

# Exploring Eye Tracking to Increase Bandwidth in User Modeling

Cristina Conati, Christina Merten, Kasia Muldner, David Ternes

Department of Computer Science, University of British Columbia,  
2366 Main Mall, Vancouver, BC, V6T1Z4, Canada  
{conati, merten, kmuldner, ternes}@cs.ubc.ca

**Abstract.** The accuracy of a user model usually depends on the amount and quality of information available on the user's states of interest. An eye-tracker provides data detailing where a user is looking during interaction with the system. In this paper we present a study to explore how this information can improve the performance of a model designed to assess the user's tendency to engage in a meta-cognitive behavior known as self-explanation.

## 1 Introduction

One of the key dimensions that characterizes a user modeling problem is *model bandwidth* [15], i.e., the amount and quality of information available to the model to assess the user's states of interest (e.g., knowledge, goals, emotions). If a model assesses a user's task performance (or a user's *final states*, following the classification in [15]), high bandwidth is already achieved through information on task-related interface actions. However, if the model must assess the higher level *mental states* underlying a given behavior, high bandwidth requires explicit information on these states, which are seldom fully observable. In this case, bandwidth can be increased through interface mechanisms that force the user to make the states of interest explicit (e.g., by showing all the steps used to generate a problem solution). Unfortunately, this approach has the potential to be highly intrusive.

In this paper, we present research on exploring eye tracking as a means to unobtrusively raise bandwidth in user models. In particular, we discuss findings from a user study that explores the usage of users' gaze patterns to understand whether students engage in a meta-cognitive behavior known as *self-explanation* [4], during interaction with an Intelligent Learning Environment for mathematical functions.

Retrospective analysis of eye movements has been long used in Cognitive Psychology as a tool to help understand both motor and cognitive processes (e.g., [9]), as well as in HCI for off-line interface evaluation (e.g., [8]). There has also been fairly extensive research in using eye gaze as an alternative form of input to allow a user to explicitly operate an interface (e.g., [8, 11]).

There is a much smaller body of work on real-time processing of a user's gaze to interpret a user's behavior beyond interface operation to enable on-line adaptation of the interaction. Some of this work uses gaze tracking to help assess user *final states*,

such as reading performance in a system for automatic reading remediation [13], or *what* task a user is performing independently from the underlying application (e.g., reading email vs. reading a web page) [14]. Others have explored using gaze data to assess user *mental states* such as *interest* in various elements of an interactive story [7], or *problem-solving strategies* in a tutoring system for algebra [6]).

Our work extends this body of research by exploring if and how eye tracking can help assess *mental states* related to the meta-cognitive, domain-independent skill of *self-explanation*. Self-explanation is the process of explaining to oneself a piece of instructional material, and has been shown to greatly improve learning [4]. It has also been shown that many students tend to not self-explain spontaneously. For this reason, there has been increasing interest in devising computer-based tools that can help students self-explain. The support provided by most of these tools, however, is not based on an explicit model of a student's self-explanation behavior. The Geometry Explanation Tutor prompts students to self-explain *every* problem-solving step in an Intelligent Learning Environment (ILE) for geometry [1]. Normit-SE prompts students to self-explain every *new* or *incorrect* problem-solving step in an ILE for data normalization [10]. This approach is potentially intrusive, since it may force spontaneous self-explainers to produce redundant and unnecessary self-explanations. In contrast, [5] proposes a framework that provides individualized support for self-explanation based on an explicit model of a student's self-explanation needs. The model uses information on both student knowledge and reading patterns to assess self-explanation during example studying in the domain of Newtonian physics. Reading patterns are tracked via a *poor-man-eye-tracker* interface that forces students to explicitly uncover the various parts of the studied example via mouse movements.

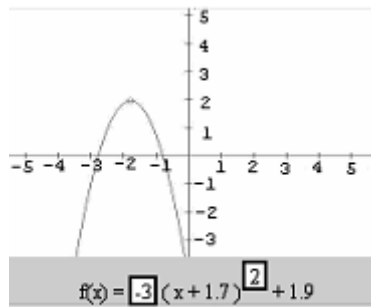
We have been working on a similar model of self-explanation to aid the assessment of the effectiveness of student *exploratory* behavior and consequent learning in the Adaptive Coach for Exploration (ACE) [2, 3]. ACE is an ILE designed to help students learn about mathematical functions through free exploration of interactive simulations, rather than through more traditional problem solving activities. Like [5], ACE could benefit from information on student attention patterns to more reliably assess whether a student is self-explaining the phenomena observed in the interactive simulations. However, because of the nature of the interaction, i.e. unconstrained exploration, we felt that it would be too intrusive to use a poor-man-eye-tracker mechanism to track user attention. Thus, we are exploring the usage of real-time eye-tracker data to inform our model. In the rest of the paper, we first describe ACE. We then provide a high level description of the ACE student model. Next, we illustrate a user study that we have conducted to understand what information an eye-tracker can provide about a student's self-explanation behavior. Finally, we discuss the implications of our findings.

## 2 The ACE Open Learning Environment

ACE is an adaptive open learning environment for the domain of mathematical functions. Open learning environments rely on the assumption that if a learner can freely explore the instructional material, she can acquire a deeper understanding of the

target domain. However, various studies have shown that not all students can explore effectively on their own (e.g., [12]). Thus, ACE provides activities for students to freely explore mathematical functions, tracks their exploratory behavior and provides tailored suggestions to improve this behavior when needed.

ACE's activities are divided into units, which are collections of exercises. Figure 1 shows the main interaction window for the Plot Unit. We will focus on this unit



**Fig. 1.** The Plot Unit

throughout the paper because it is the most relevant to the eye tracker research presented in later sections. In the Plot Unit, a learner can explore the relationship between a function's graph and equation by moving the graph in the Cartesian plane and observing how that affects the equation (displayed below the graph area). The student can also change the equation parameters and see how these affect the graph.

All student interface actions are used to update the ACE student model, designed to assess if a student is exploring and learning effectively or if she needs help from ACE. For more detail on ACE's interface and coaching component see [2]. In the next subsection, we describe the high level structure of the ACE student model, and the components that allow the model to assess self-explanation behavior.

## 2.1 The ACE Student Model

ACE's student model uses a Dynamic Bayesian Network to assess the effectiveness of a student's exploratory behavior in ACE. The main source of evidence in the model comes from observing students perform *Relevant Exploration Cases* in an exercise (e.g., changing the position of a function graph in the Plot Unit, so that it has a *positive intercept* with the x-axis; changing the equation so that it has an *even exponent*). Evidence of these cases is then propagated in the model, together with the current assessment of relevant student knowledge, to assess higher level dimensions of student exploration, such as exploration of exercises and of general domain concepts (e.g., the input/output relation for different types of functions) [2].

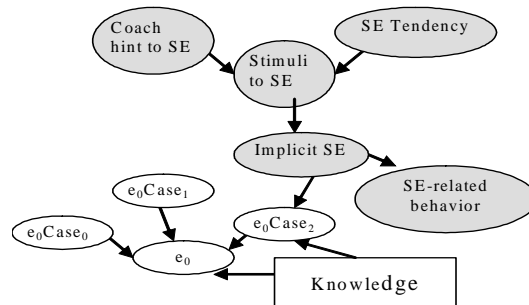
For a student to effectively explore a case, she must both perform an action and self-explain changes that it generates in the environment [3]. Thus, the ACE student model includes self-explanation as one of the factors that influence the assessment of student exploration.

## 2.2 Assessing Self-Explanation in the ACE Student Model

Assessing whether a student is spontaneously self-explaining is a typical user modeling problem in which it is hard to achieve high bandwidth, unless we ask students to explicitly input their self-explanation in the system. Doing so, however, can be intrusive and annoying for those students who can self-explain on their own.

The alternative is to gather information from sources that may provide indirect evidence on *implicit* self-explanation, i.e. self-explanation that happens in the student’s head.

Figure 2 exemplifies how we leverage these sources in the part of the ACE student model that tracks implicit self-explanation. In Figure 2, nodes  $e_0Case_0$ ,  $e_0Case_1$  and  $e_0Case_2$ , represent three relevant exploration cases of a generic exercise  $e_0$ . This model fragment corresponds to the learner having performed an action corresponding to the exploration of  $e_0Case_2$ . Nodes representing the assessment of self-explanation are shaded grey. As the figure shows, the two sources of information that the model uses to assess the occurrence of implicit self-explanation for a given exploration case are *Stimuli to SE* and *SE-related-behavior*. *Stimuli to SE* is the probability that the learner has stimuli to self-explain either from her general *SE tendency* or from one of the hints that ACE is designed to provide when a student is assessed to be a low self-explainer (node *Coach hint to SE*). The node *SE-related-behavior* represents all the available evidence that a student is actually self-explaining the exploration case just generated. The first version of this model that we proposed in [3] only included time spent on each exploration case as behavioral evidence. The conditional probabilities defining the relation between time and self-explanation were based on our subjective judgment, to represent the assumptions that (1) no self-explanation can happen if a student switches too rapidly from one exploration case to the next; (2) the longer a student dwells on a case the more likely it is that she is trying to self-explain it. Time, however, can be an ambiguous predictor. First, it is hard to define what “too rapidly” means for different students. Furthermore, a student may be completely distracted during a long interval between exploration cases.



**Fig. 2.** The ACE student model

Thus, we chose to explore an additional source of evidence of self-explanation behavior, i.e., the student’s attention patterns during the exploration of a given case. The intuition here is that self-explanation may be more likely if the student attends to the parts on the interface showing the effects of a specific exploratory action (if the student, for instance, switches attention from the graph area to the equation area after moving the graph in the Plot Unit). To unobtrusively obtain evidence on student attention patterns we used real-time processing of eye-tracking data. To collect empirical data on the mapping between actual student self-explanations, time and attention patterns, we ran a user study, described in the next section.

### 3 User Study

In this study, we collected data from 18 students using ACE while their gaze was tracked by an Eyelink I eye-tracker, developed by SR Research Ltd., Canada. This is a fairly intrusive head mounted eye tracker, that we used because it was available to us through the psychology department at the University of British Columbia. However, the same data could be easily obtained through a completely non-intrusive remote eye-tracker, consisting of a small camera which sits on top of the monitor or on some other flat surface (e.g. IView X Red from SensoMotoric Instruments, USA).

All the study participants were non-science university students (i.e. students that had not taken high school calculus or first-year college math). Each participant received a brief introduction to ACE and instructions to try and verbalize all his/her thought processes while using the system. The participant then went through a calibration phase with the eye tracker, and finally used the system for as much time as needed to go through all the units. This varied from 20 minutes to close to an hour. All the student exploration cases were logged (567 in total), along with raw data from the eye tracker, as well as complete video and audio data of the interaction.

#### 3.1 Data analysis

To understand how attention patterns and time per exploration case relate to self-explanation, we needed to obtain from the study data points on actual explicit positive and negative self-explanation episodes. (Here, “negative self-explanation” indicates situations in which students did not self-explain, not situations in which students self-explained incorrectly, consistent with the original definition of self-explanation [6].)

We had two observers analyze the recorded audio protocols in search of such episodes, and then create the link between the verbal episodes and the corresponding exploration cases in the log files. This turned out to be a much more laborious process than expected, due to two factors.

First, we quickly realized that not all verbal episodes could be unambiguously classified as positive or negative self-explanations. This is not surprising because, although there has been extensive research on what constitutes self-explanation in various problem solving domains (e.g., Newtonian physics, statistics, geometry), ours is the first attempt to understand self-explanation in an exploratory learning environment for mathematical functions. We tackled this problem by having the two observers independently label a subset of the audio data, then compare their classifications, possibly reconcile them and devise a detailed coding scheme based on this discussion. The coding scheme was then used to analyze the rest of the data, and only episodes on which the coders fully agreed were used in the rest of the analysis (the intercoder reliability was 93% in this phase). In the coding scheme, students utterances were classified as self-explanation if they expressed a conclusion about a domain-specific principle related to the exploration process (e.g., “*when I increase the coefficient here, the line gets steeper*”) regardless of correctness, or if they predicted the result of an action just before it occurred (e.g., “*putting a negative sign here will*

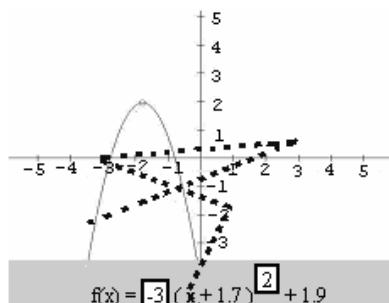


Fig. 3. Sample gaze shift

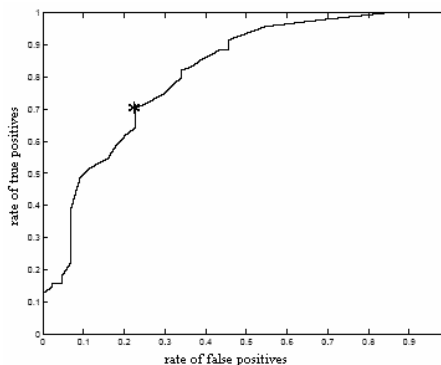


Fig. 4. ROC curve for time as a filter for self-explanation

turn the curve upside-down”). It is assumed here that if a student predicts the result of an action, she will watch to see if she is right and thus self-explain *after* the action. Simply narrating the outcome of each action once it happened (e.g., “*this number just changed to a 3*”), or isolated statements of confusion (e.g., “*I don’t understand*”) were not considered self-explanation. However, tentative explanations followed by expressions of confusion were coded as self-explanation.

The second factor that increased the complexity of data analysis was difficulty in determining which action each coded utterance corresponded to. The observers at first assumed that subjects’ utterances always pertained to whatever exploratory action they had just taken. However, while analyzing the video data they realized that this was not always the case, particularly for users who showed great reluctance to think aloud. These learners had to be repeatedly prompted by the observers to speak, so some of the conclusions they shared weren’t reached as they spoke, but related to self-explanation that occurred a few minutes earlier. The observers solved this problem by looking at every coded episode and matching it to its corresponding action. Thirteen coded episodes were discarded because the match was ambiguous.

While both parts of the above coding process resulted in the elimination of data points, the factor that had the greatest impact on the amount of data that we could get from the study was students’ willingness to verbalize their thoughts. We found that a number of students were incapable or unwilling to think aloud, even if they were periodically reminded to do so. Without such verbalizations, the coders could not tell whether a student had self-explained or not. Thus, of the 567 exploration cases recorded in the log files for all students, only 149 could be classified in terms of associated self-explanation.

Once positive and negative self-explanation episodes were identified and mapped onto specific exploration cases, we proceeded to analyze the correspondence between these episodes, gaze information, and time students devoted to each case.

Raw eye tracker data was parsed by a pattern detection algorithm we developed to detect switches of attention (“gaze shifts”) among the graph panel, the equation area, and any other non-conspicuous areas in the Plot Unit. As we mentioned earlier, these are the gaze patterns that we hypothesize to be associated with self-explanation in the plot unit. A sample gaze switch appears in figure 3. Here a student’s eye gaze (shown

as the dotted line) starts in some untracked area below the screen, moves to the equation region and then hovers around the graph region above. The data-parsing algorithm uses fixation coordinates from the eye-tracker and matches them to the appropriate ACE interface region. Next, it searches the data for the pattern of making changes in one region and then looking at the other to observe the outcome, i.e. having a gaze shift. When this pattern is found, a tag is placed in the ACE log file to synchronize the switch with the appropriate exploration case.

To analyze the relationship between time per exploration case and self-explanation, we first compared average time spent on exploration cases that were accompanied by self-explanation (24.7 seconds) and those that were not (11.6 seconds). The difference is statistically significant at the 0.05 level, suggesting that time per case is actually a fairly reliable indicator of self-explanation.

To turn time into a predictor of self-explanation, we then determined a threshold  $T$  so that an action could be classified as self-explained if the student spent more than  $T$  seconds on it. To choose the optimal threshold, we built a Receiver Operating Characteristic (ROC) curve (figure 4). The ROC curve is a standard technique used in machine learning to evaluate the extent to which an information filtering system can successfully distinguish between relevant data (episodes the filter correctly classifies as positive, or *true positives*) and noise (episodes the filter incorrectly classifies as positive, or *false positives*), given a choice of different filtering thresholds. Figure 4 shows the ROC curve we obtained for time, where each point on the curve represents a different threshold value. As it is standard practice, we chose as our final threshold the point on the curve that corresponds to a reasonable tradeoff between creating too many false positives and creating too few true positives (16 seconds, labeled by an asterisk on the curve in figure 4).

### 3.2 Results

Figure 5 categorizes our 149 data points into episodes with and without self-explanation (99 circles and 50 triangles, respectively). The vertical line further categorizes the points into those with and without a gaze shift (GS) between graph and equation pane in the plot unit. The horizontal line separates points with elapsed time above or below 16 seconds. The raw data is also presented in a table adjacent to the histogram. ROC curves were used to find that when time is used in combination with eye tracking data, 16 seconds continues to be the optimal threshold.

Table 1 shows different measures of self-explanation classification accuracy if we use as predictor: (i) the eye-tracker to detect gaze shift; (ii) time per self-explanation case; (iii) both predictors. Accuracy is reported in terms of true positive rate (i.e. percentage of self-explanation cases correctly classified as such, or *sensitivity* of the predictor) and true negative rate (i.e. percentage of “no self-explanation” cases correctly classified as such, or *specificity* of the predictor). We also report a combined measure, which is the average of the two accuracies. As the table shows, time alone has a higher *sensitivity* than gaze shift, i.e. the episodes involving self-explanation were more likely to take over 16 seconds than to include a gaze shift. However, the eye-tracker alone has comparably higher *specificity*, i.e. the cases without self-

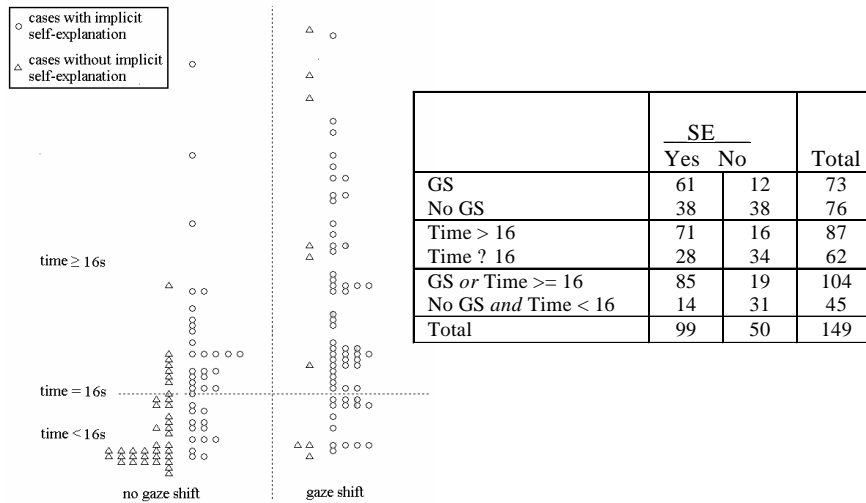


Fig. 5. Dual histogram (left) and raw time/GS data (right)

explanation were more likely to involve the absence of a gaze shift than shorter time per exploration case. The two predictors have comparable combined accuracy.

This may suggest that the gain of using an eye tracker is not worth the cost of adding this information to the ACE model. However, there are a few counterarguments to this conclusion.

First, it should be noted that time accuracy here is probably artificially high. One of the drawbacks of using time as predictor of self-explanation is that the amount of time elapsed tells the model nothing about the student's behavior between actions. During a long time spent on a given case, a student may be doing or thinking of something completely unrelated to ACE. This seldom occurs in our data, but we should bear in mind that students were in a laboratory setting with little available distractions, in the presence of an observer and wearing a rather intrusive device on their head. All these factors are likely to have made it more difficult for the students' thoughts to wander from the task at hand, resulting in time being a more reliable indicator of self-explanation than it would be in actual practice.

Second, we found that the sensitivity of the eye-tracker may be higher than our data shows. The program that synchronizes gaze shifts with actions assumes that a

Table 1. Classification accuracy of different predictors

	Eye-tracker	Time	Eye-tracker + Time
True Positive Rate ( <i>sensitivity</i> )	61.6%	71.7%	85.8%
True Negative Rate ( <i>specificity</i> )	76.0%	68.0%	62.0%
Combined Accuracy	68.8%	69.85%	73.9%

student performs an exploratory action and then carries out a gaze shift to observe the changes it generates. Thus, each gaze shift or lack thereof is associated with the preceding action. In our logger, an action involving a change in the function equation would be captured only when a student finishes typing and presses "enter". However,



it is possible that in some cases the student wanted to catch the change in the graph when it happened, and thus would look up at the graph region after typing but before pressing enter. Our logger would incorrectly record this gaze shift to be associated with the action *before* the current one. Of the 38 false negatives generated by the eye-tracker, 21 had a gaze shift associated with the preceding action in the log file, and are thus consistent with the above scenario. If we were to switch the matching of these gaze shifts with the following log file action, the sensitivity of the eye-tracker would increase to 86.8%, and that of eye tracker plus time would reach 92.5%. We plan to run more subjects with a revised logger to clarify this issue.

Third, combining gaze shift and time into one predictor substantially improves sensitivity. That is, if an action is classified as self-explained when there is *either* a gaze shift *or* more than 16 seconds elapsed time, most of the self-explanation episodes (85.8%) are correctly recognized. This increase also causes the combined accuracy to improve. However, as sensitivity increases, specificity is reduced and only 62% of the episodes that lack self-explanation are discovered by the model. This situation is shown in figure 5. With the combined model, all data points to the right of the vertical line or above the horizontal time threshold line are classified as self-explained. As a result, most of the episodes with self-explanation are found but many of those without self-explanation are incorrectly classified.

Here a tradeoff appears between sensitivity and specificity. Depending on how the system is used, it may be most important to correctly classify self-explanation when it occurs than to detect the lack thereof. This is the situation when letting natural self-explainers explore without interruption is given highest priority. Here, using the combination of eye-tracker and time data is best. Alternatively, it may be more important to make sure that the system intervenes wherever it is necessary. Then failing to identify lack of self-explanation is a bigger problem than failing to detect it when it occurs. In this case, the eye-tracker alone is a more appropriate predictor because students who need help will be more likely to get it..

### 3.3 Discussion and Future Work

In this paper, we have presented a study to ascertain whether using eye tracking information can increase the accuracy of a user model that needs to assess the meta-cognitive skill known as self-explanation. An alternative, easier to obtain source of evidence for this assessment is time per relevant interface action. In the study, we have collected data to compare the two sources.

Our results have shown that, in a laboratory setting, time is actually a much better predictor of self-explanation than expected. However, our data suggests that eye tracking data combined with time can increase the model bandwidth when a system that uses this model is mostly concerned with detecting the presence of self-explanation to avoid interfering with students who spontaneously self-explain. Furthermore, the eye-tracker alone may be more appropriate when the system priority is to detect when students do not self-explain. The data analysis also uncovered possible sources of inaccuracies in the data collection that may underestimate the value of eye tracker data.

Given these considerations, we plan to continue exploring the usage of eye tracker data with further experiments. One goal is to improve our data collection procedure to more reliably assess accuracy of eye-tracker data. A second goal is to collect data to test the addition to the ACE student model of nodes to represent evidence from both eye tracker and time. We plan to experiment by adding a naive Bayesian classifier structure. The advantage of this structure is that it is highly modular, allowing the eye-tracker and time data to be included or ignored as needed. In addition, the necessary conditional probabilities are readily available from sensitivity/ specificity frequencies in our data. We are also planning to perform the analysis described in this paper for the data collected on the other ACE units during the study. This will require extending the gaze detection algorithm to attention patterns relevant for those units.

## References

1. Alevan, V. and K.R. Koedinger, *An Effective Meta-Cognitive Strategy: Learning by Doing and by Explaining with a Computer-Based Cognitive Tutor*. Cognitive Science, 2002. **26**(2): 147-179.
2. Bunt, A. and C. Conati, *Probabilistic Student Modelling to Improve Exploratory Behaviour*. Journ of User Modeling and User-Adapted Interaction, 2003. **13**(3): 269-309.
3. Bunt, A., C. Conati, and K. Muldner. *Scaffolding self-explanation to improve learning in exploratory learning environments*. in *7th Int Conf on Intelligent Tutoring Systems*. 2004. Maceio, Brazil.
4. Chi, M.T.H., et al., *Self-explanations: How students study and use examples in learning to solve problems*. Cognitive Science, 1989. **15**: 145-182.
5. Conati, C. and K. VanLehn, *Toward Computer-based Support of Meta-cognitive Skills: A Computational Framework to Coach Self-Explanation*. Int Journ of Artificial Intelligence in Education, 2000. **11**.
6. Gluck, K.A. and J.R. Anderson, *What role do cognitive architectures play in intelligent tutoring systems?*, in *Cognition & Instruction: Twenty-five years of progress*, D. Klahr and S.M. Carver, Editors. 2001, Erlbaum. 227-262.
7. Iqbal, S.T. and B.P. Bailey. *Using Eye Gaze Patterns to Identify User Tasks (to appear)*. in *The Grace Hopper Celebration of Women in Computing*. 2004.
8. Jakob, R., *The Use of eye movements in human computer interaction techniques: what you look at is what you get*, in *Readings in Intelligent User Interfaces*, M.T.a.W. Maybury, W., Editor. 1998, Morgan Kaufmann Press: San Francisco. 65-83.
9. Just, M. and P. Carpenter, *The Psychology of Reading and Language Comprehension*, ed. A.a. Bacon. 1986, Boston.
10. Mitrovic, T. *Supporting Self-Explanation in a Data Normalization Tutor*. in *Supplementary Proc of AIED2003*. 2003.
11. Salvucci, D. and J. Anderson. *Intelligent Gaze-Added Interfaces*. in *SIGCHI conf on Human factors in computing systems*. 2000. The Hague, The Netherlands.
12. Shute, V.J. and R. Glaser, *A large-scale evaluation of an intelligent discovery world*. Interactive Learning Environments, 1990. **1**: 51-76.
13. Sibert, J.L., M. Gokturk, and R.A. Lavine. *The reading assistant: eye gaze triggered auditory prompting for reading remediation*. in *13th annual ACM symposium on User interface software and technolog*. 2000. San Diego, California: ACM Press.
14. Starker, I. and R.A. Bolt. *A Gaze-Responsive Self- Disclosing Display*. in *CHI: Human Factors in Computing Systems*. 1990. Seattle, WA: ACM.
15. VanLehn, K., *Student modeling*, in *Foundations of Intelligent Tutoring Systems*, M. Polson and J. Richardson, Editors. 1988, Lawrence Erlbaum Associates: Hillsdale, NJ. 55-78.