of cognitive distance, the mental effort needed to translate from the input actions and output representations to the operations and objects in the problem domain itself. Using a display screen, the visual images chosen to depict the objects of the problem or application domain should be easy for the user to translate to and from that domain. Conversely, for input, the actions required to effect a command should be closely related to the meaning of the command in the problem domain. Research suggests that these two factors each contribute separably and additively to the higher performance observed with direct manipulation interfaces (Ballas, 1992).

Direct manipulation interfaces are thought to be ''modeless,'' in contrast to other interface styles. Modes or states refer to the varying interpretation of a user's input. In each mode, an interface may give different meanings to the same input operations. In a modeless interface, the system is always in the same mode, and inputs always have the same interpretation. Direct manipulation user interfaces appear to be modeless, because many objects are visible on the screen and at any time the user can apply any of a standard set of commands to any object. In fact, this view ignores the input operation of designating the object of interest. That operation

sets the mode. For example, moving the cursor to lie over an object *is* the command to cause a mode change, because once it is moved, the range of acceptable inputs is reduced and the meaning of each of those inputs is determined. The benefits of ‘‘modelessness’’ come from the fact that the mode is always clearly visible (as the location of a cursor in this example), and it has an obvious representation (simply the echo of the same cursor location just used to enter the mode change command) (Jacob, 1986).

Direct manipulation interfaces, including those set within virtual environments, then, provide a rich and powerful style of interaction, well tuned to both the perceptual and the cognitive characteristics of the user. However, these advantages come to an abrupt end when considering interaction over more than a single transaction. In a typical direct manipulation interface, every user operation is an isolated one-shot transaction with the computer, unconnected to previous or future interactions. Perhaps the goal of ‘‘modeless’’ operation has been carried too far.

A Framework for Human-Computer Dialogue

Human-computer interface design is typically decomposed into the semantic, syntactic, and lexical levels (Foley, 1990):

- The semantic level describes the functions performed by the system. This corresponds to a description of the functional requirements of the system, but it does not address how the user will invoke the functions. The semantic level defines ‘‘meanings,’’ rather than ‘‘forms’’ or ‘‘sequences,’’ which are left to the lower levels. It provides the high-level model or abstraction of the functions of the system.
- The syntactic level describes the sequences of inputs and outputs necessary to invoke the functions described. That is, it gives the rules by which which sequences of words (‘‘tokens’’) in the language are formed into proper (but not necessarily semantically meaningful) sentences. The design of the syntactic level describes the sequence of the logical input, output, and semantic operations, but not their internal details. A logical input or output operation is an input or output token. Its internal structure is described at the lexical level, while the syntactic describes when the user may enter it and what will happen next if he or she does (for an input token) or when the system will produce it (for an output token).
- The lexical level determines how the inputs and outputs are actually formed from primitive hardware operations or lexemes. It represents the binding of hardware actions to the hardware-independent tokens of the input and output languages. While tokens are the smallest units of meaning with respect to the syntax of the dialogue, lexemes are the actual hardware input and output operations that comprise the tokens.

Extending the linguistic analogy, we add a higher level to the interface design, above the semantic level:

- The discourse level is concerned with the flow of the human-computer dialogue over the course of more than one transaction. The semantic, syntactic, and lexical levels are concerned with a single user-computer transaction or brief interaction. The discourse level introduces elements that relate one transaction to another, such as dialogue focus.

Synthesis

We hope to bring some of the higher-level dialogue properties of natural language to the direct manipulation interaction style by adding the discourse level to the interface design. Previous work on graphical interaction techniques has been restricted to single transactions and

single modes. Dialogue includes phenomena that happen over a series of transactions and integrates actions that occur in several modes. For example, a precise meaning can often be gleaned by combining imprecise actions in several modes, each of which would be ambiguous in isolation. We thus attempt to broaden the notion of interaction techniques in these two dimensions (multiple transactions and multiple modes). The rest of this paper describes, first, our ideas about how these areas can be melded and, second, our results results to date in obtaining single-transaction natural human-computer communication using eye movements as input.

Natural Dialogue in a Direct Manipulation Interface

We will begin with simple examples that allow movement from the utterly isolated-command-at-a-time situation in current graphical user interfaces. First, the notion of a single, prominently highlighted currently selected object (CSO), to which commands are applied, can be viewed as one simple step along these lines. It was incorporated into the first desktop interface, the Xerox Star, and nearly all subsequent direct manipulation interfaces. A further simple example would be for the computer to know where the user is looking and interpret his or her command in the light of that information. We will describe our current work in that area below. More interesting cases involve examining the recent history of the user's behavior—what objects he has referred to, looked at, and manipulated over a longer time scale. We will describe our embryonic work in that area.

One useful property of dialogue that can be applied to a graphical interface is focus (Grosz, 1978). The graphical user interface could keep a history of the user's current focus, tracking brief digressions, meta-conversations, major topic shifts, and other changes in focus. Unlike a linguistic interface, the graphical interface would use inputs from a combination of graphical or manipulative modes to determine focus. Pointing and dragging of displayed objects, user gestures and gazes as well as the objects of explicit queries or commands all provide input to determine and track focus (Perez, 1993). Moreover, focus would not be maintained as a single object, but rather a history of the course of the user-computer dialogue. It is necessary to track excursions or digressions in the dialogue so focus can be restored as necessary. In addition, it is helpful to track focus by categories. This allows the user to refer to ''the ship'' even though the current focus is another object. In that case, the recent history of focus would be searched to find a ship of the appropriate category. Finally, focus is not necessarily a concrete object; it may be a class or category of objects (''all blue ships'') or a more abstract entity (''the previous command'').

As a simple example of the use of such focus information, the user might give a command (verb) without specifying its object, and the interface would supply the object based on the user's current focus. A more sophisticated approach would deduce the object of the command based on the recent history of the user's focus, rather than its single latest manifestation. The nature of the command might constrain the possible objects. For example, ''display hull speed'' might apply only to ships. If the current focus were not such a ship, the interface would backtrack through recent focus objects to find the last applicable ship and use it as the inferred object of the command. Further, a ''retrieve data'' command might indicate a shift from a digression back to the main dialogue, hence the appropriate object of this command would be not the current (digression) focus but the previous (main dialogue) focus.

Another use of focus is not based on supplying data to complete explicit commands, but rather passively deducing the user's current area of interest. The computer could, unasked, simply provide more detail about such an area. If the system had limited resources for obtaining data, it might even use this information to deploy its resources preferentially. For example, it might concentrate its display rendering computations in a particular area of the display or

concentrate scarce communication bandwidth on the data that lie in that area. Extending outward from the computer itself, it might even order a change in the targeting of its external measuring instruments or sensors to the area of interest or initiate background tasks to collect additional data in that area.

Human dialogue often combines inputs from several modes. Deixis often involves a pointing gesture that does not precisely specify its object; the listener deduces the correct object from the context of the dialogue and, possibly, from integrating information from the hand gesture, the direction of the user's head, tone of his or her voice, and the like (Hill, 1991). A user could, similarly, give a command and point in a general direction to indicate its object. The interface would disambiguate the pointing gesture based on the recent history of its dialogue with the user and, possibly, by combining other information about the user from physical sensors. An imprecise pointing gesture in the general direction of a displayed region of a map could be combined with the knowledge that the user's recent commands within that region referred principally to one of three specific locations (say, river *R*, island *I*, and hill *H*) within the region and the knowledge that the user had previously been looking primarily at islands displayed all over the map. By combining these three imprecise inputs, the interface could narrow the choice down so that (in this example) island *I* is the most likely object of the user's new command.

This example combined inputs in several modes and interaction history to disambiguate an imprecise pointing gesture. The same approach applies in the absence of a pointing gesture. The user might simply ask for ''that'' without pointing. Recent history and focus plus physical information about the user may still be adequate to disambiguate the referent of ''that.''

The problem is usefully constrained if the user asks for ''the aircraft carrier'' rather than simply ''that.'' History, focus, and other information may then be combined with the restriction that aircraft carriers are the only pertinent objects to search for the last-referenced aircraft carrier (rather than the last-focused object in general) and thereby determine unambiguously the correct object of the user's command.

Another quite different use of dialogue in a graphical interface is to maintain a history of what the user has or has not processed. As a crude example, when new and urgent information appears on the display, it may be necessary to alert the user with a tone or a mandatory confirmation—but only if he or she has not already seen or acknowledged the information. Otherwise, in a critical situation warning tones may be sounding continually, often for information the user has processed, and the user will thus begin to ignore (or disable) the tones or become used to performing the mandatory confirmation action automatically. More generally, knowing whether the user has seen a particular object recently can often be helpful in providing the context in which to interpret the next action or command. Some cognitive user models attempt to keep track of what the user knows or does not know at any point in time in a dialogue. The present approach relies instead on explicit physical user actions (pointing to an object, seeing it, querying it) to determine what the user is aware of. This is somewhat more crude, but gives quite positive indication of knowledge the user does *not* have. Specifically, with this physically-based approach the interface can determine positively that the user has *not* seen something or has not seen it in a given time period. It can also determine that he or she *has* seen it, but it cannot be positive that he or she understood or remembered it.

Natural Eye Movement

This section reports our work to date on using eye movements as a medium for human-computer communication, in single isolated transactions (Jacob, 1993, Jacob, 1991, Jacob, 1990). This work differs from much other work on eye movements for input in that it emphasizes the use of natural eye movements rather than trained ones. Our approach has been,

wherever possible, to obtain information from the *natural* movements of the user's eye while viewing a display, rather than requiring the user to make specific *trained* eye movements to actuate the system. To date, we have implemented only single transactions, that is, the human-computer interface has no longer-term dialogue or discourse properties. We are currently proceeding toward these longer-term properties, by beginning with the methods outlined in the previous section.

We began this work because HCI technology is currently stronger in the computer-to-user direction than user-to-computer, hence today's user-computer dialogues are typically one-sided, with the bandwidth from the computer to the user far greater than that from user to computer. Using the movements of a user's eyes as input to the computer can provide an additional high-bandwidth channel for obtaining data from the user conveniently and rapidly. Our focus has not been on technology for measuring a user's eye movements, but rather on developing appropriate interaction techniques that incorporate eye movements into the user-computer dialogue in a convenient and natural way. We thus begin by studying the known characteristics of natural eye movements and then attempt to design dialogues based on these characteristics.

Classes of Eye Movement-based Interaction

In eye movements as with other areas of user interface design, it is helpful to draw analogies that use people's already-existing skills for operating in the natural environment and then apply them to communicating with a computer. One of the reasons for the success of direct manipulation interfaces is that they draw on analogies to existing human skills (pointing, grabbing, moving objects in physical space), rather than trained behaviors; and virtual environment offer the promise of usefully exploiting people's existing physical navigation and manipulation abilities. These notions are more difficult to extend to eye movement-based interaction, since few objects in the real world respond to people's eye movements (except for other people). In describing eye movement-based human-computer interaction we can draw two distinctions, one in the nature of the user's eye movements and the other, in the nature of the responses. Each of these could be viewed as *natural* (that is, based on a corresponding real-world analogy) or *unnatural* (no real world counterpart):

- *User's eye movements:* Within the world created by an eye movement-based interface, users could move their eyes to scan the scene, just as they would a real world scene, unaffected by the presence of eye tracking equipment (i.e., *natural* eye movement). The alternative is to instruct users of the eye movement-based interface to move their eyes in particular ways, not necessarily those they would have employed if left to their own devices, in order to actuate the system (i.e., *unnatural* or learned eye movements).
- *Nature of the response:* Objects could respond to a user's eye movements in a natural way, that is, the object responds to the user's looking in the same way real objects do. As noted, there is a limited domain from which to draw such analogies in the real world. The alternative is unnatural response, where objects respond in ways not experienced in the real world.

This suggests a range of four possible styles of eye movement-based interaction:

- *Natural eye movement/Natural response:* This area is a difficult one, because it draws on a limited and subtle domain, principally how people respond to other people's gaze. Starker and Bolt provide an excellent example of this mode, drawing on the analogy of a tour guide or host who estimates the visitor's interests by his or her gazes (Starker, 1990).

- *Natural eye movement/Un-natural response:* In the work described below, we try to use natural (not trained) eye movements as input, but we provide responses unlike those in the real world. This is a compromise between full analogy to the real world and an entirely artificial interface. We present a display and allow the user to observe it with his or her normal scanning mechanisms, but such scans then induce responses from the computer not normally exhibited by real world objects.
- *Un-natural eye movement/Un-natural response:* Most previous eye movement-based systems have used learned (''unnatural'') eye movements for operation and thus, of necessity, unnatural responses. Much of that work has been aimed at disabled or hands-busy applications, where the cost of learning the required eye movements (''stare at this icon to activate the device'') is repaid by the acquisition of an otherwise impossible new ability. However, we believe that the real benefits of eye movement interaction for the majority of users will be in its naturalness, fluidity, low cognitive load, and almost unconscious operation; these benefits are attenuated if unnatural, and thus quite conscious, eye movements are required.
- *Un-natural eye movement/Natural response:* The remaining category created by this taxonomy is anomalous and not seen in practice.

Interaction Techniques based on Natural Eye Movements

In order to work toward natural eye movement-based interaction, we began by studying the nature of human eye movements and attempted to develop interaction techniques around their characteristics. To see an object clearly, it is necessary to move the eyeball so that the object appears on the fovea, a small area at the center of the retina. Because of this, a person's eye position provides a rather good indication (to within the one-degree width of the fovea) of what specific portion of the scene before him of her is being examined. The most common way of moving the eyes is a sudden, ballistic, and nearly instantaneous saccade. It is typically followed by a fixation, a 200-600 ms. period of relative stability during which an object can be viewed. During a fixation, however, the eye still makes small, jittery motions, generally covering less than one degree. Smooth eye motions, less sudden than saccades, occur only in response to a moving object in the visual field. Other eye movements, such as nystagmus, vergence, and torsional rotation are relatively insignificant in a user-computer dialogue.

The overall picture of eye movements for a user sitting in front of a computer is a collection of steady (but slightly jittery) fixations connected by sudden, rapid saccades. The eyes are rarely entirely still. They move during a fixation, and they seldom remain in one fixation for long. Compared to the slow and deliberate way people operate a mouse or other manual input device, eye movements careen madly about the screen. During a fixation, a user generally thinks he or she is looking steadily at a single object—he or she is not consciously aware of the small, jittery motions. This suggests that the human-computer dialogue should be constructed so that it, too, ignores those motions, since, ultimately, it should correspond to what the user *thinks* he or she is doing, rather than what the eye muscles are actually doing. The most naive approach to using eye position as an input might be to use it as a direct substitute for a mouse: changes in the user's line of gaze would directly cause the mouse cursor to move. This turns out to be an unworkable (and annoying) design for two main reasons. The first is in the eye itself, the jerky way it moves and the fact that it rarely sits still, even when its owner thinks he or she is looking steadily at a single object; and the second is the instability of available eye tracking hardware.

Moreover, people are not accustomed to operating devices just by moving their eyes in the natural world. They expect to be able to look at an item without having the look ''mean'' something. Normal visual perception requires that the eyes move about, scanning the scene

before them. It is not desirable for each such move to initiate a computer command. At first, it is empowering simply to look at what you want and have it happen. Before long, though, it becomes like the Midas Touch. Everywhere you look, another command is activated; you cannot look anywhere without issuing a command. The challenge in building a natural eye movement interface is to avoid this Midas Touch problem. Ideally, a natural interface should act on the user's eye input when he wants it to and let him just look around when that's what he wants, but the two cases are impossible to distinguish in general. Instead, we investigate interaction techniques that address this problem in specific cases.

Finally, our goal is to provide interactions that are faster, more convenient, and more natural than competing alternatives such as gesture, key presses, and other conventional input media. This is in contrast to much previous work using eye movements as an input medium, which has focused on constrained situations such as disabled users or users whose hands are otherwise occupied (e.g., pilots). Since the other input media are typically unavailable to such users, any practical eye movement interface is an improvement; and requiring careful, deliberate, slow, or trained eye movements from the user is still workable. We seek to satisfy a more rigorous standard of comparison: the user's hands are available for input, but our eye movement-based interaction technique should be faster or more natural than using the hands.

Feedback

In an eye movement-based user-computer dialogue an interesting question arises with respect to feedback, which is an important component of any dialogue: Should the system provide a screen cursor that follows the user's eye position (as is done for mice and other conventional devices)? This would provide feedback at the lexical level, as does the mouse cursor or the echoing of keyboard characters.

However, if the eye tracker were perfect, the image of such a cursor would occupy a precisely stationary position on the user's retina. An image that is artificially fixed on the retina (every time the eye moves, the target immediately moves precisely the same amount) will appear to fade from view after a few seconds (Pritchard, 1961). The large and small motions the eye normally makes prevent this fading from occurring outside the laboratory, and few eye trackers can track small, high-frequency motions rapidly or precisely enough for this to be a problem, but it does illustrate the subtlety of the design issues.

A more immediate problem is that an eye-following cursor will tend to move around and thus attract the user's attention. Yet it is perhaps the *least* informative aspect of the display (since it tells you where you are already looking). Further, if there is any systematic calibration error, the cursor will be slightly offset from where the user is actually looking, causing the user's eye to be drawn to the cursor, which will further displace the cursor, creating a positive feedback loop. This is indeed a practical problem, and we often observe it.

Finally, if the calibration and response speed of the eye tracker were perfect, feedback would not be necessary, since a person knows exactly where he or she is looking (unlike the situation with a mouse cursor, which helps one visualize the relationship between mouse positions and points on the screen).

Experience with Interaction Techniques

An interaction technique is a way of using a physical input device to perform a generic task in a human-computer dialogue (Foley, 1990). It is an abstraction of some common class of interactive task, for example, choosing one of several objects shown on a display screen. We have developed natural eye movement-based interaction techniques for performing some basic operations in direct manipulation systems, such as selecting and moving objects.

Selecting an object among several displayed on the screen is customarily done with a mouse, by pointing at the object and then pressing a button. With the eye tracker, there is no natural counterpart of the button press. We reject using a blink for a signal because it detracts from our goal of natural interaction by requiring the user to think about when to blink. We examined two alternatives. In one, the user looks at the desired object then presses a button on a keypad to indicate that the looked-at object is his choice. In the second, the user must continue to look at the object for a dwell time, after which it is selected without further operations. In practice, however, the dwell time approach—with a short dwell time—is preferable in meeting our goal of natural interaction. While a long dwell time does ensure that an inadvertent selection will not be made by simply ''looking around'' on the display, we found it unpleasant to use, probably because it does not exploit natural eye movements (people do not normally fixate one spot for that long). Short dwell time selection requires that it be possible trivially to undo the selection of a wrong object. For example, if selecting an object causes a display of information about that object to appear and the information display can be changed instantaneously, then the effect of selecting wrong objects is immediately undone as long as the user eventually reaches the right one. This approach, using a 150-250 ms. dwell time gives excellent results. The lag between eye movement and system response (required to reach the dwell time) is hardly detectable to the user, yet long enough to accumulate sufficient data for our fixation recognition and processing. The subjective feeling is of a highly responsive system, almost as though the system is executing the user's intentions before he expresses them. For situations where selecting an object is more difficult to undo, button confirmation is used.

For moving an object on a display, we segregated the two functions usually performed by the mouse—selecting an object to be manipulated and performing the manipulation. We experimented with using eye position for the selection task and hand input for the move itself. The eye selection is made as described above; then, the user grabs the mouse, presses a button, drags the mouse in the direction the object is to be moved, and releases the button. We also experimented with using the eye to select *and* drag the object and a pushbutton to pick it up and put it down. With that approach, the user selects the object, then presses a button; while the button is depressed, the object drags along with the definite fixations (not raw movements) of the user's eye. The effect is that the object actually jumps to each fixation after about 100 ms. and then remains steadily there—despite actual eye jitter—until the next fixation. At first, we thought the second method would be unnatural and difficult to use, because eye movements would be better for selecting an object than picking it up and dragging it around. This was not borne out. While the eye-to-select/mouse-to-drag method worked well, the user was quickly spoiled by the eye-only method. Once you begin to expect the system to know where you are looking, the mouse-to-drag operation seems awkward and slow. After looking at the desired object and pressing the ''pick up'' button, the natural thing to do is to look at where you are planning to move the object. At this point, you feel, ''I'm looking right at the destination I want, why do I now have to go get the mouse to drag the object over here?'' With eye movements processed to suppress jitter and respond only to recognized fixations, the motion of the dragging object is reasonably smooth and predictable and yet appears subjectively instantaneous. It works best when the destination of the move is a recognizable feature on the screen rather than an arbitrary blank spot. If that is a problem, it can be solved by temporarily displaying a grid pattern during dragging.

Another interaction technique was developed for a scrolling window of text, in which not all of the material to be displayed can fit. We present arrows below the last line of the text and above the first line, indicating that there is additional material not shown. If the user looks at an arrow, the text itself starts to scroll. However, the text never scrolls when the user is actually reading it (rather than looking at the arrow). The assumption is that, as soon as the text starts scrolling, the user's eye will be drawn to the moving display and away from the

arrow, which will stop the scrolling. The user can thus read down to end of the window, then, after he or she finishes reading the last line, look slightly below it, at the arrow, in order to retrieve the next part of the text.

Since pop-up menus inherently assume a button, we experimented with an eye-operated pull-down menu. If the user looks at the header of a pull-down menu for a brief dwell time, the body of the menu will appear on the screen. Next, he or she can look at the items shown on the menu. After a brief look at an item (100 ms.), it will be highlighted, but its command will not yet be executed. This allows the user time to examine the different items on the menu. If the user looks at one item for a much longer time (1 sec.) or presses a button at any time, the highlighted item will be executed and the menu erased. As with the long dwell time object selection, here, too, the button is more convenient than the long dwell time for executing a menu command. This is because the dwell time necessary before executing a command must be kept quite high, at least noticeably longer than the time required to read an unfamiliar item. This is longer than people normally fixate on one spot, so selecting such an item requires an unnatural sort of ''stare.'' Pulling the menu down and selecting an item to be highlighted can both be done effectively with short dwell times.

Another appropriate use of eye movements is to designate the active window in a window system. Current systems use an explicit mouse command to designate the active window. Instead, we use eye position—the active window is simply the one the user is looking at. A delay is built into the system, so that user can look briefly at other windows without changing the listener window designation. Fine cursor motions within a window are still handled with the mouse, which gives an appropriate partition of tasks between eye tracker and mouse, analogous to that between speech and mouse used by Schmandt, Ackerman, and Hindus (Schmandt, 1990).

Acknowledgments

I want to thank my colleagues in the Dialogue Research Program at NRL and its leader, Helen Gigley, for helpful debates and discussions on these issues. I thank Robert Carter, Connie Heitmeyer, Preston Mullen, and Linda Sibert for much help on the eye movement work reported herein, and Linda Sibert for fruitful collaboration on all of this research. This work was sponsored by the Office of Naval Research.

References

Ballas,.
J.A. Ballas, C.L. Heitmeyer, and M.A. Perez, ''Evaluating Two Aspects of Direct Manipulation in Advanced Cockpits,'' *Proc. ACM CHI'92 Human Factors in Computing Systems Conference*, pp. 127-134, Addison-Wesley/ACM Press, 1992.

Foley,.
J.D. Foley, ''Interfaces for Advanced Computing,'' *Scientific American*, vol. 257, no. 4, pp. 127-135, October 1987.

Foley,.
J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley, Reading, Mass., 1990.

Grosz,.
B.J. Grosz, ''Discourse,'' in *Understanding Spoken Language*, ed. by D.E. Walker, pp. 229-284, Elsevier North-Holland, New York, 1978.

Hill,. W.C. Hill and J.D. Hollan, ''Deixis and the Future of Visualization Excellence,'' *Proc. IEEE Visualization'91 Conference*, pp. 314-319, IEEE Computer Society Press, 1991.

Hutchins,.
E.L. Hutchins, J.D. Hollan, and D.A. Norman, ‘‘Direct Manipulation Interfaces,’’ in *User Centered System Design: New Perspectives on Human-computer Interaction*, ed. by D.A. Norman and S.W. Draper, pp. 87-124, Lawrence Erlbaum, Hillsdale, N.J., 1986.

Jacob,.R.J.K. Jacob, ‘‘A Specification Language for Direct Manipulation User Interfaces,’’ *ACM Transactions on Graphics*, vol. 5, no. 4, pp. 283-317, 1986. http://www.eecs.tufts.edu/˜jacob/papers/tog.txt [ASCII]; http://www.eecs.tufts.edu/˜jacob/papers/tog.pdf [PDF].

Jacob,.R.J.K. Jacob, ‘‘What You Look At is What You Get: Eye Movement-Based Interaction Techniques,’’ *Proc. ACM CHI’90 Human Factors in Computing Systems Conference*, pp. 11-18, Addison-Wesley/ACM Press, 1990.

Jacob,.R.J.K. Jacob, ‘‘The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look At is What You Get,’’ *ACM Transactions on Information Systems*, vol. 9, no. 3, pp. 152-169, April 1991.

Jacob,.R.J.K. Jacob, ‘‘Eye Movement-Based Human-Computer Interaction Techniques: Toward Non-Command Interfaces,’’ in *Advances in Human-Computer Interaction, Vol. 4*, ed. by H.R. Hartson and D. Hix, pp. 151-190, Ablex Publishing Co., Norwood, N.J., 1993. http://www.eecs.tufts.edu/˜jacob/papers/hartson.txt [ASCII]; http://www.eecs.tufts.edu/˜jacob/papers/hartson.pdf [PDF].

Perez,.M.A. Perez and J.L. Sibert, ‘‘Focus in Graphical User Interfaces,’’ *Proc. ACM International Workshop on Intelligent User Interfaces*, Addison-Wesley/ACM Press, Orlando, Fla., 1993.

Pritchard,.
R.M. Pritchard, ‘‘Stabilized Images on the Retina,’’ *Scientific American*, vol. 204, pp. 72-78, June 1961.

Schmandt,.
C. Schmandt, M.S. Ackerman, and D. Hindus, ‘‘Augmenting a Window System with Speech Input,’’ *IEEE Computer*, vol. 23, no. 8, pp. 50-56, 1990.

Shneiderman,.
B. Shneiderman, ‘‘Direct Manipulation: A Step Beyond Programming Languages,’’ *IEEE Computer*, vol. 16, no. 8, pp. 57-69, 1983.

Starker,.
I. Starker and R.A. Bolt, ‘‘A Gaze-Responsive Self-Disclosing Display,’’ *Proc. ACM CHI’90 Human Factors in Computing Systems Conference*, pp. 3-9, Addison-Wesley/ACM Press, 1990.