Proactive Response to Eye Movements

Aulikki Hyrskykari, Päivi Majaranta & Kari-Jouko Räihä

University of Tampere, FIN-33014 University of Tampere, Finland

{ah, curly, kjr}@cs.uta.fi

Abstract: Proactive computing reinforces the need for non-command interfaces that fulfill the user's intentions without explicit commands. Eye gaze is a natural modality to be used in this connection, because it is also used proactively in everyday life. Previously eye-tracking has been used mostly for research purposes or in applications targeted for users with special needs. We present iDict, a gaze-assisted application for reading electronic documents written in a foreign language. iDict observes the reader's eye movements and behaves proactively by providing help when the user has comprehension problems. The lessons learnt (coping with inaccurate input data, need to control the level of proactivity, and delicacy of proactive adaptation) are of interest to the development of proactive applications in general and to eye-aware applications in particular.

Keywords: eye movements, proactive interaction, non-command interfaces, eye-tracking

1 Introduction

Proactive applications are able to adapt their behavior to the needs of the user without explicit, constant commands from the user. Proactive computing is a growing trend and a challenge for user interface design (Tennenhouse, 2000).

Eye movements are proactive in nature. Land and Furneaux (1997) write: "In everyday life, eye movements enable the eyes to gather the information required for motor actions. They are thus proactive, anticipating actions rather than just responding to stimuli." Similarly, our aim is to make the computer anticipate the user's actions based on information from natural eye movements: we try to predict the intent and needs of the user, so that the application may perform the action on behalf of the user.

Eye gaze has some unique properties that make it an excellent modality for proactive interaction between human and computer. The direction of gaze carries information about the focus of the user's attention (Just & Carpenter, 1976). The eyes are also extremely fast. For example, if the user wants to select something on the screen, the eyes are on the target long before the user moves the mouse (Zhai et al., 1999). By monitoring the eye movements we can know more about the user's state and intentions (Jacob, 1991), and react in a more natural way. This brings us closer to the goal of making the interface transparent, thus helping the user to work on the task instead of interacting with the computer (Nielsen, 1993).

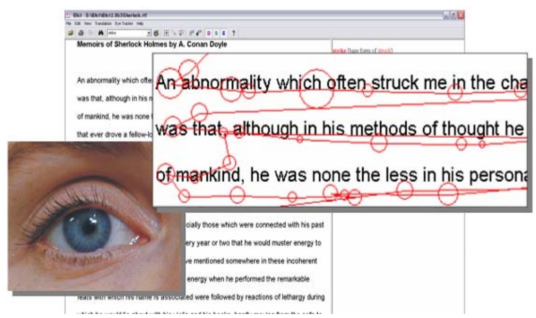

Figure 1: iDict monitors the user's gaze path.

We have developed iDict (Hyrskykari et al., 2000), a gaze-assisted application for reading electronic documents written in a foreign language. The iDict application can predict when the user needs help with foreign text by monitoring and interpreting the user's eye gaze behavior (see Figure 1). Even though the idea sounds simple, there are a lot of pitfalls to avoid when using natural gaze as input for an application. Many of the advantages and problems are discussed by Jacob (1993) in his paper on eye-movement based HCI techniques and non-command interfaces. We introduce practical solutions to some of the problems pointed out by Jacob.

In this paper, we discuss the lessons learned in designing, implementing and testing iDict. Many of the interaction issues are common to all eye-aware applications, and we will provide guidelines for other developers of gaze-assisted interaction.

2 Previous Work

To our knowledge, iDict is the first full application to make use of eye-tracking in a proactive manner. However, some demonstrators have been built and trials carried out in the past. We first review this previous work and then give some background on eye movement research in reading.

2.1 Eye-aware applications

The following examples demonstrate how the information of the user's point of gaze can be utilized to make the human-computer interaction more natural. These applications have "do-nothing", or "non-command" interfaces that do not require the user to explicitly command the computer.

In the Little Prince Storyteller (Starker & Bolt, 1990), an embodied agent reads aloud the story "Little Prince" by Antoine de Saint-Exupéry. The system "knows" the locus of interest based on the user's gaze behavior as the user looks at the objects on the screen. The program then changes the order of the narration according to the listener's interest.

An application developed in the Naval Research Laboratory (Jacob, 1991) shows information about ships and their locations. The screen is divided into two parts: on the right side of the screen there is a map with several ships, on the left there is extra information about the selected ship. As the user looks at a ship in the right window, the information in the left window changes accordingly. As soon as the user glances in the window on the left, the information about the ship is already there.

The eye-movement enhanced Translation Support System (Takagi, 1998) assists in the difficult task of translating text from Japanese into English. The system analyzes eye movements during the translation, detects patterns in eye movements and responds appropriately. For example, when the user scans through a Japan-English translation corpus the system automatically removes the already scanned corpus and continuously retrieves new information. If the user is at a standstill (pause due to hesitation), the system finds keywords in the sentence under focus and retrieves new corpus entries that are more likely to help in translation.

The MAGIC pointing technique (Zhai et al., 1999) works like magic; when the user looks at a selectable object on the screen (such as a button) the cursor is automatically warped to that object or to its proximity. Fine-grain tuning, if needed, is done using the mouse, which is much more accurate but slower.

The Reading Assistant (Sibert et al., 2000) uses eye gaze to trigger auditory prompting for remedial reading instruction. The application follows the user's gaze path and highlights the words of the text as the reading proceeds from word to word. As soon as the program notices hesitation it speaks out the word. It provides unobtrusive assistance to help the user with recognition and pronunciation of words. Similarly to iDict, the reading assistant application exploits the knowledge of how the gaze behaves during reading.

2.2 Gaze behavior during reading

When we read, our gaze jumps from word to word (see the visualization of a gaze path in Figure 1). The pauses between the jumps are called fixations. The jumps between the fixations are called saccades. Information is only acquired during fixations, when the eyes are still. The duration of a saccade is between 10 and 100 milliseconds. The duration of a fixation varies from 100 to 500 ms depending on the reader and the text (Rayner, 1998).

Reading is one of the most studied areas in eye movement research. We know much about how the eyes behave during reading, how the letters, words and sentences are perceived, and what kind of mental processes are assumed to take place during reading. Rayner (1998) gives a broad review of the different phases, of the development of theories, and of the main research results in the field.

Many researchers (e.g. Morrison, 1984; Henderson and Ferreira, 1990; Reichle et. al., 1998) have developed different models of reading, attempting to combine the vast amount of separate research results. Even though the models gave us insight about the relation between eye movements and the reading process, they are on far too low a level for our purposes. Long reading sessions produce eye movement data that is spatially inaccurate and affected by distracting elements that get the reader's attention in real life situations. The data is very different from the data collected in controlled experiments. Still, in many of our design decisions we were inspired by basic reading research.

3 iDict: A Reading Aid

The iDict application exploits knowledge of normal gaze behavior during reading by monitoring the reader's point of gaze and detecting incongruities.

iDict is a gaze-assisted environment for reading electronic documents written in a foreign language

(English). The use of iDict begins with a calibration of the eye-tracker. Then, the user just starts reading the text. As soon as the user hesitates while reading a word or a phrase, the embedded dictionaries are automatically consulted and a gloss (an instant translation) is provided (see Figure 2). A linguistic analysis of the text has been carried out in a pre-processing phase, and the glosses can be shown in a grammatically correct form: verbs are translated as verbs, nouns as nouns. Compound words and phrases can also be detected and translated (provided the dictionary supports them).

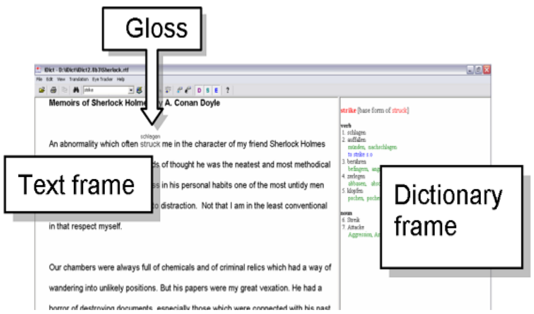

Figure 2: iDict provides help in reading foreign text.

If the reader wants more help than the gloss, she can get it by simply looking at the dictionary frame beside the text frame. When the user glances in the dictionary frame, the entry for the unfamiliar expression is retrieved from a dictionary and displayed in full. For the interaction the user does not have to do anything but read the text. In addition, the dictionary look-ups can also be triggered manually by clicking on the unfamiliar word with a mouse.

At the moment the target language can be chosen from Finnish, Italian, German, or English. The user can define how she wants the help to be given. For example, the gloss can be either shown (as in Figure 2), spoken, or both. The user can also specify what kind of information is shown in the dictionary frame: definitions, examples and synonyms can be displayed in addition to the translation.

4 Proactive Eye-tracking – Avoiding Pitfalls

When we started developing iDict, Finnish-speaking readers were observed in a series of tests in which they were reading English text. We also interviewed the readers about their reading habits. We found two main behavior patterns the readers follow when they encounter an unfamiliar word.

If the word seems essential for understanding the text, they pause reading and consult either a printed or an electronic dictionary. However, leaving the text interrupts the normal flow of reading and it takes time to recapture the text context when the reading continues. For this reason the readers often do not bother to look up the unfamiliar word, and the other behavior pattern takes place. The reader either trusts that—as the reading proceeds—the context will reveal the meaning of the unfamiliar word, or the word will turn out not to be critical in terms of understanding the text. Naturally, disregarding the word may lead to incorrect, or at least incomplete, comprehension. In fact, proactivity appears to be exactly what the readers need. They are not willing to take any explicit actions to get help because that would disturb their concentration on the text, even though they acknowledge that the help would be appreciated if it did not disrupt the reading process.

The goal of a proactive system is a transparent interface—in an ideal case. However, if the application is poorly designed the automatically triggered action very easily becomes irritating. During the development of iDict we uncovered many specific design details that have a distinct effect on how the user experiences the use of the application. So far eye-aware applications have mainly been targeted at special user groups, such as disabled users. In “the fourth era” of eye-tracking research (Rayner, 1998; Duchowski, 2002) the technology is becoming sufficiently mature to attract wider audiences. On the other hand, as a new input modality eye gaze creates new challenges, particularly in proactive applications. While developing iDict we made many observations and learnt many lessons that can be of interest for developers of other related applications.

Most of the problems we had to solve concern either the inaccuracy of the measured point of gaze or the too eagerly triggered proactive actions.

4.1 Coping with inaccuracy

The inaccuracy of the measured eye coordinates is a problem that is often bypassed with either a remark that the eye-tracking equipment will improve with time, or by setting restrictions on the use conditions (such as requiring the use of an exceptionally large font size or line spacing). However, the inaccuracy is a feature which we have to accept; gaze input will never achieve the accuracy of the mouse. Especially with general purpose applications the users are not likely to accept restrictions or requirements that affect their normal work environment. If the eye-tracking proves burdensome the cons may override the pros achieved from eye input information.

In order to solve this problem we developed algorithms that intelligently matched the measured fixation coordinates with the target words during reading. Nevertheless, occasional misinterpretations are unavoidable, particularly as we used a rather small font size (14 pt with 1.5 line spacing on a 1024x768 screen with 19" display). Therefore the application should provide some kind of feedback of the interpretations made by the application. Even in transparent interfaces the principles of exploiting eye movements should be visible to some extent to help the user understand the actions made by the application. The user should also have the option to easily correct the mistakes caused by inaccurately measured gaze coordinates.

4.1.1 Appropriate feedback

iDict offers two different feedback options for monitoring the performance of the eye movement interpretation; the user can activate them both, either, or neither of them.

The obvious feedback is to show a small "gaze cursor", which renders the measured point of gaze on the screen. This gives the reader an opportunity to intentionally "look off" (for example below the actual target) to get the target word construed right. Evidently a straightforward implementation of this feedback is not acceptable. The location of the gaze cursor is seldom precisely on the spot the user is really looking at, and trying to control the measured point distracts the process of reading. Furthermore, the constant movement of the visualized gaze cursor distracts quite a lot, as noted by Jacob (1993).

In our experience, there is a clear tendency that vertical inaccuracy dominates over horizontal inaccuracy. In the case of successful calibration the trackers give the horizontal coordinates much more accurately than the vertical coordinates. Often the measured coordinates hit on the previous or the next line instead of the line being read. Therefore we paid special attention to keeping track of the current line. We then tied the vertical coordinate of the gaze cursor with the presumed line of reading (Figure 3).

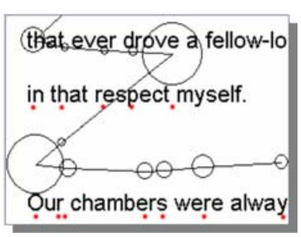

Figure 3: A visualization of the measured gaze path, and the corresponding gaze cursor movement.

This generated an illusion that the gaze cursor locked in to the line the user was reading. Without tracking the currently read line the gaze cursor was very unstable and provoked the reader to follow its movements; now the gaze cursor (its history is seen as small spots in Figure 3) appeared to follow smoothly along.

The other form of feedback the reader may choose is a line marker, which is a faint gray underline below the line of reading—or below the line that iDict assumes to be the line of reading (Figure 4). We presumed that the line marker is a more sensitive way to show feedback. It generates less visual noise since it changes only when the line of reading changes. The preliminary tests with users indicated that the line marker does not disturb the reader; on the contrary, it helps the reader to keep the right line, especially when she shifts to the next line, or when she returns to the text after checking the word entry from the dictionary frame.

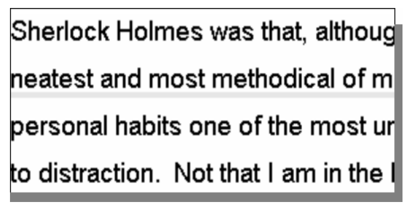

Figure 4: A line marker helps to stay on the right line.

At the moment we are not sure which of the feedback modes the users prefer, as the pilot testers have reported very different preferences. Some of them absolutely wanted to see the gaze spot while others found the line marker very pleasant.

4.1.2 Sense of being in control

In proactive applications the user may experience a loss of control. She does not know what is happening, why it is happening and if there is anything she can do to affect it. The feedback modes iDict provides make the application's behavior understandable, and in addition to that, the reader can correct the mistakes the background interpreter algorithms make. If she notices that iDict assumes the line of reading to be incorrect, she can adjust the interpretation by pressing the arrow keys up or down. Similarly she may correct the word horizontally to the previous or next one by pressing the left or right arrow keys.

4.2 Proper level of proactivity

The level of proactivity is a very delicate issue, as illustrated by the frustration often created by the Microsoft Office Assistant. Individual differences between users make the problem even harder.

Without careful design, the user becomes frustrated and very quickly turns off features that react too eagerly to her behavior.

4.2.1 Different reading styles?

In the beginning we concentrated on trying to detect different "modes" of the user: whether the user is reading the text, skimming it, or maybe just in a state of dormant gazing (Hyrskykari et al., 2000). In some cases, it is quite difficult to distinguish the different modes, for example the mode of dormant gazing from the prolonged gazing due to comprehension difficulties. However, we discovered that knowing the mode of the user is not decisive for this kind of application. Reading styles vary a lot. Some of the readers read linearly with few regressions to the previously read words or sentences while some make regressions regularly, either in a structured manner or quite randomly (Hyönä et al., 2002).

In the present version of iDict we use the total time spent on a word as the main threshold value for activating the automatic help (the gloss). The total time also includes the durations of the regressive fixations. The motivation to use it as the main trigger is that it works well for all reading styles. The findings in reading research show that large reinspecting saccades are often extremely accurate (Kennedy, 2001; Kennedy et. al., 2003). Let us consider the situation in which the reader has trouble with an unfamiliar word. What happens? She either stays staring at the word for several fixations, or she makes regressive fixations to the word trying to figure out whether she already understands the meaning of the word in the context. In either case the total time spent on the word increases and eventually triggers the translation.

In addition to total time, we also use other guiding factors to make the decision whether the help is needed or not. They include word frequency (a rare word is interpreted as an unfamiliar word more likely than a common word) and word length (a long word is allowed to have a longer perception time than a short one).

Even though we try to avoid unwanted translations they are bound to appear every now and then. Therefore they must be designed in an unobtrusive way.

4.2.2 Delicacy of showing a gloss

The visual system is sensitive for changes in the visual field. A moving or flickering object draws attentions even though the object is not in the focus area (Bartram et al., 2001). In applications performing actions proactively we must pay special attention to avoid the situations that needlessly distract the user's main function (in this case reading).

The gloss is shown right above the word or phrase that appears to be problematic. It is designed to be as discreet as possible to avoid extra visual noise. In fact, the pilot testers reported that they seldom noticed the unwanted translations. An explanation for this is that the user's eyes have already moved on to the succeeding words when the gloss is shown, thus leaving it out of the focus area. The user's attention remains on reading and on understanding the text. Correspondingly, the removal of the gloss should happen imperceptibly without needless flashing or flickering.

The user can define how many glosses are visible at a time. If, for example, she chooses to keep ten glosses visible, the most recent one is displayed in black, but the nine preceding glosses fade to gray in time. Sometimes the reader may want to recheck a gloss given before. The total time of the word does not accumulate while its gloss is visible, thus preventing needless (distracting) redrawing.

4.2.3 Choosing the correct dictionary entry

What is the level of acceptable proactivity when dictionary entries are retrieved and displayed in the dictionary frame? What does the reader expect to see when she transfers her eyes there? There are two possibilities. The word for which the dictionary entry is given may be either the last word the reader focused (fixated on), or the word for which the most recent gloss was given.

When we tried the first solution the effect was that the reader was never quite sure for which word she would get the dictionary look-up. She is not necessarily fully aware of which word she had her eyes on at the moment when she decided to look for additional help. On the other hand, the last fixation may sometimes be attached to a wrong word due to inaccuracy problems. Since the reader gets no feedback of the fixated words, the words for which she gets the dictionary entries may seem somewhat random.

As before, the principle of understandable behavior was the leading guideline for designing this feature, and we chose the second solution in its implementation. When the gloss and the dictionary entry are given for the same word, the reader knows what to expect when she looks in the dictionary frame. This way she feels that she is in control of the contents of the dictionary frame: she first triggers a gloss for a word in the text and after that she may ask for more details if she stills needs them (Figure 5).

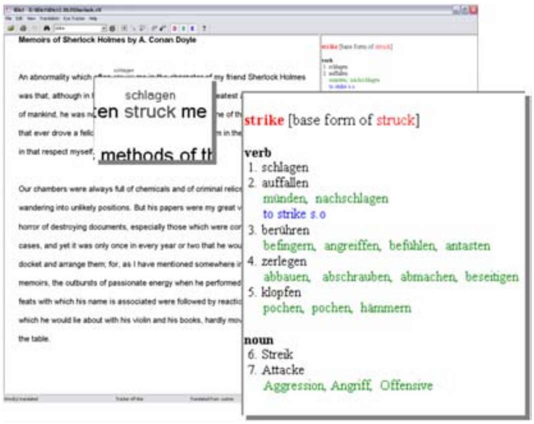

Figure 5: A gloss and a dictionary entry for *struck*.

In order to avoid needless visual noise, the dictionary frame is updated (to show a new entry) only when the reader turns her eyes on it, otherwise its contents stay stable.

At first we wondered why it was so difficult for the readers to trigger the first dictionary entry by looking at the dictionary frame. By examining the gaze path we realized that instead of looking in the frame, the users looked at the frame borders and consequently iDict did not react. In the beginning there was nothing in the frame to look at; it is difficult to fixate on an empty space. The problem disappeared when we initialized the dictionary frame at the beginning of each session by adding a prompt "Look here to get a dictionary entry".

4.2.4 Sensitivity control

As mentioned before, reading habits are individual. Additionally, reading skills vary a lot. This is especially the case when reading text written in a foreign language. Thus, the level of proactivity should also adapt to the reader. We are at the moment developing a personalized triggering function that takes advantage of the user's reading history. Furthermore, it is important that the user can adjust the level of proactivity herself. If she feels that iDict is too swift in giving help she can turn down the overall sensitivity of the system or, if she feels that she clearly has to prolong her gaze in order to get help, she may turn the sensitivity up.

5 Experiences

Even though controlled tests of iDict have not yet been performed, a fairly extensive test of the application took place in the IST Conference Exhibition in Copenhagen in November 2002.

5.1 iDict showcased: observations

We demonstrated iDict on a stand for three days and provided all visitors with a chance to try out the application themselves (Figure 6). It was very instructive for us in the sense that the testers covered a very large distribution of different people starting from groups of school children to senior officials working in the administration of the European Union. In the exhibition we used SMI's remote iView X eye-tracking system, which has an ability to tolerate moderate head movements. During the exhibition about 60 people tried out the application. We filled an evaluation sheet for each tester, whenever we had time to do so, resulting in a total of 47 evaluation note sheets.

Figure 6: Visitors of the IST conference trying out iDict.

We recorded the success of calibration and the experiences of iDict's performance. Considering that many of the testers were totally unacquainted with the whole concept of eye-tracking, and that this was the first large-scale exposure of the system, we were more than satisfied with the results. 57 % of the testers expressed that they felt iDict worked very well. For 24% of testers the performance was satisfactory and for only 18% of testers the performance appeared to be poor. Many of the testers for whom iDict performed well were excited enough about its performance to bring other visitors to experience it themselves.

A large share of the problems was due to problems with calibration. For 44% of the users the calibration was easy, for 30% we had to repeat the calibration and for 26% the calibration took a longer time, several trials. Hence, for a number of users, the poor calibration had a straightforward effect on iDict's performance. However, there was only one user for whom we could not get the tracker calibrated at all.

Another considerable source of problems was that users often looked aside or even turned around to discuss the system. Even though iViewX is able to follow smooth head movements, it loses track if the subject makes sudden movements. Luckily, iViewX has a nice feature of being able to catch the eye again if the user returns back near to the position where she was during the calibration.

5.2 Natural vs. intentional eye movements

The original idea of iDict was to provide help on the basis of natural eye movements. However, when appropriate feedback is provided, the users quickly adapt to use their gaze intentionally; when they want help with a word they prolong gazing on it.

We found this phenomenon in preliminary tests when some readers reported that they could not get a gloss for the word even if they "try to get iDict to respond". A closer look at the eye movement data revealed that the readers had an ability to make surprisingly long continuous fixations when they concentrated on staring at a word. The fixations reported by the eye-tracker could last from 1000 ms to even nearly 2000 ms, and normally they do not enter the application until the fixation ends. A typical total time threshold we use for triggering a word is 800 ms (it varies depending on other triggering factors). The readers experienced that the application freezes: "it does not respond to my request". For that reason we could not settle for the fixations the tracker provides us, but we had to use the raw data and split the long fixations in order to get them forwarded in time to the application.

6 Conclusions

We have described experiences obtained when using eye movement information for implementing a proactive application iDict. When we started the development of iDict, we confronted many skeptics. An expert reviewer of one of our research proposals just recently stated that "The use of gaze for interaction has appeared in the literature 7 or 8 years ago and has generally been rejected as difficult to learn and unpleasant to use". We believe that iDict proves that the concept of using eye gaze data as input for applications is viable and that eye-tracking has a lot to contribute in the field of human-computer interaction.

It is indisputable that turning from the use of eye-trackers as diagnostic and research devices to using them as input devices is not without problems. The problems arise from the inaccuracy of the measured point of gaze and from the inaccuracy of the interpretation of eye behavior. Based on our experiences we introduced design principles that should be taken into consideration when eye-aware applications are developed.

In the early 1990's, non-command interfaces were predicted to be one of the future trends of interface evolvement (Nielsen, 1993; Jacob, 1993). Proactive applications share the idea of making the interface disappear by helping the user to concentrate on working with the task instead of giving instructions to the computer. A good design should find the right balance between proactive and reactive behavior: there are levels of appropriate proactivity and transparency depending on the context. We summarize below the three main principles we uncovered for designing proactive eye-aware applications.

First, even in transparent interfaces, the system state should be visible. The user should be provided with appropriate feedback. The user should understand why the application takes a turn to make some actions automatically. This is especially important in eye-aware applications where occasional misinterpretations are unavoidable. Moreover, if the user understands the principles according to which the proactive actions take place, also the occasional mistakes are much easier to accept. An example of applying the principle is the line marker used to give feedback of the presumed line being read.

Second, even non-command interfaces should be controllable. The user should always have a feeling that she is in charge. For example, in iDict the user can correct the misinterpreted line of reading.

Third, proactive applications should not be too active. The users experience the automatically triggered actions individually, and too eagerly triggered actions easily become irritating. The user should always be able to tune the sensitivity of the system to satisfy her personal needs. Also the proactive actions should be designed in a way that the potentially needless, unwanted actions do not disturb the primary functions of the user.

The examples given in this paper show how the principles are materialized in iDict. The first experiences are extremely encouraging, but there is much to be improved. Even though the function used for triggering the gloss was quite rough it worked surprisingly well. Careful research on the factors that reveal the comprehension difficulties can make the triggering even more precise. At the moment we are also studying how iDict could automatically adapt to the user's individual characteristics of reading. Further development of iDict requires testing the performance of iDict in controlled experiments.

Acknowledgements

The work was supported by the EU's IST Programme. We want to take the opportunity to thank our project partners, Timo Järvinen and Pasi Tapanainen from Connexor Oy, Wulf Teiwes from Sensomotoric Instruments GmbH, Geoffrey Underwood, Kate Roberts and Lorraine Bell from the University of Nottingham, and Giancarlo Bo and Roberto Vaccaro from Giunti Interactive Labs, who all provided us with inspiring discussions and feedback. The linguistic analysis for the text was implemented in co-operation with Connexor.

References

Hyrskykari, A., Majaranta, P., Aaltonen, A. & Räihä, K.-J. (2000), Design issues of iDict: a gaze-assisted translation aid. In *Eye Tracking Research & Applications Symposium 2000,* ACM, 9–14.

Bartram, L., Ware, C., & Calvert, T. (2001), Moving Icons: Detection and Distraction. In *Proc. Interact '01*, Tokyo, IOS Press, 157–165. Available at http://www.cs.sfu.ca/~lyn/personal/Publications/Interact2001Final.pdf

Duchowski, A. T. (2002), A Breadth-First Survey of Eye Tracking Applications. *Behavior Research Methods, Instruments, & Computers (BRMIC), 33 (4)*. Available at http://www.vr.clemson.edu/~andrewd/research/vislab/docs/BET107cr.pdf

Henderson, J.M., Ferreira, F. (1990), The effects of foveal difficulty on the perceptual span in reading: Implications for attention and eye movement control. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 16*, 417–429.

Hyönä, J., Lorch, R.F., Jr., & Kaakinen, J.K. (2002). Individual differences in reading to summarize expository text: Evidence from eye fixation patterns. *Journal of Educational Psychology, 94*, 44–55.

Jacob, R.J.K. (1991), The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Transactions on Information Systems 9* (*3*), 152–169.

Jacob, R.J.K. (1993), Eye Movement-Based Human-Computer Interaction Techniques: Toward Non-Command Interfaces. H. R. Hartson & D. Hix, (eds.), *Advances in Human-Computer Interaction, Vol. 4,* Ablex Publishing Co., 151–190. Available at http://www.cs.tufts.edu/~jacob/papers/hartson.pdf

Just, M. A. & Carpenter, P. A. (1976), Eye fixations and cognitive processes. *Cognitive Psychology 8*, 441–480.

Kennedy, A. (2001), The Reader's Spatial Code (abstract only). 11th European Conference on Eye Movements Turku, Finland. Available at http://congress.utu.fi/ecem11/reading/kennedy.htm

Kennedy, A., Brooks, R., Flynn, L-A., & Prophet, C. (2003), The Reader's Spatial Code. In Hyönä, J. et al. (eds.), *The Mind's Eye. Cognitive and Applied Aspects of Eye Movement Reseach.* Elsevier Science.

Land, M. F. & Furneaux, S. (1997), The knowledge base of the oculomotor system. *Philosophical Transactions: Biological Sciences, 352 (1358)*, The Royal Society 1997, 1231–1239.

Morrison R. E. (1984), Manipulation of stimulus onset delay in reading: Evidence for parallel programming of saccades. *Journal of Experimental Psychology: Human Perception and Performance, 10*, 667–682.

Nielsen, J. (1993), Noncommand user interfaces. *Comm. ACM 36 (4)*, 82–99. Available at http://www.useit.com/papers/noncommand.html.

Rayner, K. (1998), Eye Movements in Reading and Information Processing: 20 Years of Research. *Psychological Bulletin, 124 (3)*, 372–422.

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

Sibert, J. L., Gokturk, M., & Lavine, R. A. (2000), The Reading Assistant: Eye Gaze Triggered Auditory Prompting for Reading Remediation. In *Proc. UIST'00*, ACM, 101–107.

Starker, I. & Bolt, R. A. (1990), A gaze-responsive self-disclosing display. In *Proc. CHI '90*, ACM, pp. 3–9.

Takagi, H. (1998), Development of an eye-movements enhanced translation support system. In *Proc. APCHI'98* (Asian-Pacific Computer-Human Interaction Conference), 114–119.

Tennenhouse, D. (2000), Proactive computing. *Comm. ACM 43 (5)*, 43–50.

Zhai, S., Morimoto, C., & Ihde, S. (1999), Manual And Gaze Input Cascaded (MAGIC) Pointing. In *Proc. CHI'99,* 246–253. Available at http://www.almaden.ibm.com/u/zhai/papers/magic/magic.pdf