User-Adaptive Explanatory Program Visualization: Evaluation and Insights from Eye Movements

Tomasz D. Loboda and Peter Brusilovsky
School of Information Sciences, University of Pittsburgh, PA 15260, USA

April 08, 2010

Abstract. User-adaptive visualization and explanatory visualization have been suggested to increase educational effectiveness of program visualization. This paper presents an attempt to assess the value of these two approaches. The results of a controlled experiment indicate that explanatory visualization allows students to substantially increase the understanding of a new programming topic. Furthermore, an educational application that features explanatory visualization and employs a user model to track users' progress allows students to interact with a larger amount of material than an application which does not follow users' activity. However, no support for the difference in short-term knowledge gain between the two applications is found. Nevertheless, students admit that they prefer the version that estimates and visualizes their progress and adapts the learning content to their level of understanding. They also use the application's estimation to pace their work. The differences in eye movement patterns between the applications employing adaptive and non-adaptive explanatory visualizations are investigated as well. Gaze-based measures show that adaptive visualization captivates attention more than its non-personalized counterpart and is more interesting to students. Natural language explanations also accumulate a big portion of students' attention. Furthermore, the results indicate that working memory span can mediate the perception of adaptation. It is possible that user-adaptation in an educational context provides a different service to people with different mental processing capabilities.

Keywords: user-adaptation, program visualization, explanatory visualization, eye movements, eye tracking, evaluation, user study, working memory

1. Introduction

Software visualization (SV) is considered to be one of the most important educational tools in Computer Science and Information Science education (Naps et al., 2002; Naps et al., 2003). Researchers distinguish two subfields of SV that focus primarily on education: program visualization and algorithm visualization (Price, 1993). Dynamic approaches to visualization (also called animation) are among the most popular in both subfields. Program visualization focuses on dynamic visualization of programming constructs while algorithm visualization focuses on dynamic visualization of higher level software descriptions (Kerren and Stasko, 2002). Both types of SV can provide a clear visual metaphor for

complicated concepts and uncover the dynamics of important processes that are otherwise hidden.

For many years, the focus of SV research has been upon developing better tools and exploring new application contexts. Relatively few studies investigated the effectiveness of SV because the benefits of visualization were taken for granted by researchers and teachers alike. Yet, several studies have found the educational benefits associated with observing SV to be low (e.g., Byrne et al., 1999, Stasko et al., 1993). The presence of a well-developed visualization failed to help students understand the mechanics of a computer program or an algorithm. Those findings motivated the research on making SV more effective educationally. Three main approaches to that problem that have been explored so far are engaging, explanatory, and user-adaptive SV.

The most popular approach to increasing the effectiveness of SV is *engaging SV*. The idea behind engaging visualization (Naps et al., 2000) is to change students from passive observers to active learners by engaging them in some activity related to the visualization. A whole spectrum of activities has been explored. On one side of this spectrum are relatively low-engagement activities, such as asking students to provide their own data for algorithm animation (e.g., use student-defined array in array sorting visualization instead of a predefined array). On the other side of the spectrum are high-engagement activities, such as requiring students to construct entire visualization themselves instead of watching a prepared one (Hundhausen et al., 2002). Between these extremes is a range of medium-engagement activities, such as having students predict results of a program execution (Jarc et al., 2000; Naps et al., 2002), or simulate the behavior of an algorithm on a stepwise basis (Krebs et al., 2005). Most of these innovations showed positive effects (e.g., Byrne et al., 1999; Hundhausen et al., 2002; Jarc et al., 2000; Naps et al., 2002).

The two other alternatives, *explanatory visualization* and *adaptive visualization*, have been explored to a far lesser degree. The idea behind explanatory SV is to augment visualization with natural language explanations. These explanations should facilitate understanding by providing narration relevant to what is taking place in the visualization. The need to supplement SV with explanations was first expressed by Brusilovsky (1993) and Stasko et al. (1993). Brusilovsky (1994) provided evidence that explanations can help students understand what they see and Nevalainen and Sajaniemi (2006) demonstrated that users follow explanations more attentively than the visualization itself. Currently, explanatory visualization is used in a number of SV projects (e.g., Blumenktants et al., 2006; Brusilovsky and Spring, 2004; Lahtinen and Ahoniemi, 2007; Yamamoto and Hirose, 2005; Kerren et al., 2006).

While the majority of those projects use human-authored explanations, model-based dynamic generation of explanations has been investigated as well (e.g., Dancik and Kumar, 2003; Kumar, 2003). User studies demonstrated that both human- and computer-generated explanations are effective in improving student understanding and learning.

Adaptive SV is based on the premise that different students may have different knowledge of programming constructs or parts of algorithm that are being visualized. Proponents of adaptive visualization argue that presenting visualization on the same level of detail to all students is inherently inefficient; students with more knowledge may get bored, while students with less knowledge may not get enough details to understand the visualization (Brusilovsky, 1993; Kerren et al., 2006). To avoid this potential conflict scenario, the level of detail of visualization related to each programming construct or an algorithm's aspect is matched with the student's level of knowledge about it. The lower the student's level of understanding, the greater the level of detail. This approach is motivated by a stream of work on adaptive presentation. Boyle and Encarnacion (1994) and Kobsa et al. (2001) have shown that by adapting the level of explanation to user knowledge of a subject (i.e., providing additional explanations to novice users, while offering highly specific details to expert users) can result in faster comprehension and decreased error rate. Those prior good results could be transplantable to the visualization ground. In a preliminary investigation involving a simple mini-language, Brusilovsky (1993) obtained optimistic results showing that adaptive program visualization could improve understanding. More recently, Brusilovsky and Su (2002) developed the WADEIn application for adaptive visualization of the C programming language expression evaluation. Several classroom studies of WADEIn showed positive results; more than 80% of students found WADEIn and its adaptive visualizations helpful or very helpful.

Because the three approaches to SV discussed above are complementary in nature we have attempted to combine them. In order to do that, we have developed the cWADEIn Web-based educational application (Brusilovsky and Loboda, 2006), which integrated user-adaptive and explanatory visualization with elements of engaging visualization. Like its predecessor (WADEIn), cWADEIn focuses on visualization of expression evaluation. Visual expression evaluation can be considered a special case of program visualization. Visualization of expression evaluation is important for understanding complex expressions in languages such as C, C++, or Java and has been implemented in several program visualization tools (e.g., Moreno et al., 2004; Kumar, 2005). In cWADEIn, both the content of visualization and explanations are adapted to the level of the student's knowledge.

To examine the value of adaptive explanatory visualization, several semester-long classroom evaluations of cWADEIn were run. While student feedback collected in these studies provided some evidence in favor of the educational benefits of the application, the format of a classroom study was not sufficient to reliably assess the value of adaptive explanatory visualization. This paper presents the results of an experiment that allowed us to address that problem. We attempted to collect more reliable evidence of the application's educational impact, and to assess the added value of user-adaptation (referred to simply as adaptation in the reminder of the paper) in the context of explanatory program visualization. We found that cWADEIn helped students to greatly improve their understanding of expression evaluation which suggests that explanatory visualization may be a desirable educational aid. However, we did not observe the effect of adaptation on the learning outcome. We argue that this may be due to a ceiling effect. Nevertheless, adaptation helped students to explore substantially more expressions. Additionally, students preferred the adaptive version and used its estimation of the progress they were making to pace their work.

An important component of our study was eye tracking which allowed us to investigate the effect of adaptation on eye movements of students and inspect how they allocated their attention. We found that adaptation rendered visualization more interesting to students and captivated bigger chunk of their attention than did the non-adaptive visualization. Explanations accumulated a major portion of viewing time as well which signifies their importance. Additionally, working memory capacity seemed to have had a mediating effect on the perception of adaption.

Several prior research endeavors show that eye tracking can serve as a solid research tools in the areas of user-adaptive tools and SV. Conati and Merten (2007) used eye-tracking data for on-line assessment of user meta-cognitive behavior in an adaptive educational application. Bednarik et al. (2005) used it to discover differences between novice and intermediate users working with program visualization. Nevalainen and Sajaniemi (2006) explored the role of textual information in a visualization tool. The current paper adds on to the body of work employing eye tracking in user modeling and SV. The work presented here is an extension of work discussed by Loboda and Brusilovsky (2008).

In what follows, we first introduce the cWADEIn application (Section 2). Then, we describe the experiment we ran (Section 3) and report our findings (Section 4). Next, we discuss the results we obtained and talk about limitation of this discourse (Section 5). Finally, we present some concluding remarks (Section 6).

2. An Adaptive Educational Application

cWADEIn[1] (Brusilovsky and Loboda, 2006) is a Web application (a Java applet) for students in introductory C programming language oriented courses. It addresses the problem of explaining the evaluation of expressions. This problem is both relatively complex and rarely addressed by visualization tools. cWADEIn supports twenty four operators covering simple arithmetic, comparison, increment/decrement, logic, and assignment operations, but excluding pointer arithmetic. The application tracks the progress of the student and employs adaptive visualization and adaptive textual explanations.

The user interface of cWADEIn (Figures 1 and 2) has been divided into four regions: Goals and Progress, Settings, Navigation, and Blackboard. The Goals and Progress region contains a list of explicit concepts for the current topic (e.g., assignment operators), along with progress indicators which allow students to monitor their progress. The Settings region allows students to select the expression to be evaluated and set the initial values of variables. The student can select the expression from a list of predefined expressions or (in the spirit of engaging visualization) provide an expression to be visualized and explained. The Navigation region allows users to control the evaluation process: proceed on a step-by-step or operator-by-operator basis, both forward or backward. Finally, the Blackboard region is where the expression evaluation visualization and explanations are shown.

cWADEIn features two mutually exclusive modes: Exploration and Evaluation. In the Exploration mode (Figure 1), the student can step-through the evaluation, observe the visualizations, and read the associated explanations. The primary purpose of the Exploration mode is to help students understand the material, but it can also be used by teachers during classroom demonstrations.

In the Evaluation mode (Figure 2), the student is asked questions at every step of the evaluation. First, the student is asked to specify the order of evaluation, which checks the knowledge of operators precedence. After that the application asks the student to predict the result of the evaluation of each operation. Evaluation mode challenges and engages students and allows a better assessment of the state of their knowledge. In this mode, the application uses only as much visualization as is necessary to clarify questions asked to the student, e.g., illuminating the current operation. No explanations are presented in that mode.

[1] The previous name of the application was WADEIn II. It has been renamed to cWADEIn since a Java version, named jWADEIn, has been developed.

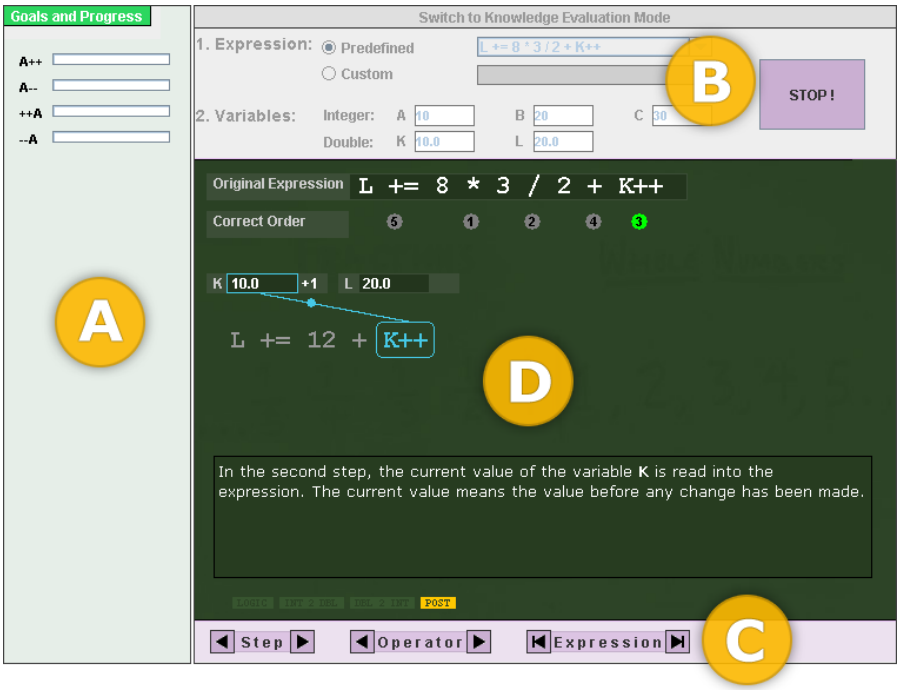

Figure 1. A screenshot of the cWADEIn application while in the Exploration mode. The interface is organized into four regions: Goals and Progress (A), Settings (B), Navigation (C), and Blackboard (D).

Again, from a research vantage, cWADEIn has been created to investigate the blend of adaptive visualization and adaptive textual explanations.

2.1. Explanatory Visualization

In cWADEIn, evaluation (and therefore visualization) of each expression is broken down into the evaluation of individual operations:

$$\underbrace{B = 1 + (\overbrace{A \;\%\; 3}^{operation} \times 2) - (0 \;||\; 4.18) \times C + +}_{expression} .$$

The Blackboard region (Figure 1, D), that is used to present visualization and explanations, can be divided into three subregions: vis–s, vis–d, and txt (Figure 5). The vis–s subregion denotes the static part of visualization and spans the area showing the expression being evaluated in its original form (never changes during the evaluation) along with

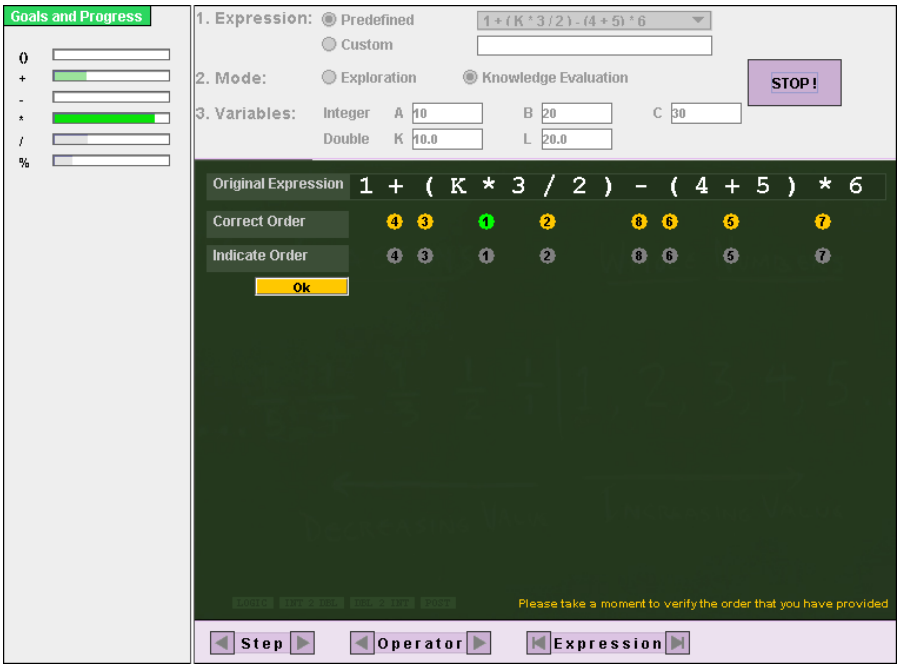

Figure 2. A screenshot of the cWADEIn application while in the Evaluation mode. Evaluation mode was not used in the experiment and is presented here only to complement the description of the application.

the indication of the operator being evaluated at the moment. Changes to the content presented in that subregion are limited to illuminating the current operator.

The vis–d subregion denotes the dynamic part of visualization and spans values of variables participating in the expression (if any), the expression as it looks at any given point of evaluation,[2] and animations related to the evaluation of all operators.

Some visualizations may be difficult to understand when presented on their own. cWADEIn tries to address that problem by adding natural language explanations associated with most of visual events. Those explanations are displayed in the txt subregion.

cWADEIn color-codes different aspects of visualization. For example, green is used exclusively to mark the current operation.

[2] At the very end of every evaluation, after all operators have been evaluated, the expression is collapsed to a single number.

2.2. Adaptation

cWADEIn models the student's progress on two levels: explicit and implicit concepts. The application visualizes the progress the student makes with explicit concepts. The progress made with implicit concepts is tracked, but not visualized. All operators are modeled as explicit concepts. Implicit concepts are: (1) reading a variable, (2) implicit casting, and (3) logical value representation.

The application adapts the speed of animations to the progress the student has done so far with each operator. The more progress the higher the pace of animations. Eventually, animations are collapsed into single-step events.

Each explanation is constructed from one or more passages of text. Each passage is associated with one concept (explicit or implicit) and addresses a different idea. The application chooses to present a given passage based on the evidence of the student's progress. A passage is judged to be relevant for the student, and therefore displayed, until they reach a certain level of knowledge. Eventually, no explanations are shown.

cWADEIn associates two levels of knowledge with each concept: *exploration knowledge* (k_{ex}) and *evaluation knowledge* (k_{ev}). These two levels represent the student's progress in the two modes the application can work in. Additionally, each concept has a complexity assigned to it, which allows the application to grant more credit towards mastering easier concepts, while treating more difficult ones as requiring more effort on the part of the student.

Five types of student activities, originating in the two modes, are used as evidence of progress:

- exO: The student is presented with the order of evaluation.
- exV: The student is presented with a visualization relevant to a given concept or set of concepts.
- evO: The student indicates the order of evaluation.
- $ev+$: The student provides the correct value of an operation.
- $ev-$: The student provides an incorrect value of an operation.

The occurrences of those activities are counted (n) and aggregated to yield the two levels of knowledge using the following formulas

$$k_{ex,i} = \frac{g_{exO}\, n_{exO,i} + g_{exV}\, n_{exV,i} - l_{ex}\, n_{ev-,i}}{\sqrt{c_i}} \,,$$

$$k_{ev,i} = \frac{g_{evO}\, n_{evO,i} + g_{ev}\, n_{ev+,i} - l_{ev}\, n_{ev-,i}}{\sqrt{c_i}} \,,$$

where i is the index of the concept, g is the knowledge gain parameter, l is the knowledge loss parameter, n is the number of times a particular activity occurred, and c is the complexity of the concept. Levels of both types of knowledge can range from 0 to 5. The g and l parameters define the model's sensitivity to activity of the student. Higher values of g and lower values of l cause the model to reach the upper bound quicker. cWADEIn uses the same values of those parameters for all students, but in principle, they could be adjusted to account for varying degrees of learning potential different students possess. The values used in cWADEIn reflect our own judgement and satisfy rules like: animations and explanations related to the easiest operators are shown no more than four times, roughly twice more knowledge is attributed for correctly answering a question than there is for only seeing an evaluation, etc. They are given below

$$k_{ex,i} = \frac{0.15\, n_{exO,i} + 1.6\, n_{exV,i} - 0.6\, n_{ev-,i}}{\sqrt{c_i}} \ ,$$

$$k_{ev,i} = \frac{0.25\, n_{evO,i} + 2.4\, n_{ev+,i} - 1.6\, n_{ev-,i}}{\sqrt{c_i}} \ .$$

Note, that since the term n_{ev-} appears in the k_{ex} formula, activity of the student in the Evaluation mode influences the application's adaptive behavior in the Exploration mode.

The two knowledge levels are represented differently on the progress indicators. The exploration knowledge is shown as the length of the progress bar. The evaluation knowledge is indicated by the intensity of the color of the bar; the higher the level of knowledge the more intense the color. This discrimination helps the student to identify operators they did less work with and to decide when to switch to the Evaluation mode, where they can check their understanding.

3. Experiment

3.1. Stimulus

For the purpose of the experiment, cWADEIn could be launched with adaptive mechanisms enabled or disabled. As described in Section 2, the adaptive version attempted to tailor its behavior to the student's progress. The non-adaptive version did not alter its behavior in any way; animations were always played using the same speed and fragments of explanations were never hidden. Additionally, the progress indicators did not show the student's progress. They were still displayed, but only as a reminder of current learning goal (i.e. concepts

to be mastered). Only the Exploration mode was employed in the experiment.

3.2. Subjects

Fifteen students with normal or corrected to normal vision were recruited at the University of Pittsburgh. The only graduate student was subsequently excluded from the analysis as an outlier – their gain score (posttest-pretest difference) was more than three standard deviations below the mean gain score in both of the two learning trials (i.e. they understood very little if anything at all). Cook's distance[3] for that subject was > 0.8 providing further support for exclusion. Eight of the retained fourteen subjects were female and the age range was 18–25 ($M = 20.5$, $SD = 1.7$).

The majority of the regular users of cWADEIn come from a Data Structures and Programming Techniques course at the School of Information Sciences, University of Pittsburgh. The school receives applications from people with different backgrounds and no assumptions about the level of technical proficiency can be made. Because of that, the two following recruitment criteria were enforced: (a) none of the subjects could be a student in the Computer Science program and (b) all subjects could have at most *very limited* self-assessed programming skills (i.e., at most 1 on a scale of 0-5). Effectively, 11 subjects never programmed before and three had very limited programming experience.

3.3. Protocol

Figure 3 shows the timeline of the experiment. After filling in the entry questionnaire and performing the modified digit span (MODS) task (see Appendix A), each subject was given an introduction to the study and a short training to minimize the effect of the unfamiliarity with the application. During the training, subjects were asked to attend the evaluation of several expressions to know what to expect with regard to visualization and explanations. Ten expressions with three simple arithmetics operators ($+$, $-$, and $*$) were used. Subjects were free to finish the training when they felt ready, but not before they attended to the first three expressions and at least one of the more structurally complex ones.

After training, subjects were asked to perform two 15-minute learning trials using cWADEIn – one with the adaptive version (experi-

[3] Cook's distance (D) is an influence statistic estimating the effect of the deletion of an observation on the parameter estimates.

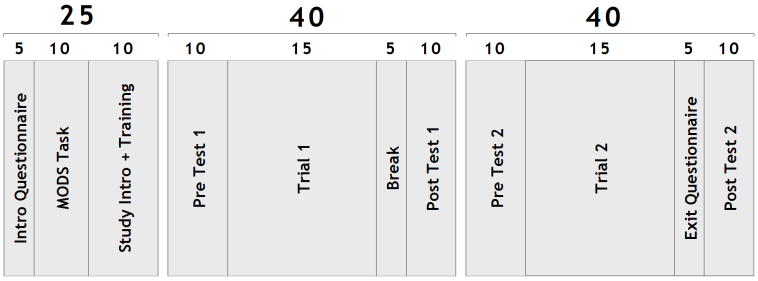

Figure 3. The timeline of the experiment (time shown in minutes).

mental condition) and the other with the non-adaptive version (control condition).[4] Each trial was framed as preparation for an in-class test on the C programming language expressions. Subjects were made aware of the dual nature of the problem involving semantics and precedence of the operators. Six subjects completed their first trial using the non-adaptive version, with the other eight starting with the adaptive version, to control for carryover effects. A total of twelve operations from the twenty four supported by cWADEIn were used in the experiment. The selected operations were divided into two sets to accommodate the two trials (Table I). The operator sets were designed to be equal in overall difficulty and orthogonal with respect to knowledge transfer. For instance, assignment operators were cumulated in one set; distributing them between the two sets would increase the likelihood of inter-trial dependency. Thirty expressions were associated with each operation set. They were available as elements of a drop down list and ordered by difficulty, with the easier ones at the top. Subjects were aware of the ordering but were not forced to follow it.

Table I. Operators used in both sets.

Set	Group	Operator
1	comparison	<, >=, !=
	modulo	%
	increment	++A, A−−
2	parenthesis	()
	assignment	=, +=, *=
	logical	||, &&

[4] Initially, we scheduled the trials to be 20-minute long, but after a few pilot subjects we cut them five minutes shorter.

Prior to starting each learning task, subjects were given a pretest. After the task was finished subjects had to take a break of an approximate length of five minutes. That time lag was introduced to control for the recency effect. In the case of the first trial, the break was a scheduled break in the experiment. In the case of the second trial, the break was in a form of the exit questionnaire. A posttest was administered after the break. The corresponding questions on pretest and posttest checked the understanding of the same operator. The tests were designed not to give away the answers. Both the semantics and the precedence of operators were tested, with a greater emphasis on the semantics.

Apart from questionnaire and test responses, user interface events (Section 3.4) and eye movement (Section 3.5) protocols were collected. The eye tracker calibration routines were part of the experiment, but constituted only a minor portion of the experiment time (about five minutes). Calibration immediately preceded each learning trial. The whole experiment took between 1.5 and 2 hours. That variation was due to the difference in the time it took subjects to solve the tests and fill out the questionnaires (which were not time constrained) and the fact that subjects could finish both trials when they felt ready (i.e. before 15 minutes passed). In fact, 11 out of all 28 sessions were finished earlier by subjects themselves.

3.4. Interaction Logs

While subjects were performing their task, cWADEIn was logging all user interface events, e.g. pressing a button, opening a drop-down list, selecting an item from it. Additionally, all changes to the user interface that were not caused by subjects were also logged, e.g., the time when an animation terminated. We used those logged events information to investigate material exposition, as described in Section 4.2.

3.5. Eye Movements

A Tobii 1750[5] remote eye tracker was used to display the stimulus at a resolution of 800x600 pixels and to collect subjects' eye movement data. The experiment took place in a usability laboratory with a constant ambient light. Subjects were sited to maintain an approximate viewing distance of 60 cm. A nine-point eye tracker calibration routine immediately preceded each trial. Because the eye tracker utilizes infrared diodes and camera, a ball mouse was used instead of an optic one to avoid potential interference.

[5] http://www.tobii.com

Eye movement signals were sampled at 50 Hz yielding a temporal resolution of 20 ms. We used the ClearView 2.6.3 software provided with the eye tracker to transform raw gaze samples into fixation data. The fixation identification algorithm implemented in ClearView is parameterized by the spacial and temporal thresholds: *fixation radius* and *minimum fixation duration*. The algorithm declares two points belonging to the same fixation if the distance between them was not more than the fixation radius and the second one occurred within the minimum fixation duration from the first one. Only data points for which information about both eyes were available were used in the analysis (ClearView's validity filter set to "high"). Overall, we analyzed nearly seven hours of eye movement recordings.

To look at the mental processing bearing fixations only we filtered out those shorter than 300 ms. Therefore, overall, we analyzed two eye movement data sets, based on two minimum fixation duration thresholds: 100 and 300 ms. Fixations longer than 1200 ms (1.26% of all fixations) were excluded from the analysis. Figure 4 shows a histogram of fixation durations. The distribution has an anticipated positively skewed shape. The 300 ms threshold is marked on the histogram. Roughly two thirds of all fixations are filtered out when only long fixations are considered (the sum of percentages over the first two bars).

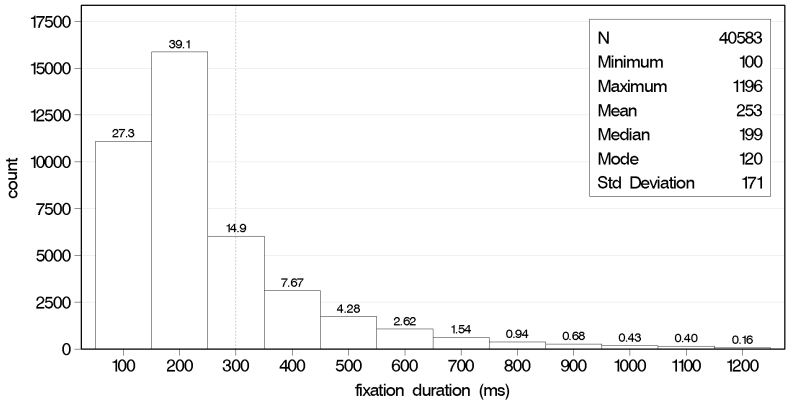

Figure 4. Histogram of fixation durations (percentages over the bars; bins of 100 ms).

An in-depth discussion of the choices we made and procedures we employed surrounding the fixation radius and duration thresholds can be found in Appendices B and C.

3.5.1. *Stimuli and Regions of Interest*
Two distinct general application usage patterns can be isolated in the recordings of all trials: *expression selection* and *expression evaluation*. Each subject would select an expression of interest and go through its evaluations, then select another expression, and so on. For the purpose of the eye movement data analysis we treated those two patterns as two separate stimuli. We divided each of them into several regions of interest (ROIs; Figure 5).

The expression selection stimulus (select) had two ROIs: one for the list of operators and progress indicators (op) and one for the list of available expressions (expr). Such organization of the user interface allowed us to check if subjects used the application's indication of their progress, especially while choosing their next expression (progress indication was available only in the adaptive version). The expression evaluation stimulus (eval) had four ROIs: one for the list of operators and progress indicators (op), one for the static part of visualization (vis–s), one for the dynamic part of visualization (vis–d), and one for the textual explanations (txt).

Defining ROIs too close to one another may cause misclassifications of gaze location samples, and consequently, resulting fixations. To avoid this problem we used buffer zones. Any two ROIs were at least 30 pixels apart, a value suggested by the manual of the eye tracker.

Please note, that the screenshots of the application shown in Figure 1 and 5 differ slightly. Figure 1 depicts the application as it looks "on a regular day". Figure 5 depicts the application customized for our experiment and therefore as seen by the subjects. The difference is that a few user interface controls were hidden during the experiment. The rationale behind it was to keep the interface as clean as possible and restrict the number of controls to the set that would be used by the subjects. That was especially important because of eye tracking. We did not want the subjects to focus on controls that presented no value for the task they faced. For instance, cWADEIn allows its users to enter and explore an arbitrary expression. Since we did not want our subjects to enter their own expressions, showing the controls that allowed that would only serve to confuse.

Figure 6 depicts an example of eye movements recorded during an evaluation of one of the expressions for one of the subjects.

3.5.2. *Measures*
To investigate the differences in the patterns of visual usage of the two versions of the application we looked at four measures. The first one was the *fixations rate* (*FR*), i.e. the number of fixations controlling for the exposition factor. The second measure was the *average fixation*

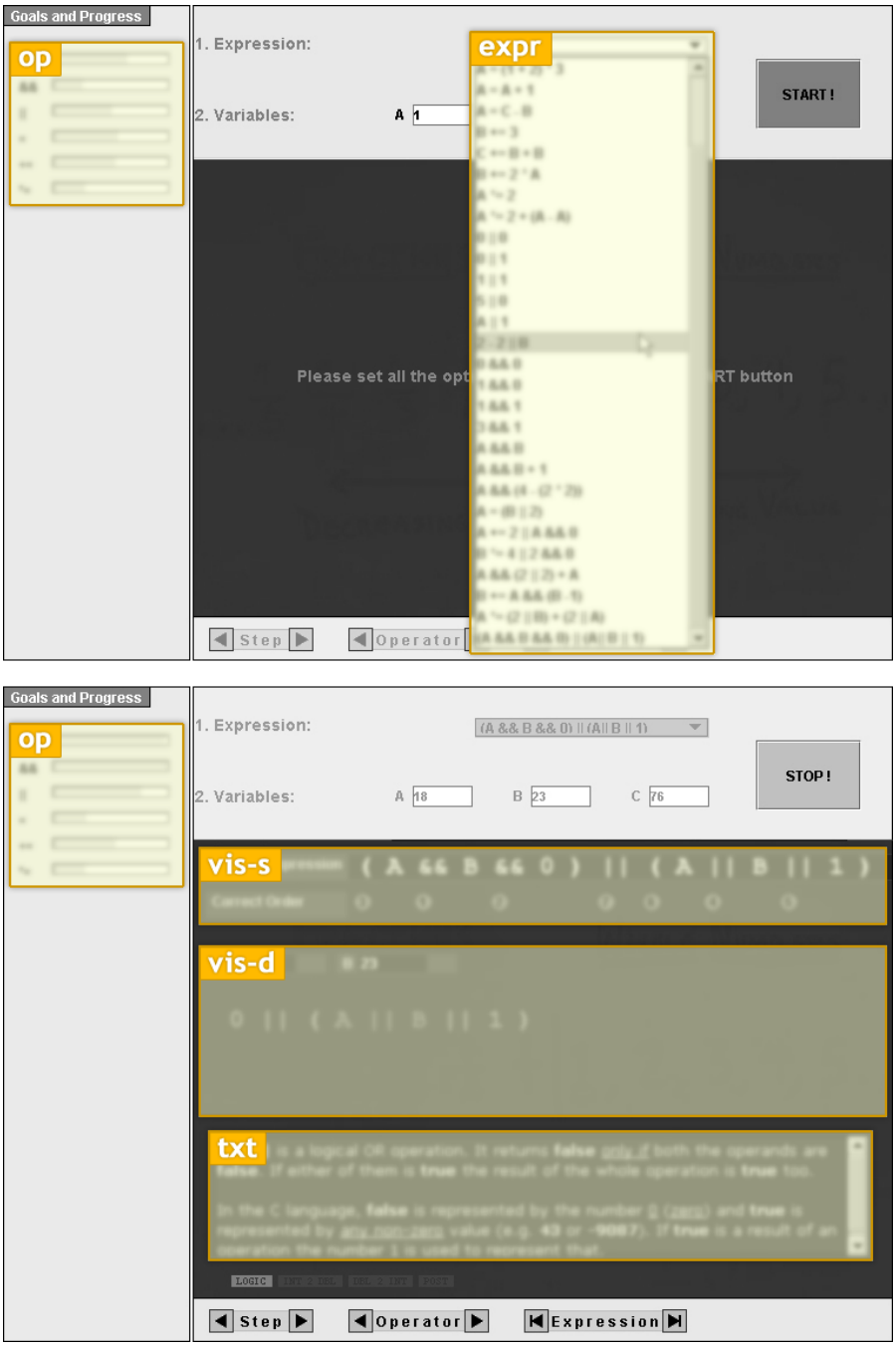

Figure 5. On the top are the regions of interest of the select stimulus: list of operators (op) and list of available expressions (expr). On the bottom are the regions of interest of the eval stimulus: list of operators (op), static visualization (vis–s), dynamic visualization (vis–d), and textual explanation (txt).

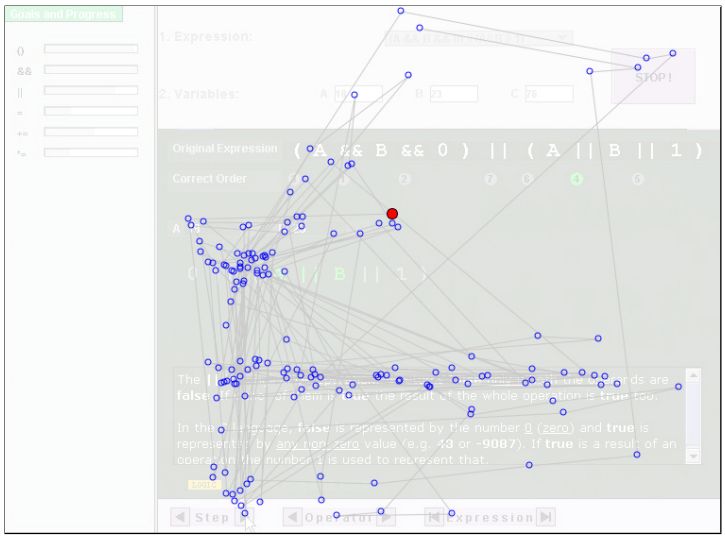

Figure 6. An example of eye movements recorded during an evaluation of one of the expressions for one of the subjects. The first fixation is marked with a big red dot.

duration (*AFD*) defined as an arithmetic mean of all fixation durations. The third measure was the *gaze time proportion* (*GTP*), i.e. the sum of fixation durations (on a given ROI) divided by the total gaze time (for all ROIs together). The fourth measure was the number of *gaze transitions* (*GT*) between two ROIs, irrespective of the direction of those transitions.

We used different exposition variables for different ROIs. More detailed discussion of the eye movement analyses is given in Section 4.3.

4. Results

In this section, we present the results of our analyses of the knowledge gain (Section 4.1), material exposition (Section 4.2), eye-movement patterns (Section 4.3), and subjective responses (Section 4.4). Below, we refer to the non-adaptive and adaptive versions of cWADEIn as control and experimental, respectively.

We enforced an alpha level of .05 in all statistical tests. We report the exact two-tailed p-values unless they are smaller than .001. All results were obtained using SAS System version 9.2 (SAS Institute Inc., 2008) and R version 2.10.1 (R Development Core Team, 2009). We discuss the issues of multiplicity adjustment and model selection and fit in the Appendix D.

4.1. Pretest-Posttest Difference

The primary response variable in our experiment was the difference between the posttest and pretest scores, that we refer to as the *gain score* and denote as δ. We assumed it to be normally distributed. The two independent variables were the application version and the trial order. Both were dichotomous and within-subject. We used the *working memory index* ω (see Appendix A) as a between-subject covariate ($\rho_{\delta,\omega} = 0.36$).

As mentioned in Section 3.3, the version was varied across trials in a counterbalanced fashion with subjects assigned randomly to the two possible orderings. In order to check if that assignment generated equivalent groups we used a paired t-test to compare the pretest scores. We found no significant difference between the mean score of the first group ($M = 2.57$, $SD = 2.24$) and the second group ($M = 4.57$, $SD = 3.13$), $t(13) = .48$, $p = .642$, $\hat{g} = .24$.

First, we looked at the difference between the posttest and pretest scores to assess if cWADEIn helped subjects in improving their understanding of the problem of expression evaluation. The results of a paired t-test indicate that they got a significantly higher score on the posttest ($M = 21.54$, $SD = 5.46$) than they did on the pretest ($M = 3.57$, $SD = 2.86$), $t(27) = 19.73$, $p < .001$, $\hat{g} = 8.67$. This six-fold increase in the test score looks quite remarkable given that the subjects used the application for no more than 15 minutes. If explanatory visualization was responsible for this difference then it would be a rather strong case in favor of the educational value of that feature.

To test the effect of both factors on the gain score we fitted the following linear mixed model (LMM)

$$y_{ijk} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij} + \omega_k + b_{ik} + e_{ijk} \ ,$$

where μ is the overall mean, α_i is the effect of application i, β_j is the effect of trial j, ω_k is the working memory index of subject k, b_{ij} is the random effect of subject k assigned to application i, and e_{ijk} is the random measurement error. A studentized conditional residuals diagnostic panel indicated a good fit of the model (Figure 7). The model above is the most parsimonious one; all higher order terms were not significant.

Working memory index was a significant covariate, $F(1, 23) = 5.72$, $p = .025$. The trial order explained a significant amount of variability in the gain score. Subjects achieved higher gain scores on the second trial ($M = 20.43$, $SD = 4.18$) than they did on the first one ($M = 15.50$, $SD = 4.20$), $F(1, 23) = 12.59$, $p = .002$. There was no difference between the mean gain score achieved by subjects with the experimental

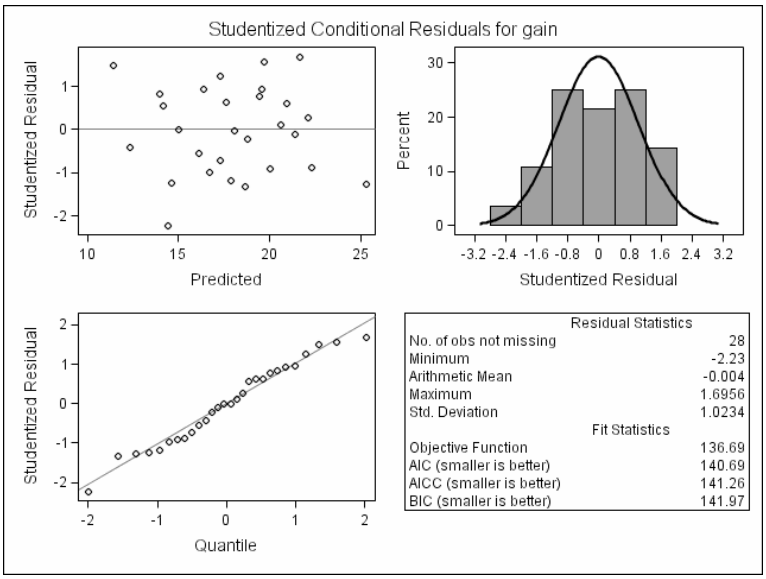

Figure 7. Gain score model fit assessment. No apparent patterns in the residuals cloud. The shape of the distribution of the residuals given random effects reasonably well follows the shape of the normal distribution.

($M = 17.35$, $SD = 5.05$) as compared to the control version of the application ($M = 18.57$, $SD = 4.68$), $F(1, 23) = 1.84$, $p = .195$.

If the difference between the two versions of application existed it might have been masked by the ceiling effect. It is possible that 15 minutes was still too much for the learning task in the case of our quite simple domain. Some evidence of the ceiling effect comes from the fact that 11 out of the total of 28 trials were finished before the allotted time by subjects themselves (the shortest session being 12 minutes). Five of those 11 sessions originated in the control condition while the remaining six in the experimental one. Therefore, it is unlikely that it was the version of the application that was responsible for subjects finishing earlier. Because it seemed plausible that subjects who left the study early were those who knew more at the beginning, we looked at the pretest scores. Indeed, a directional paired t-test revealed that the average pre-test score of those subjects who ended at least one session before time ($M = 4.36$, $SD = 3.32$) was higher than those who did not finish either of their sessions early ($M = 2.79$, $SD = 2.15$), $t(13) = -1.89$, $p = .042$, $\hat{g} = .54$.[6]

[6] Two-tail purists should understand this difference as leaning towards significance with $p = 0.083$.

Additional evidence for the ceiling effect stems from the fact that subjects explored significantly more expressions in the experimental version (next subsection).

4.2. Material Exposition

In addition to investigating the gain score difference, we checked the amount of material that subjects were exposed to. For that purpose, we used the interaction logs collected during the experiment. If we treat the events of exploring the evaluation of an expression as independent, their total number will be Poisson distributed. We compared rates instead of total numbers to control for the variation in the trial time.

We fitted a generalized LMM (GLMM) parameterized as

$$\log y_{ijk} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij} + \omega_k + \log \tau_{jk} + b_{ik} + e_{ijk} \ ,$$

where μ is the overall mean, α_i is the effect of application i, β_j is the effect of trial j, ω_k is working memory index of subject k, τ_{jk} is the session time for subject k in trial j and entered the model as an offset variable, b_{ik} is the random effect of subject k assigned to application i, and e_{ijk} is the random measurement error. Studentized conditional residual plots showed a good fit of the model. The variance of the Pearson residuals was 0.71 indicating no problems with overdispersion. That was the most parsimonious model.

Figure 8 shows a cumulative graph of the number of expressions explored by subjects in both versions of the application. Subjects working with the experimental version of the application explored expressions at a significantly higher rate ($M = 1.94$ per min., $SD = .46$) as compared to the control version ($M = 1.53$ per min., $SD = .68$), $F(1, 24) = 7.65$, $p = .010$. Subjects were also exploring expressions significantly faster in the second trial ($M = 2.03$ per min., $SD = .69$) than they did in the first one ($M = 1.44$ per min., $SD = .33$), $F(1, 24) = 13.66$, $p = .001$. The effect of working memory index ω was not reliable, $F(1, 14.21) = 3.31$, $p = .090$. This result shows that adaptation can lead to an increase in material exposition and therefore be a desirable feature of an educational tool.

4.3. Eye Movements

In this section we present our investigation of the effect of the application version (app) and the dichotomized (median split) working memory index (wm) on eye movement variables. Table II shows exposition (offset) variables we associated with responses on all ROIs. Those variables were used when fitting count and proportion models we describe below.

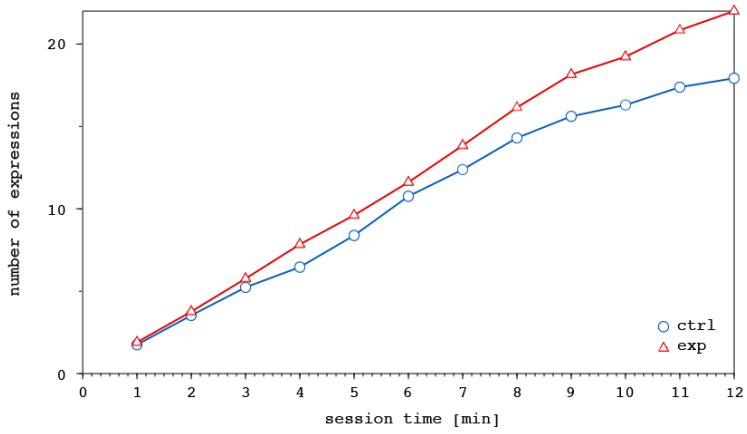

Figure 8. Number of expressions explored (cumulative) for the experimental and the control versions of the application. Sessions were as long as 15 minutes, but the first early-finisher subject exited the task after 12 minutes.

Table II. Exposition variables associated with responses on all ROIs.

Stim.	ROI	Exposition variable
select	op	Number of expressions explored (expr-cnt)
	expr	Number of expressions explored (expr-cnt)
eval	op	Session time (sess-t)
	vis-s	Total gaze time (sum of all fixation durations; gt-tot)
	vis-d	Total duration of animations (anim-dur)
	txt	Total length of textual explanations (expl-len)

4.3.1. *Aggregate Measures*

Figure 9 shows box plots for FC, GT, and FD per ROI. One of the most evident things that this figure shows is the large amount of time the subjects spent on dynamic visualization (vis-d) and explanations (txt). We postpone further interpretation of data presented on that figure until the discussion (Section 5).

To analyze FRs we fitted the following GLMM assuming a Poisson distribution of the response

$$\log y_{ijk} = \mu + \alpha_i + \omega_k + \alpha\omega_{ik} + \log \tau_{jk} + b_{ik} + e_{ijk} \ ,$$

where μ is the overall mean, α_i is the effect of application i, ω_k is working memory of subject k, τ_{jk} is the exposition variable for subject k in trial j (Table II), b_{ik} is the random effect of subject k assigned to application i, and e_{ijk} is the random measurement error.

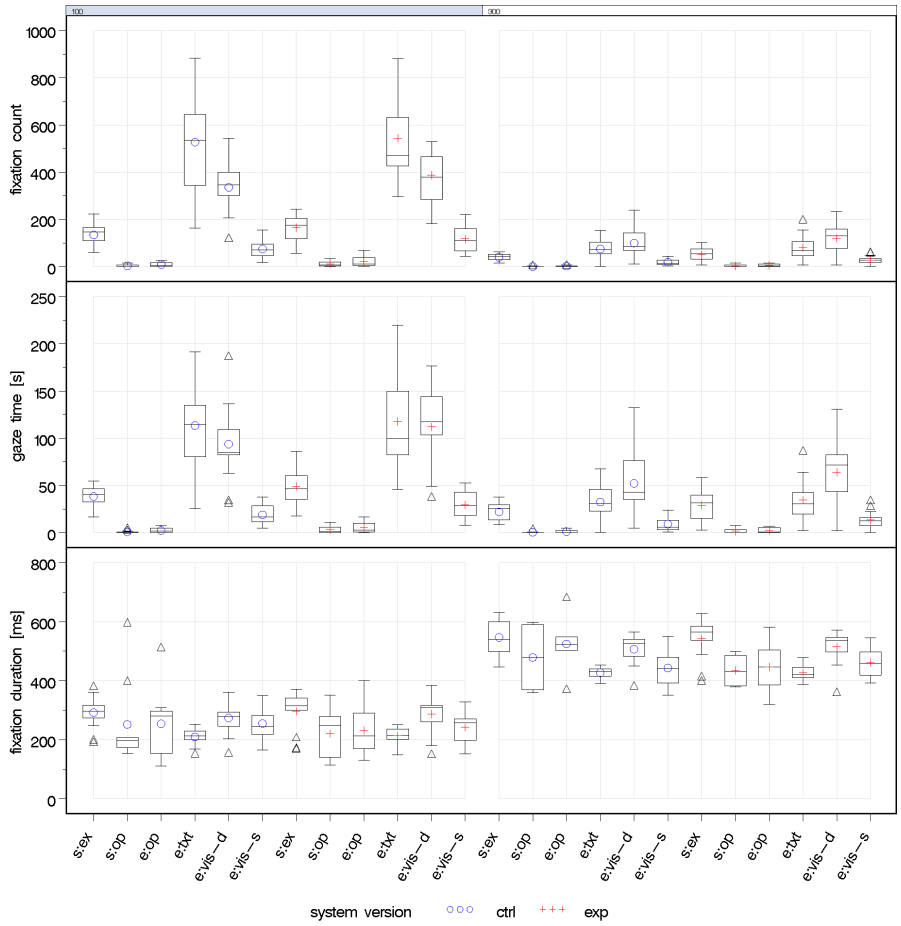

Figure 9. Box plots for fixation count, gaze time, and fixation duration per ROI for the two minimum fixation duration thresholds and the two application versions. Names of ROIs (horizontal axis) are prefixed with the first letter of a stimulus: s for select and e for eval.

The outcomes of gaze time are not binomial proportions, because they do not represent the ratio of a count over a total number of Bernoulli trials. However, because the proportion of gaze time spent on a given ROI must lie in the interval $[0, 1]$, we can proceed with the analysis with a model that treats GTP as a "pseudo-binomial" variable (see McCullagh and Nelder, 1989). We fitted the following GLMM

$$\text{logit}(p_{ijk}) = \mu + \alpha_i + \omega_k + \alpha\omega_{ik} + b_{ik} + e_{ijk} \ ,$$

where p_{ijk} is the probability of a subject looking at the ROI under consideration, μ is the overall mean, α_i is the effect of application i, ω_k

is working memory of subject k, b_{ik} is the random effect of subject k assigned to application i, and e_{ijk} is the random measurement error.

We assumed the AFD to be normally distributed and to analyze it we fitted the following LMM

$$y_{ijk} = \mu + \alpha_i + \omega_k + \alpha\omega_{ik} + b_{ik} + e_{ijk} \ ,$$

where μ is the overall mean, α_i is the effect of application i, ω_k is working memory of subject k, b_{ik} is the random effect of subject k assigned to application i, and e_{ijk} is the random measurement error.

Below, we describe the aggregate eye movement measures findings reported in the top part of Table 4.3.1. We do that ROI by ROI in the order in which ROIs are listed in the table (and in Figure 5). The vis-d and txt ROIs are the most important ones because they correspond to dynamic visualization and explanations.

Select: op
We found the op ROI (operators and progress indicators) of the select stimulus to have accumulated fixations four times faster (per expression; FR) in the experimental as compared to the control application, for long fixations only. We also found that an average subject spent about two thirds more time (GTP) looking at that ROI in the experimental application as compared to the control, for all and long fixations. Because the estimated subject's progress was shown in that ROI only in the experimental version of the application, it is perhaps not surprising that it drew subjects' attention. However, it seems that it was not just that. Subjective responses indicate that the subjects used the application estimation to select the next expression (Section 4.4, question S3; although, see also question F1).

Eval: op
For the op ROI of the eval stimulus, the rate of fixation accumulation (per minute; FR) was more than two times higher in the experimental application as compared to the control. We also found that an average subject spent about twice as much time (GTP) looking at that ROI in the experimental application than in the control. Both of the above hold only for all fixations. The interpretation of this differences is the same as with the previous ROI.

Eval: vis-s
For the vis-s ROI (static visualization), we found the difference in the number of fixations per minute of gaze (FR) between the applications to depend on the working memory index, for all and long fixations. An average low-span subject fixated on that ROI less than twice as often in

Table III. Results for fixation rate (FR; MSE), gaze time proportion (GTP; 95CL), average fixation duration (AFD; MSE), and gaze transitions (GT; MSE).

	ROI	y	MFD	x	p	EXPOS	app		wm		app×wm			
							ctrl (0)	exp (1)	low (0)	high (1)	ctrl–low	ctrl–high	exp–low	exp–high
select	op	FR	100	app	0.073	expr–cnt	0.18(0.08)	0.44(0.15)						
			300	app	0.012*	expr–cnt	0.02(0.01)	0.08(0.04)						
		GTP	100	app	0.041*	expr–cnt	OR: 0.38 (0.15, 0.95)							
			300	app	0.011*	expr–cnt	OR: 0.28 (0.11, 0.70)							
eval	op	FR	100	app	0.014*	sess–t	0.38(0.16)	0.85(0.33)						
			300	app	0.056	sess–t	0.06(0.04)	0.19(0.09)						
		GTP	100	app	0.040*	sess–t	OR: 0.49 (0.25, 0.96)							
			300	app	0.153	sess–t	OR: 0.39 (0.10, 1.54)							
	vis–s	FR	100	app×wm	0.011*	gt–tot					9.40(2.35)	7.18(1.66)	7.45(1.96)	14.17(2.78)
			300	app×wm	0.049*	gt–tot					2.55(0.85)	1.48(0.46)	2.01(0.71)	2.79(0.76)
		GTP	100	app×wm	0.035*	gt–tot	OR_l: 1.58, OR_h: 0.57		OR_c: 1.49, OR_e: 0.54					
			300	app×wm	0.047*	gt–tot	OR_l: 1.50, OR_h: 0.57		OR_c: 2.22, OR_e: 0.84					
	vis–d	FR	100	app	0.007**	anim–dur	2.70(0.43)	5.22(0.80)						
			300	app	0.005**	anim–dur	0.80(0.12)	1.63(0.12)						
		GTP	100	app	0.061	anim–dur	OR: 0.60 $(0.35, 1.03)_{95CL}$							
			300	app	0.020*	anim–dur	OR: 0.56 $(0.35, 0.90)_{95CL}$							
	txt	AFD	100	wm	0.023*	–			231.95(11.19)	197.36(9.71)				
			300	wm	0.420	–			432.47(8.47)	423.27(7.24)				
eval	vis-s - vis-d	GT	100	app×wm	0.021*	gt-tot			3.76(0.83)	3.02(0.62)	3.93(0.82)	6.61(1.17)		
	vis-d - op			app	0.046*	anim-dur	0.028(0.10)	0.064(0.22)						
	vis-d - vis-d			app	0.006**	anim-dur	1.76(0.27)	3.39(0.50)						
	vis-d - txt			app×wm	0.023*	anim-dur			0.79(0.13)	1.20(0.17)	1.76(0.29)	1.24(0.19)		

ROI – region of interest, MFD – minimum fixation duration, EXPOS – exposition variable (Table II)

the control than they did in the experimental application. Conversely, an average high-span subject fixated on that ROI about twice as often in the experimental application as they did in the control.

Moreover, we found the difference in GTP between the applications to depend on the working memory index, for all and long fixations. An average low-span subject spent almost twice as much time looking at that ROI in the experimental than in the control application. A reciprocal relationship held for an average high-span subject. That is, they spent twice as much time in the control application. An alternative interpretation that attributes the differences in gaze time proportion between the two working memory groups to the application version can be read off Table 4.3.1. The difference in GTP is bigger for long fixations although the evidence for the difference is weaker. Due to lack of space we do not provide odds ratio confidence limits for this interaction. The effects we observed for the vis-s ROI do not seem to have a clear interpretation other than pointing towards the mediating role of working memory.

Eval: vis-d

The vis-d ROI (dynamic visualization) drew more attention in the experimental application. It accumulated roughly twice as many fixations per minute of animation (FR) in the experimental then it did in the control application, for all and long fixations. An average subject spent almost twice as much time (GTP) looking at that ROI in the experimental than they did in the control application, but only for long fixations. If we were successful in assuring that long fixations correspond to instances of mental processing (as described in Appendix B and C) then we could say that an average subject exerted more mental effort when attending to dynamic visualization. That is, the experimental application presented more challenging visualization. One possibility is that the visualization was becoming increasingly uninteresting for subjects working with the control application due to presenting increasingly unnecessary (i.e., already understood) details.

The vis-d ROI was the region of the screen where all animations were displayed. Figure 10 shows a cumulative graph of the duration of those animations for both versions of the application. It is evident that adaptation was driving the animation duration down while that duration remained relatively stable in the control application. A higher number of long fixations on increasingly shorter animations could mean that the adaptation was able to assess the progress of subjects well and rendered the visualization more interesting to them. It could also be, however, that it made the visualization too confusing. To find out which of the two, interest or confusion, was more likely to explain the observed

subjects' behavior we looked at their responses to the exit questionnaire (see Section 4.4 for more details). Those responses indicate that subjects generally favored the experimental application over the control. More specifically, nine subjects (64%) said that the control application was presenting them with information they already knew (question V5) while only one subject (7%) said the same thing about the experimental application (question V6). When asked to evaluate the experimental application's ability to estimate the progress they were making, nine of the subjects (64%) said that the experimental application was assessing their progress accurately, none said it was overestimating it, two (14%) said it was underestimating it, and three (21%) could not tell. Given these results it seems likely that the adaptation induced interest instead of causing confusion.

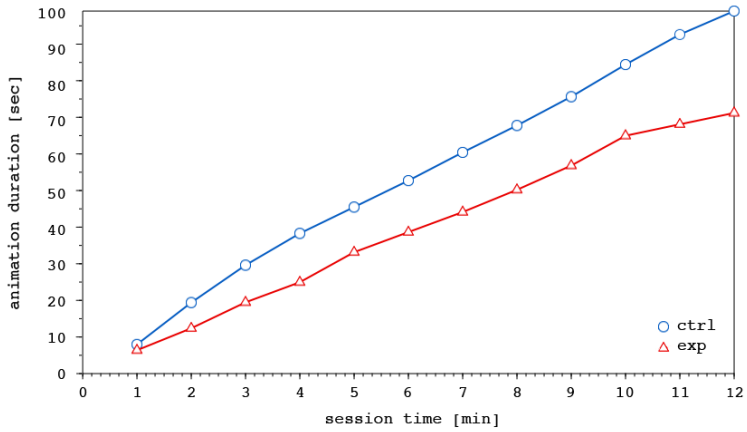

Figure 10. Duration of animation (cumulative) for the experimental and the control versions of the application. Sessions were as long as 15 minutes, but the first early-finisher subject exited the task after 12 minutes.

Overall, dynamic visualization drew a big portion of the subjects' attention (Figure 9).

Eval: txt

For the txt ROI (explanations), we found the AFD to be longer for low-span subjects, but only for all fixations. Reading-related cognitive processes (decoding, lexical access, parsing, and integration) were happening at a slower rate in low-span subjects. In general, fast readers make shorter fixations (as well as longer saccades and fewer regressions) than slow readers (Rayner, 1998). We did not perform or obtain any reading comprehension measurements, but simple working memory span (e.g., digit span) has been found to correlate with reading

comprehension (Daneman and Carpenter, 1980). Because the AFD for reading is about 250 ms, it is not surprising that we did not observe any effects for the long fixations data set. Overall, the explanations drew a big portion of the subjects' attention (Figure 9).

Summary
Here are the most important findings related to the aggregate eye movement measures that we discuss above. First, the list of operators and progress indicators (op) drew more attention in the adaptive version of the application. The subjects used the information presented in that part of the user interface when selecting the next expression. Second, the subjects spent more time looking at the dynamic visualization (vis-d) in the experimental application than they did in the control. It seems likely that adaptation rendered the visualization more interesting to them. Third, the low-span subjects read the explanations (txt) slower than the high-spans. Finally, dynamic visualization and explanations drew a major portion of subjects' attention.

4.3.2. *Gaze Transitions*
Aggregate measures tell us something about ROIs themselves, but they do not provide any information about the pattern of usage and the relationship between those ROIs. Therefore, additionally to aggregate measures, we investigated the number of GTs between all pairs of ROIs defined within each of the two stimuli.

We calculated GTs based on all-fixations data set, i.e. those having minimum fixation duration of 100 ms. To analyze the number of transitions we fitted a GLMM that was identical to the one for FR described in the previous subsection (Poisson regression), but we repeat it here for convenience

$$\log y_{ijk} = \mu + \alpha_i + \omega_k + \alpha\omega_{ik} + \log \tau_{jk} + b_{ik} + e_{ijk} \ ,$$

where μ is the overall mean, α_i is the effect of application i, ω_k is working memory of subject k, τ_{jk} is the exposition variable for subject k in trial j (Table II), b_{ik} is the random effect of subject k assigned to application i, and e_{ijk} is the random measurement error. We found significant effects only for the four pairs of ROIs from the eval stimulus reported in the bottom part of Table 4.3.1. All of them involved the ROI showing dynamic visualization (vis-d).

We found the number of GTs between the vis-s and the vis-d ROIs to depend both on the application version and working memory index at the same time. An average low-span subject would shift their visual attention more frequently in the control than in the experimental version of the application, but this difference was small (3.76 versus 3.93).

An average high-span subject, however, switched between the two ROIs more than twice less frequently in the control application than they did in the experimental one.

We found the application to be a significant predictor of the difference in the number of GTs between the vis-d and the op ROIs. There were more shifts in the experimental application than in the control. It is possible that because the experimental application was showing its estimation of the progress subjects were making, they used that estimation to focus on those operations that the application indicated as being the least understood.

A similar relationship held for the number of self-transitions for the vis-d ROI. There were more GTs in the experimental than in the control version. This is consistent with the difference in FR for the vis-d, almost twice higher in the experimental version (Table 4.3.1).

Finally, we found another interaction between application version and working memory index for the the vis-d–txt pair of ROIs. The amount of GTs per minute of animation that the high-span subjects made was almost identical in both versions of the application (1.20 versus 1.24). The low-span subjects, however, made more than twice as many transitions per minute of animation in the experimental application than they did in the control.

To summarize, all significant differences between the two versions of the application with respect to the number of GTs involved the dynamic visualization ROI. In two out of four pairs of ROIs, the working memory span mediated the effect of the application version. That provides further indication that working memory capacity may have an impact on how adaptation of explanatory software visualization is experienced by students.

4.4. Subjective Responses

Below, we present the results of a qualitative analysis of subjects' opinions about the two versions of the application. We hope that this analysis throws some light on what the subjects were thinking while performing the learning tasks. We turn to qualitative measures in part to uncover more differences between the applications since, as we argued earlier, the duration of the learning sessions might have rendered some of the qualitative differences undetectable.

Figure 11 shows the results of responses to the exit questionnaire that was administered at the end of the experiment (see Figure 3 for the experiment timeline). All responses were made on a five-point Likert-type scale. We divided the questions into the following four categories:

"general" (G), "version of the application" (V), "strategy for selecting next expression" (S), and "utility of features" (F).

Responses to the G-questions showed a positive reception of the application. Responses to the V-questions revealed that subjects favored the experimental application over the control. Please note, that positive attitude towards the experimental application required a negative response to questions V1 and V6.

Responses to the S-questions indicate that when selecting the next expression to explore, about 75% of subjects used the experimental application's estimation of the progress they made so far (question S3). While this last observation may seem like a good news for the adaptive version of the application, we prefer to view it as a caveat. Because the students tended to trust the application's estimation, one should strive to make it robust and perhaps even refrain from providing it altogether if an application does not have a sufficient certainty in the validity of its own assessment.

Responses to the F-questions show a general positive opinion the subjects had about the features of cWADEIn. However, two features stand out as less useful than others. First, the subjects had mixed feelings about the progress indicators in the adaptive application (question F1). Roughly half thought they were not useful. Interestingly, about three quarters admitted they actually used that feature (question S3). One of the reasons that those indicators received an equivocal utility rating could be that the progress bars were displayed in light gray. Using a more conspicuous color could help. The reason for light gray is that cWADEIn treats the progress made in its Exploration mode as "unconfirmed." To confirm it, a student needs to solve expression evaluation problems by themselves in the Evaluation mode (as described in Section 2). Once that happens, the progress bars become greener and therefore more visible. Again, the Evaluation mode was not used in this experiment.

Second, the beginning-end navigation (question F6) received a lower utility rating than the other two navigation options (i.e., F4: step-by-step and F5: operator-by-operator). This is not surprising given that this navigation type is used to restart the evaluation of an expression or to end that evaluation. Therefore, it must be used much less often than the other two.

5. Discussion

As we have shown above, the rate of expression evaluations explored has a potential of telling the difference between an adaptive and a non-

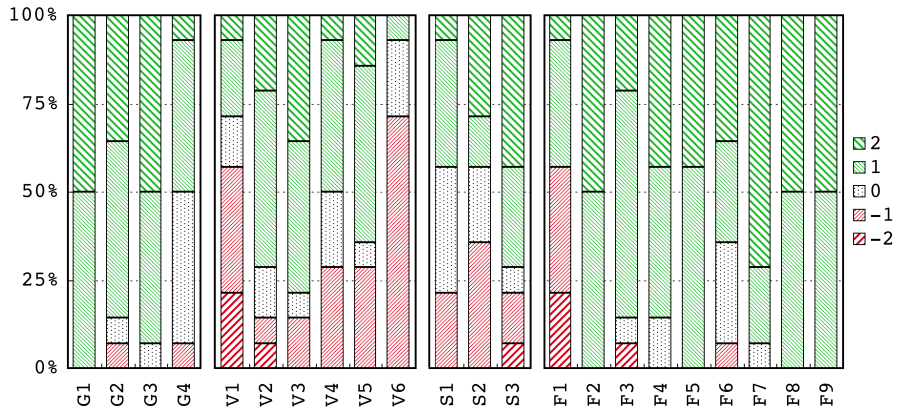

Figure 11. Responses to the exit questionnaire. -2 corresponds to the highest level of disagreement, 2 to the highest level of agreement, and 0 to "no opinion." The questions were:

G1. The applications interface is easy to use.
G2. Animations helped me to understand the material.
G3. Explanations helped me to understand the material.
G4. Using certain colors in certain contexts helped me to understand the material.

V1. I liked the *basic application* more.
V2. I liked the *progress-tracking* application more.
V3. I liked speeding up animations in the *progress-tracking* application.
V4. I liked shortening of explanations in the *progress-tracking* application.
V5. I felt the *basic* application was presenting the material I already knew.
V6. I felt the *progress-tracking* version was presenting the material I already knew.

S1. I was going through expr. as they were ordered in the drop-down list.
S2. I was using the list of current goals.
S3. In the *progress-tracking* version I was using progress indicators.

F1. Progress indicators in the *progress-tracking* version [are useful].
F2. Ordering expressions in the dropdown list according to difficulty [is useful].
F3. Ability to change values of variables [is useful].
F4. Step-by-step navigation [is useful].
F5. Operator-by-operator navigation [is useful].
F6. Beginning-end navigation [is useful].
F7. Textual explanations [are useful].
F8. Ability to see original expression at all times [is useful].
F9. Ability to see the order of execution at all times [is useful].

NOTE: The actual questions were a little bit more verbose in order to make sure that the subjects understood what each question was asking about. We removed those clarifications for succinctness.

adaptive version of an educational application featuring explanatory visualization. When working with the adaptive version of cWADEIn, subjects were able to explore more expressions. Weber and Brusilovsky (2001) also found that adaptive educational application can help students in exploring a significantly larger volume of learning content than its non-adaptive counterpart. Looked at in another way, the adaptive version of cWADEIn has a potential of allowing for a material exposition comparable with a non-adaptive version, but in a shorter amount of time.

We did not find the two versions of the application different with respect to the gain score. If we treat a performance on a task to be positively correlated with the amount of work done towards that task then those two results provide evidence for the existence of the ceiling effect in our study. The fact that the pretest scores of subjects that left the study early were significantly higher than the other subjects further emphasizes this possibility. If the learning sessions were shorter, the adaptive version of the application could have allowed subjects to explore material beyond what could be explored in the non-adaptive version.

We found that the learning task order was responsible for a significant amount of variation in the gain score and the number of expressions explored by the subjects. That this is unlikely due to anything else than practice because we counterbalanced the order of the application version and made the operator sets exchangeble in terms of difficulty and knowledge transfer. This shows that after training the subjects were not yet sufficiently accustomed to the application's user interface and features. Despite the short duration of the training session (five minutes) this is somewhat surprising given that the interface was quite simple. It might be that explanatory visualization made the interface less accessible to the subjects than what we thought it would. In experiments involving explanatory visualization, it may be worthwhile to put more emphasis on assessing the time it takes for users to get familiar with an interface.

Several interesting artifacts can be found in the descriptive statistics for gaze data (Figure 9). First, fixation count and gaze time show that the vis-d and the txt ROIs were the ones that received the biggest chunk of subjects' attention. This provides evidence for the utility of natural language explanations accompanying visualization. Second, the fact that the txt ROI accumulated more fixations than did the vis-d is a result of the fixations being shorter on the former. Because of that, fixation count, while sometimes employed as an isolated measure, may be less suitable a measure of the importance of a ROI. Gaze time, which combines both the count and duration of fixations, might be

a better choice. Third, the op ROI of the select stimulus accumulated more fixations and more gaze time in the experimental version of the application as compared to the control. Experimental version presented subjects with the application's estimation of their progress. That probably made them appreciate that ROI more in that version of the application. Fourth, some of the longest fixations happened on the expr ROI. That ROI also accumulated a relatively large number of fixations. Those two facts translate into a relatively large amount of gaze time. This suggests that the selection of the next expression was a task involving large amounts of mental processing. Subjects were most likely attempting to self-assess their understanding. The involvement in self-assessment has been shown to be beneficial for learning (Boud, 1995). It is possible that adaptive problem selection, if implemented in cWADEIn, could have had a detrimental effect on students progress by decreasing the role of self-assessment. All of the relationships mentioned above can be observed for both all and long fixations data sets.

The eye movement analysis provided interesting insights that might guide further investigation. In the case of several ROIs we observed a higher amount of perceptual activity in the adaptive version of the application. This activity was measured with the fixations rate, the gaze time proportion, and the number of gaze transitions between two ROIs. One potential explanation of this finding is that the content presented in the adaptive version of the application was more engaging or interesting and required more activity on the part of subjects. Alternatively, it is possible that the material displayed in the adaptive version was too confusing and subjects were lost when the adaptation was present. Given that subjects were exposed to the evaluation of significantly higher number of expressions, which must have increased the contribution of the novelty factor, the first explanations seems more plausible. Furthermore, the students preferred working with the adaptive version of the application, which probably would not be the case if it notoriously presented them with confusing material.

We found that the subjects made more gaze transitions between ROIs in the adaptive version of the application. In some cases this effect was mediated by the working memory index. Evidence provided by Conati and Merten (2007) indicates that a higher incidence of gaze transitions could be associated with self-explanation. It could be that our subjects were more likely to engage in self-explanation when adaptation was present.

In the case of several other ROIs, we found the working memory span to mediate the effect of the application version with respect to eye movement measures. If the differences in mental processing capabilities are reflected by the results of the modified digit span task that we

employed, it is possible that adaptation in the context of explanatory software visualization provides a different service to people with different mental processing capabilities. This finding may be pointing in the direction of the Cognitive Load Theory (Pass et al., 2003; Sweller, 1998) which postulates that working memory limitations plays an important role in learning.

There are several considerations that should be taken into account with regards to eye movement data we collected and analyzed. More accurate eye trackers than the one we used exist (e.g., EyeLink 1000). However, the fact of it being a remote eye tracker decreased intrusiveness thereby increasing ecological validity of the experiment. Precision is the more important the smaller and denser ROIs are. In our case, ROIs were fairly large and were reasonably far apart. Keeping ROIs apart, however, contributed to the loss of fixations made in-between two ROIs. It is also possible that fixations registered as being close to the border of a particular ROIs might have been misclassified. And even fixations correctly associated with a given ROI could bear a false information; subjects might had been encoding information from the parafovea. That is, however, likely to have introduced noise of a unnoticeable magnitude.

Sample size might rise some concerns. We could surely rest more confidence in the results had more than 14 subjects been recruited. However, all response variables were within-subject. Repeated measures designs provide more power than their between-subject counterparts by accounting for the correlation between observation made on the same unit and therefore require a smaller n.

In this experiment, we employed a simple working memory span task. That task includes only the memory component and emphasizes information storage and rehearsal. It would be beneficial to investigate the effect of employing a "complex" working memory span task, i.e., one that requires processing of additional information by introducing a secondary task. The most popular candidates are counting span (Case et al., 1982), operation span (Turner et al., 1989), and reading span (Daneman and Carpenter, 1980) tasks.

Finally, the current work involved the evaluation of adaptation in the context of program visualization. It is not clear whether and, if yes, to what extent do the results we obtained generalize to visualizations of different kinds.

6. Conclusions

We have presented an evaluation of user-adaptation features of the cWADEIn application which uses explanatory visualization to supports students in learning about expression evaluation in the C programming language. We compared the user-adaptive version of the application to another version, deprived of adaptive capabilities. Both versions allowed students to achieve a marked increase in their understanding of the material. However, we found the versions indistinguishable with respect to the pretest-posttest difference. We have argued that this may be due to ceiling.

We also found that students preferred the adaptive version and used its estimation of the progress they were making to pace their work. It is therefore important that user-adaptive educational applications strive for a progress estimation which is as accurate as possible.

The adaptive version allowed students to explore expressions at a significantly higher rate than did the non-adaptive version. This shows that adaptation has the potential of improving material exposition. It can also provide time savings for students by allowing them to explore the domain of interest in a smaller amount of time.

The results of an exploratory eye movement protocol analysis show that the adaptive version of the application engaged students more than the non-adaptive version. More specifically, adaptive visualization attracted more attention than its non-personalized counterpart. Together with students' questionnaire responses, this finding seems to indicate that adaptive visualization was more interesting. Furthermore, natural language explanations received a big portion of students' attention which speaks of their utility.

The results further suggest that differences in working memory capacity can influence students' experience with the feature of adaptation. At this point, it is unclear what this influence might signify and what might its implications be. Further investigation aimed at confirming the existence, direction, and quantifying the magnitude of that relationship is necessary.

Acknowledgments.

This material is based upon work supported by the National Science Foundation under Grant No. 0426021. We would like to thank Roman Bednarik, Rosta Farzan, Alfred Kobsa, Christian Shunn, Sergey Sosnovsky, and anonymous reviewers for their valuable comments on earlier versions of this manuscript.

References

Bednarik, R., Myller, N., Sutinen, E., and Tukianinen, M.T.: 2005, "Effects of experience on gaze behavior during program animation." *Proceedings of the 17th Annual Psychology of Programming Interest Group Workshop (PPIG)*, 49–61.

Blumenktants, M., Starovisky, H., and Shamir, A.: 2006, "Narrative algorithm animation." *Proceedings of ACM Symposium on Software Visualization*, 17–26.

Boud, D.: 1995, "Enhancing learning through self assessment." New York, NY: Routledge.

Boyle, C. and Encarnacion, A. O.: 1994, "Metadoc: An adaptive hypertext reading system." *User Modeling and User-Adapted Interaction*, 4(1), 1–19.

Brusilovsky, P.: 1993, "Program visualization as a debugging tool for novices." *Proceedings of the 5th International Conference on Human-Computer Interaction (INTERCHI; Adjunct Proceedings)*, 29–30.

Brusilovsky, P.: 1994, "Explanatory visualization in an educational programming environment: Connecting examples with general knowledge." *Proceedings of the 4th International Conference on Human-Computer Interaction*, 202–212.

Brusilovsky P. and Loboda, T. D.: 2006, "WADEIn II: A case for adaptive explanatory visualization." *Proceedings of the 10th Conference on Innovation Technology in Computer Science Education (ITiCSE)*, 48–52.

Brusilovsky, P. and Spring, M.: 2004, "Adaptive, engaging, and explanatory visualization in a C programming course." *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 1264–1271.

Brusilovsky, P. and Su, H.-D.: 2002, "Adaptive visualization component of a distributed Web-based adaptive educational system." *Proceedings of the 6th International Conference on Intelligent Tutoring Systems (ITS)*, 229–238.

Byrne, M. D., Catarambone, R., and Stasko, J. T.: 1999, "Evaluating animations as student aids in learning computer algorithms." *Computers and Education*, 33(5), 253–278.

Case, R., Kurland, M. D., and Goldberg, J.: 1982, "Operational efficiency and the growth of short-term memory span." *Journal of Experimental Child Psychology*, 33, 386–404.

Conati, C. and Merten, C.: 2007, "Eye-tracking for user modeling in exploratory learning environments: An empirical evaluation." *Knowledge-Based Systems*, 20(6), 557–574.

Daily, L. Z., Lovett, M. C., and Reder, L. M.: 2001, "Modeling individual differences in working memory performance: A source activation account." *Cognitive Science: A Multidisciplinary Journal*. 25(3), 315–353.

Dancik, G. and Kumar, A. N.: 2003, "A tutor for counter-controlled loop concepts and its evaluation." *Proceedings of 2003 Frontiers in Education Conference*, Session T3C.

Daneman, M. and Carpenter, P. A.: 1980, "Individual differences in working memory and reading." *Journal of Verbal Learning and Verbal Behavior*, 19(4), 450–466.

Dmitrienko, A., Molenbergs, G., Chuang-Stein, C., and Offen, W.: 2005, "Analysis of clinical trials using SAS: A practical guide." SAS Publishing.

Graf, W. and Krueger, H.: 1989, "Ergonomic evaluation of user interfaces by means of eye movement data." *Proceedings of the 3rd World Conference on Educational Multimedia, Hypermedia and Telecommunications Conference on Human-Computer Interaction (HCI)*, 659–665.

Henderson, J. M. and Pierce, G. L.: 2008, "Eye movements during scene viewing: Evidence for mixed control of fixation durations." *Psychonomic Bulletin and Review*, 15(3), 566–573.

Hundhausen, C. D., Douglas, S. A., and Stasko, J. T.: 2002, "A meta-study of algorithm visualization effectiveness." *Journal of Visual Languages and Computing*, 13(3), 259–290.

Hooge, I. Th. C., Vlaskamp, B. N. S., and Over, E. A. B.: 2003, "Saccadic search: On duration of a fixation." In van Gompel, R. P. G., Fischer, M. H., Murray, W. S., and Hill, R. L. (Eds.), *Eye Movements: A Window on Mind and Brain* (pp. 581–596). Amsterdam, The Netherlands: Elsevier.

Jarc, D. J., Feldman, M. B., and Heller, R. S.: 2000, "Assessing the benefits of interactive prediction using Web-based algorithm animation courseware." *Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education*, 377–381.

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

Kenward, M. G. and Roger, J. H.: 1997, "Small sample inference for fixed effects from restricted maximum likelihood." *Biometrics*, 53, 983–997.

Kerren, A., and Stasko, J.: 2002, "Algorithm animation - Introduction." In: S. Diehl (ed.): *Software Visualization State of the Art Survey* (pp. 1–15). Springer.

Kerren, A., Mueldner, T., and Shakshuki, E.: 2006, "Novel algorithm explanation techniques for improving algorithm teaching." *Proceedings of the ACM symposium on Software visualization (SoftVis)*, 175–176.

Krebs, M., Lauer, T., Ottmann, T., and Trahasch, S.: 2005, "Student-built algorithm visualizations for assessment: flexible generation, feedback and grading." *Proceedings of the 9th Conference on Innovation Technology in Computer Science Education (ITiCSE)*, 281–285.

Kobsa, A., Koenemann, J., and Pohl, W.: 2001, "Personalised hypermedia presentation techniques for improving online customer relationships." *The Knowledge Engineering Review*, 16(2), 111–155.

Kumar, A. N.: 2003, "Model-based generation of demand feedback in a programming tutor." *Proceedings of the 11th International Conference on Artificial Intelligence in Education (AIED)*, 425–432.

Kumar, A. N.: 2005, "Results from the evaluation of the effectiveness of an online tutor on expression evaluation." *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, 216–220.

Lahtinen, E. and Ahoniemi, T.: 2007, "Annotations for defining interactive instructions to interpreter based program visualization tools." In G. Rössling (Ed.), *Electronic Notes in Theoretical Computer Science, Vol. 178* (pp. 121–128).

Littell R. C., Milliken G. A., Stroup W. W., Wolfinger R. D., and Schabenberger O.: 2006, "SAS for mixed models, Second Edition." SAS Publishing.

Loboda, T. D. and Brusilovsky, P.: 2008, "Adaptation in the context of explanatory visualization." *Proceedings of the 3rd European Conference on Technology Enhanced Education (ECTEL)*, 250–261.

Loftus, G. R.: 1983, "Eye fixations on text and scenes." In K. Rayner (Ed.), *Eye Movements in Reading: Perceptual and Language Processes* (pp.359–376). New York, NY: Academic Press.

Manor, B. R. and Gordon, E.: 2003, "Defining the temporal threshold for ocular fixation in free-viewing visuocognitive tasks." *Journal of Neuroscience Methods*, 128(1-2), 85–93.

McCullagh, P. and Nelder, J. A.: 1989, "Generalized Linear Models." Boca Raton, FL: CRC Press.

Moreno, A., Myller, N., Sutinen, E., and Ari, M. B.: 2004, "Visualizing programs with Jeliot 3." *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, 373–376.

Naps, T. L., Eagan, J. R., and Norton, L. L.: 2000, "JHAVE – an environment to actively engage students in Web-based algorithm visualizations." *Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education*, 109–113.

Naps, T. L., Rössling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M., Rodger, S., and Velázquez-Iturbide, J. Á.: 2002, "Exploring the role of visualization and engagement in computer science education." *ACM SIGCSE Bulletin*, 35, 131–152.

Naps, T. L., Rössling, G., Anderson, J., Cooper, S., Dann, W., Fleischer, R., Koldehofe, B., Korhonen, A., Kuittinen, M., Leska, C., McNally, M., Malmi, L., Rantakokko, J., and Ross, R. J.: 2003, "Evaluating the educational impact of visualization." *ACM SIGCSE Bulletin*, 35(4), 124–136.

Nevalainen, S. and Sajaniemi, J.: 2006, "An experiment on short-term effects of animated versus static visualization of operations on program perception." *Proceedings of the 2nd International Workshop on Computing Education Research*, 7–16.

O'Regan, J. K.: 1992, "Optimal viewing position in words and the strategy-tactics theory of eye movements in reading." In K. Rayner (Ed.), *Eye Movements and Visual Cognition: Scene Perception and Reading* (pp. 333–354). New York, NY: Springer-Verlag.

Paas, F., Renkl, A., and Sweller, J.: 2003, "Cognitive load theory and instructional design: Recent developments," *Educational Psychologist*, 38, 1–4.

Price, B.: 1993, "A principled taxonomy of software visualization." *Journal of Visual Languages & Computing*, 4(3), 211–266.

R Development Core Team: 2009, "R: A Language and Environment for Statistical."

Rayner, K.: 1998, "Eye movement in reading and information processing: 20 years of research." *Psychological Bulletin*, 124(3), 372–422.

SAS Institute Inc.: 2008, "SAS 9.2 help and documentation."

Salthouse, T. A. and Ellis, C. L.: 1980, "Determinants of eye-fixation duration." *The American Journal of Psychology*, 93(2), 207–234.

Stasko, J., Badre, A., and Lewis, C.: 1993, "Do algorithm animations assist learning? An empirical study and analysis." *Proceedings of the 5th International Conference on Human-Computer Interaction (INTERCHI)*, pp. 61–66.

Sweller, J.: 1988, "Cognitive load during problem-solving: Effects on learning," *Cognitive Science*, 12, 257–285.

Turner, M. L., Engle, R. W.: 1989, "Is working memory capacity task dependent?" *Journal of Memory & Language*, 28, 127–154.

Velichkovsky, B. M., Rothert, A., Kopf, M., Dornhoefer, S. M., and Joos, M.: 2002, "Towards an express diagnostics for level of processing and hazard perception." *Transportation Research, Part F*, 5(2), 145–156.

Velichkovsky, B. M., Joos, M., Helmert, J. R., and Pannasch, S.: 2005, "Two visual systems and their eye movements: Evidence from static and dynamic scene perception." *Proceedings of the 27th Conference of the Cognitive Science Society*, 2283–2288.

Vaida, F. and S. Blanchard: 2005, "Conditional Akaike information for mixed-effects models." *Biometrika*, 92(2), 351–370.

Weber, G. and Brusilovsky, P.: 2001, "ELM-ART: An adaptive versatile system for Web-based instruction." *International Journal of Artificial Intelligence in Education*, 12(4), 351–384.

Yamamoto, Y. and Hirose, H.: 2005, "Result of applying study support system that flow chart diagram displays by synchronizing with source program code to education." *Proceedings of the 10th World Conference on E-Learning (E-LEARN)*, 1186–1192.

Appendices

A. Working Memory Index

The task of understanding expression evaluation is symbolic in nature. Since working memory span could have an impact on performance in the task we measured subjects' sensitivity to the increase in the memory load with the modified digit span task (MODS; Daily et al., 2001). This task emphasizes individual differences in working memory and reduces impact of other individual differences, e.g. prior knowledge and usage of compensatory strategies. It was administered right after the entry questionnaire (demographics) and before study introduction and the first pretest (see Figure 3 for the experiment timeline).

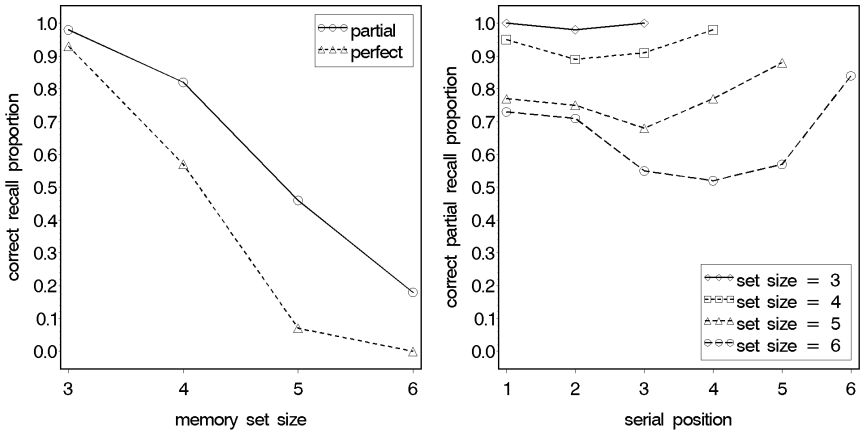

Figure 12. The proportion of correct (perfect and partial) recall as a function of memory set size (left) and the proportion of correct (perfect) recall as a function of serial position for the four different set sizes (right).

In each trial of the MODS task a subject was presented with a string of letters and digits. Their task was to remember the digits for later recall. To suppress subvocal rehearsal subjects were asked to read all characters aloud as they appeared on the screen. A metronome tick sound was used to help time the articulation. Each letter was presented for 0.5 sec. The presentation time for digits was lengthened to 0.91 sec to help to encode them in the memory. The number of letters preceding each digit was three or four (selected randomly). This variation was introduced to mask the position of the next digit.

Each of the subjects started with three digits and went through stimuli with four, five, and six of them, which yields a total of four conditions. All subjects went through four trials per condition, which

yields a total of 16 trials (not counting the three training ones). The total length of the stimulus was between 12 and 30 characters. Figure 12 depicts plots of the proportion of correct answers as a function of memory set size and serial position. The primacy and recency effects are clearly visible on the second plot.

Each subject started each trial with a screen showing 30 empty boxes (Figure 13). The stimuli presentation begun after the subject acknowledged their readiness by clicking the "Ready" button. After the entire stimulus was presented all boxes were cleared and a recall prompt was presented. This prompt highlighted boxes previously occupied by digits. The subjects had to provide the digits in the order they were originally presented. Backtracking was not possible. They could skip a digit they did not remember by using the letter "X".

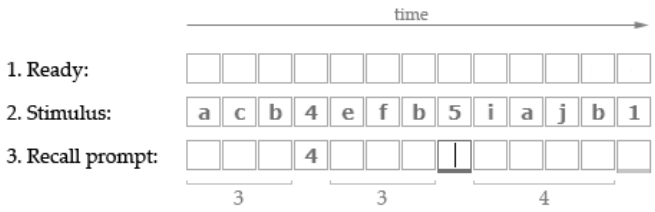

Figure 13. A sample trial in the MODS task (set size 3). The actual stimulus was composed of 30 empty boxes.

To differentiate subjects with respect to their working memory capacity we calculated an index ω for each of them. We did that by averaging the partial recall proportions from the MODS task for set sizes four, five, and six. We excluded set size three due to its small influence (almost no variability in the partial recall proportions; Table IV).

Table IV. Partial recall proportions in the MODS task.

Set size	M (SD)
3	0.99 (0.02)
4	0.93 (0.08)
5	0.77 (0.16)
6	0.65 (0.15)
ω	0.79 (0.08)

Figure 14 depicts the values of w for all subjects. To get an idea of how it relates to more "tangible" determinants of performance, we also

plotted self-reported SAT (Scholastic Aptitude Test; twelve subjects) and ACT (American College Testing; 2 subjects) math test scores, after normalizing them to the $[0, 1]$ range ($\rho_{math,\omega} = 0.54$). As shown on the figure, if we performed a median split on those test scores in order to divide our sample into two groups, we would have obtained the same result as with median split on w for all but subject 7.

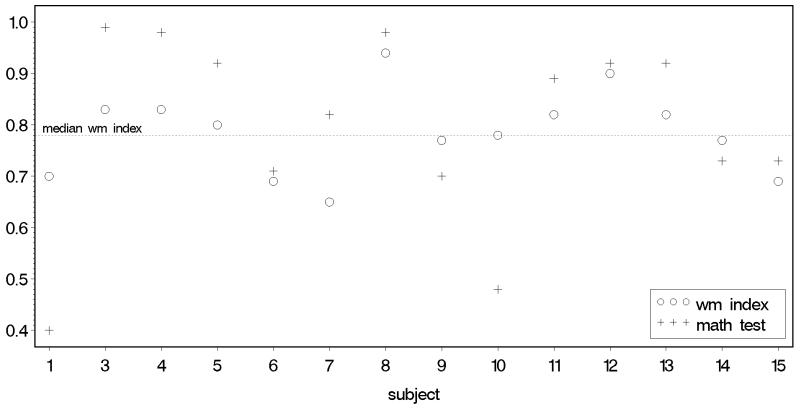

Figure 14. Working memory index w and self-reported math aptitude test scores ($\rho_{math,w} = 0.54$).

B. Fixation Radius

Typically, fixation radius α is considered to be approx. 1° (Loftus, 1983). Since ClearView 2.6.3 only accepts values in pixels (px) we used the following formula[7] to convert α into pixels

$$P = D \tan \alpha \left(\frac{d \sin \arctan(\frac{s_y}{s_x}) \cdot 25.4}{s_y} \right)^{-1} ,$$

where D is the distance of the subject's eyes from the screen in mm, d is the diagonal of the screen in inches, s_y is the vertical resolution of the screen in pixels, and s_x is the horizontal resolution of the screen in pixels. Please note, that the above formula takes into account the fact that using different screen resolutions result in different actual pixel sizes (in mm). Below is the calculation performed for the case of our experiment

$$P = 600 \text{ mm} \cdot \tan 1^\circ \left(\frac{17'' \cdot \sin \arctan(\frac{600 \text{ px}}{800 \text{ px}}) \cdot 25.4}{600 \text{ px}} \right)^{-1} = 24.25 \text{ px} .$$

[7] http://www.sis.pitt.edu/~tloboda/res/research/deg-to-fix

We rounded that value up and thus used 25 pixels as the fixation radius.

C. Fixation Duration

From a temporal point of view, saccades and fixations mix if the temporal fixation threshold is smaller than approx. 60 ms. Therefore fixations should be defined as not shorter than that. Indeed, 100 ms seems to be a reasonable minimum (Manor and Gordon, 2003), and that was the value we used as the minimum fixation duration (MFD) in our analysis.

Additionally to focusing on all fixations, we wanted to look at fixations for which the association with mental processing should be more pronounced. As suggested by Just and Carpenter (1976), the duration of a fixation is indicative of the amount of mental processing the object of that fixation receives. The results of a recent study conducted by Henderson and Pierce (2008) provide strong evidence that during scene viewing individual fixations can be directly and immediately controlled by the current visual stimulus. Using larger values for the *minimum fixation duration* would allow us to focus on fixations more likely representing the occurrences of mental processing. As Hooge et al. (2003) report, longer average fixations are usually found in more difficult tasks.

Just and Carpenter (1976) investigated eye movement behavior associated with mental processing. They looked at mental rotation, sentence verification, and qualitative comparison tasks. Even though their unit of analysis was gaze time (i.e. the sum of durations of fixations), they report the average duration of a fixation in the mental rotation task increase from 200 ms at 0° to 320 ms at 180°.

In their tachistoscopic experiments, Salthouse and Ellis (1980) established two facts relevant to this discussion. First, they found the minimum fixation duration while processing information to be about 250 ms. Second, they found the time required for actual stimulus processing to be less than half this duration, approximately 80 to 100 ms. In their case that processing pertained to a relatively easy task of deciding whether a letter was a vowel or a consonant.

Graf and Krueger (1989) introduced a distinction between voluntary (>320 ms) and involuntary (<240 ms) fixations. O'Regan (1992) considered fixations lasting 300 ms or more as long fixation. Rayner (1998) reports approximate mean fixation durations for silent reading (225 ms), oral reading (275 ms), visual search (275 ms), scene perception (330 ms), music reading (375 ms), and typing (400 ms).

Velichkovsky et al. (2002) report results of a simulated driving task. Their experimental data suggests fixations up to 300 ms long decrease in numbers around the appearance of an immediate hazard (car in front

braking, traffic light turning red, a pedestrian walking onto the street) while those longer than 600 ms become more frequent. The number of fixations being 300 to 600 ms long seemed not to be affected. They provide a distinction between the preattentive fixations (shorter than 250 ms) and attentive fixations (longer than 280-300 ms). Velichkovsky et al. (2005) propose those shorter fixations to appear in the ambient mode of processing while longer ones to be indicative of focal processing. They also note that fixations are shorter in static than in dynamic environments.

D. Data Analysis Considerations

Multiplicity Adjustment

We collected four protocols: short-term knowledge gain, eye movements, interaction logs, and subjective responses. Multiplicity adjustment usually pertains to multiple tests run on a single data set. Because the knowledge gain and the interaction logs data sets were separate and only one comparison involved each of them we did not perform any adjustment.

The results related to the analysis of eye movement data are exploratory in nature. One important implication of this is that we did not feel obliged to perform any type of multiplicity adjustment here either. A statistically inclined reader will be interested to know that some of the measures we employed were highly correlated (Table V). This is important because if multiple tests procedures that make full use of the joint distribution of test statistics are used, the higher the correlation between end-points the smaller the resulting p-values adjustment (Dmitrienko et al., 2005). The correlations were higher for long fixations (i.e. MFD=300 ms) than for all fixations (i.e., MFD=100 ms).

Table V. Correlations between the eye movement dependant variables (averaged over all ROIs).

y1	y2	MFD	ρ
FR	GTP	100	0.55
FR	GTP	300	0.79
FR	AFD	100	-0.04
FR	AFD	300	0.39
GTP	AFD	100	0.23
GTP	AFD	300	0.52

Subjective responses were used to perform qualitative analysis only and gain insight into what the subjects thought about the application and its features.

Model Selection and Fit

In a few of the comparisons we dealt with the response variable representing counts or rates. A rate is derived from counts by dividing the count by the value of the exposition variable (e.g., number of fixations per minute). Count and rate data is usually treated with Poisson regression models. The Poisson distribution arises from counts of independent events. Because of that, whenever we used Poisson regression, for the purpose of model fitting the occurrences of events being counted were implicitly assumed to be independent.

Overdispersion is a common problem of Poisson models. It stems from the equidispersion property of the Poisson distribution, i.e. the mean of the distribution being equal to its variance. That property makes the model poorly fit data for which the variance is higher or lower than the mean.

To check for overdispersion we looked at the variance of the Pearson residuals. Values close to 1 indicate that the variability in the data has been properly modeled. The mean of values we obtained was .64 ($SD =$ .13). The ratio of the generalized chi-square statistic and its degrees of freedom is sometimes used to detect problems with dispersion. As Littell et al. (2006) point out, values of that ratio that are much higher (or lower) than 1 are not necessarily indicative of overdispersion (or underdispersion).

To account for overdispersion we included a multiplicative scale parameter on the variance function in all models showing evidence of Pearson residual variablity much higher than 1. Fitting a negative binomial model is an alternative and often employed way of dealing with overdispersion. However, the counting process matches the nature of our data better.

Overdispersion is also a problem in logistic models, because just like in Poisson models there is a relationship between the mean and the variance. More specifically, if π is the probability of an event of interest, the variance is given as $\pi(1 - \pi)$. To account for that we allowed the estimation of the scale parameter.

All models we fitted were mixed models, i.e. they contained both fixed and random effects (specifically, subject random effect). We assumed random effects to be normally distributed and independent of each other. We used the variance component structure for the covariance matrix of random effects. In all models we fitted we used the Kenward-Roger correction for degrees of freedom and standard errors,

a method suggested for repeated measures data with small sample sizes (Kenward and Roger, 1997). To assess the model fit we looked at the studentized conditional residual plots.

Vita

Tomasz D. Loboda is a Ph.D. candidate in Information Science at the University of Pittsburgh. He received his M.S. in Computer Science from the Wroclaw University of Technology in 2003. His primary interests lie in the areas of user modeling and Bayesian networks, although previous research has included work in information retrieval.

Peter Brusilovsky is Associate Professor of Information Science and Intelligent Systems at the University of Pittsburgh, where he also directs Personalized Adaptive Web Systems (PAWS) lab. He has been working in the field of adaptive educational systems, user modeling, and intelligent user interfaces for more than 20 years. He published numerous papers and edited several books on adaptive hypermedia and the adaptive Web. Peter received his Ph.D. degree in Computer Science from the Moscow State University in 1987. He also holds Doctor honoris causa degree from by the Slovak University of Technology in Bratislava.