An Experiment on the Effects of Engagement and Representation in Program Animation Perception

Seppo Nevalainen and Jorma Sajaniemi

University of Joensuu, Department of Computer Science,
P.O.Box 111, 80101 Joensuu, Finland,
seppo.nevalainen@cs.joensuu.fi

Abstract. When visualization tools utilized in computer science education have been evaluated empirically, the results have been controversial. The extent to which the tools have benefited learning has remained unclear, as well as the ways through which the benefit has been achieved. In our research, we have chosen to vary type of students' engagement and representation of the visualization tool in a series of experiments, in order to investigate the phenomena taking place during individual viewing of visualizations.

In the current experiment, we varied student engagement using two different tasks to perform during viewing; data flow task (D) and control flow task (C). Representation of visualization was varied by using two versions of the program animator; one with special images and animation, and one without. The results show that while the distribution of visual attention of the participants performing task D was steady throughout the time, the participants performing task C focused their visual attention at the beginning almost solely on the code, and increased their visual attention to the other parts of the visualization on the second half of the viewing. The participants performing task D also benefited most from the tool, at least regarding programming knowledge.

1 Introduction

Algorithm and program visualization tools are utilized for example in computer science education (CSE), where their use has been rationalized by their proposed ability to make program related abstract entities more concrete, and this way more accessible to students. However, when visualization tools have been evaluated empirically, the results have been controversial [8]. It has remained unclear to what extent tools actually benefit learning, in what way, and in what kind of situations.

There are several possible factors that can affect the outcome of the utilization of visualization tools in CSE, e.g., intelligibility of the visualization's symbols to the user and appropriateness of the speed of animation. In our research, we have chosen to vary the type of students' engagement and the representation of visualization in a series of experiments, where we move the focus of investigation from measuring and analyzing effects of visualization tools after the viewing to the investigation of the phenomena taking place during viewing. We will use a program visualization tool, whose long-term beneficial effects have been verified earlier, and investigate visual attention during viewing, participants' mental models of the viewed programs, and their general programming knowledge immediately after viewing. This way, we collect information that can be used to explain in what situations, and in what ways the tool's immediate and short-term effects are accumulated into the beneficial long-term effects.

In the current experiment, student engagement was varied by using two different tasks to perform during viewing. Representation was varied by using two versions of program animator. Visual attention was measured using eye-tracking equipment, participants' mental models of the studied programs were investigated by analyzing program summaries, and their programming knowledge development was measured with a pretest and a post-test.

Section 2 provides some background and describes the research carried out so far. Section 3 describes the experiment and its results. In section 4 the results of the experiment are discussed. Finally, Section 5 contains the conclusions.

2 Background

The role of engagement in utilizing visualization techniques in CSE has been discussed in literature for example by Hundhausen and Douglas [7], and later by Hundhausen et al. [8]. Hundhausen and Douglas divided evaluations to two groups according to whether they varied level of learner involvement or representational characteristics of the visualizations. They found out that evaluations varying level of learner involvement produced always significant results, whereas evaluations varying representational characteristics of the visualizations produced significant results only rarely. Naps et al. [10] have reviewed literature on visualization technology and suggested a taxonomy of learner engagement with visualization technology, along with a framework for experimental studies of visualization effectiveness.

The role of representational characteristics of visualizations in CSE has been studied for example by Lawrence [9], and by Stasko et al. [18]. The results of these studies, however, are mixed. Even though the notion that pictures are worth thousand words is intuitively appealing, the studies do not build up into a coherent evidence for the beneficiality of the use of visualizations in teaching.

2.1 Immediate and Short-term Effects of Visualizations

Long-term effects of visualizations are affected by the immediate and short-term effects visualizations have on the student in individual occasions of tool use. For investigating these effects, we have developed a framework of the cognitive phenomena taking place during viewing. The model is shown in Figure 1. Students' engagement and representation used by the visualization tool are the varying factors in the model. They guide visual attention between different parts of a tool's interface, determining what kind of information is available to the student at what time. Engagement presumably directs visual attention to the parts of the interface that provide information considered useful by the student in that situation. For example, when a student is requested to view the visualization with a given task, the student's visual attention seeks its way to the parts of the screen that help in performing the task. Alternatively, if engagement is quite low, for example when a student is asked simply to view a visualization, it encourages random browsing of the different parts of the screen. Representation, on the other hand, directs visual attention in a more immediate way; for example movement on the screen often draws immediate visual attention to itself, also when we do not even come aware of it.

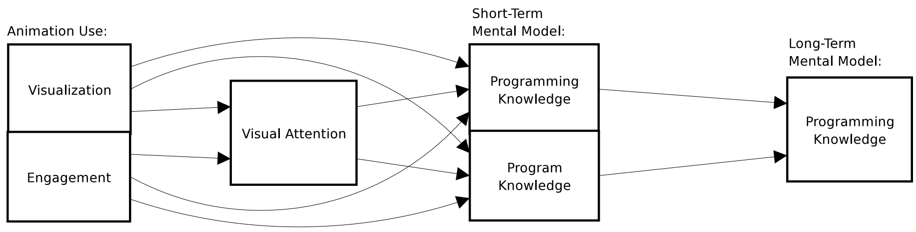

Fig. 1. The relationships between animation, visual attention, and mental models.

Even though the location of visual attention does not always correspond to what a person is focusing her thoughts on, it provides a strong indication on what a person chooses to target her interest on during viewing. Moreover, if some parts of the visualization tool's interface do not manage to capture viewer's attention, information from those parts does not manage to affect the formation of the viewer's mental model.

Short-term mental models of program and programming knowledge are affected by the information filtered from the visualization tool's interface by the distribution of visual attention. The mental models are affected also by the type and level of engagement, which reflect the student's commitment to the task, and this way the strength of mental processing; and by the representation of the visualizations, which can carry either useful, irrelevant or even harmful information for the creation of correct and meaningful shot-term mental models.

The final long-term programming knowledge is a combination of the short-term mental models of both program and programming knowledge; built up of short-term information from many tool use sessions and unconscious memory elaboration. General programming level observations made by a student during viewing influence her long-term programming knowledge directly. Even though details of individual programs are forgotten, short-term program knowledge affects the final long-term knowledge [4, 5, 13, 14]. The model is presented in more detail in [12].

2.2 The Research

We will investigate the above described phenomena and their interactions in detail by a series of empirical experiments. This way we collect information of the immediate and short-term effects of visualizations, and how they are accumulated into long-term effects.

Representation of visualizations can consist of several factors, including shape, size, color, texture, position, and orientation [1]. Visualizations can also utilize graphics, animation and sounds. Engagement with a visualization tool can range from viewing to actually constructing the visualizations and even to presenting the visualization to an audience [10]. Due to practical reasons, we will not investigate the interactions between each of these factors separately. Instead, we have chosen a subset of the factors we find most promising for achieving an extensive overview of the phenomena.

In our research, we will use a visualization tool, PlanAni [16], for the visualizations. PlanAni visualizes roles of variables [15] and operations on the variables during program execution, and its interface is presented in Figure 2. Variable roles are visualized with role images, which illustrate the central behavioural aspects of the roles. For example, a fixed value is visualized with a tombstone, since its value is never changed after initialization. The effects of PlanAni on long-term programming knowledge have been studied by Sajaniemi and Kuittinen [17], and Byckling and Sajaniemi [3] with a class-room experiment, in which students were divided into three even groups. The first group received traditional teaching and used the Turbo Pascal debugger in exercises, the second group received role-based teaching and used also the Turbo Pascal debugger in exercises, and the third group received role-based teaching and used the PlanAni program animator in exercises.

The results indicated that PlanAni users had better programming skills and their mental representations of (new, not animated) programs were different from that of Turbo Pascal debugger users. Mental representations of PlanAni users were, in fact, similar to that of good code comprehenders.

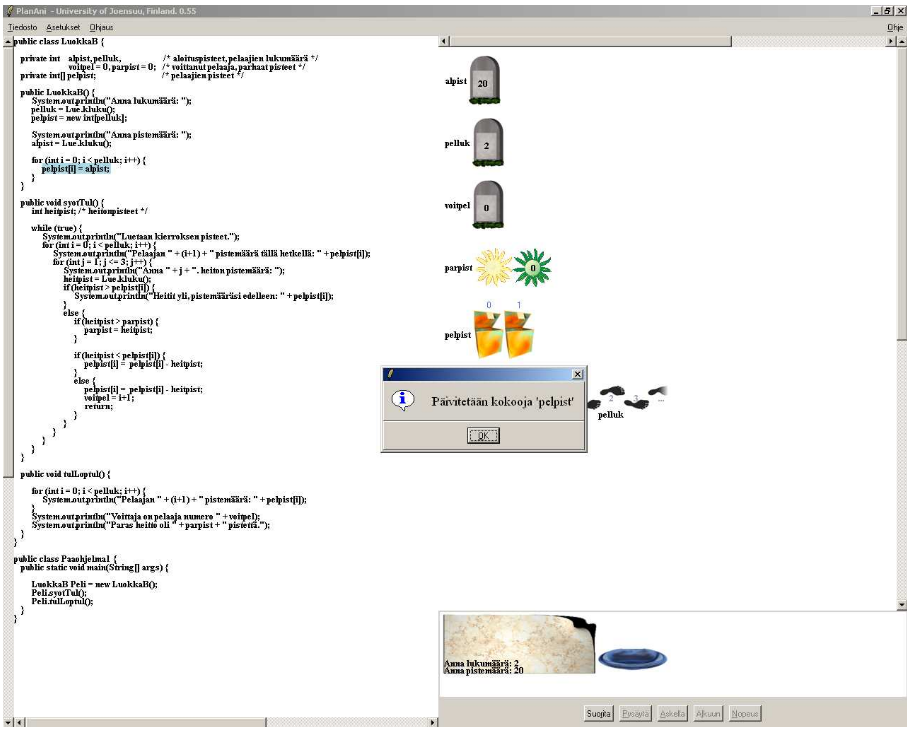

Fig. 2. PlanAni graphical user interface.

The effects of the content of visualizations on short-term mental model of roles were investigated by Stützle and Sajaniemi [19]. In their experiment, PlanAni's original role images were compared with neutral control images; engagement was not varied and visual attention was not measured. The results indicated that the use of the original role images enhanced learning of role knowledge when compared with neutral control images.

In an experiment carried out by Nevalainen and Sajaniemi [11], visualizations were either graphical and animated (PlanAni) or textual and static (Turbo Pascal). The target of the investigation was the effect of the content of visualizations on visual attention, and on short-term mental model of a studied program. The results showed a clear difference in the targeting of visual attention between the two tools. With the graphical tool, visual attention was targeted much more to the variables, and the increase of visual attention to variables increased the proportion of high-level information in program summaries and decreased the proportion of low-level code-related information[1].

Nevalainen and Sajaniemi [12] investigated also the effects of presence or absence of animation. In the experiment, two versions of PlanAni were used; the first version contained all the features of PlanAni, including role-based animations and notifications, which were lacking from the second version. Visual attention, and short-term mental models concerning program and programming knowledge were investigated. The results showed that the participants spent only little time viewing variables in both groups. Instead, the participants resorted heavily to the textual cues, even when rich visual information in the form of animations was provided.

3 Experiment

In the earlier experiments [11, 12, 19], the type of engagement has been kept fixed and only the representation used for the variables has been varied. Collected information has been related to the location and size of the visualizations, and to presence or absence of role animation. The effects of presence or absence of role images have not been investigated. In the current experiment, we focused our attention on both viewer's engagement with the visualization tool and the presence or absence of role images, and their influence on the immediate and short-term effects.

3.1 Method

The experiment was a 2*2 between-subject design with two independent variables: the task to be performed during viewing (engagement) and the version of PlanAni (representation). *Group ID (Images, Data flow)* used PlanAni with role images and role animation to perform a task which requires information mainly of program's data flow. *Group IC (Images, Control flow)* used PlanAni with role images and role animation to perform a task which requires information mainly of program' s control flow. *Group TD (Text, Data flow)* used PlanAni without role images and role animation to perform

[1] In the described experiment, the analysis scheme presented in [6] was used to divide the summaries into information types, which were further divided into high-level information or low-level information according to the level of abstraction.

a task which requires information mainly of program’ s data flow. *Group TC (Text, Control flow)* used PlanAni without role images and role animation to perform a task which requires information mainly of program’ s control flow. The pre-test score on role knowledge (applied from [19]) was used to divide the participants evenly into the groups.

The dependent variables were the locations of the participant’s gaze (visual attention), the participant’s post-test score on role knowledge (short-term programming knowledge), and the program summary provided by the participant (short-term program knowledge). A Tobii eye-tracking camera [20] was used to measure the locations of gaze. A post-test from [19] was used to measure the knowledge on variable roles. Good’s program summary analysis scheme [5] with the additional categories presented by Byckling et al. [2] was used to analyze the program summaries.

Participants Twentyfour participants, 20 male and 4 female, were students recruited from a university-level introductory Java programming course. Participation was voluntary and the participants received a fee of 15 euros.

Materials The pre-test consisted of written descriptions of all roles and examples of their use, and three small Java programs with 14 variables, whose roles the participants were asked to determine. The post-test consisted of four small Java programs with 19 variables. The post-test material did not contain role descriptions. The names of the roles were, however, written below each program.

Two versions of PlanAni were used. The first version (*version I*) was a normal PlanAni with role images and animations. In the second version (*version T*), role images and animations were removed, but textual variable names were retained. Both versions were prepared so that participants were able to execute each program once, step by step. PlanAni covered the entire screen of a 1280*1024 resolution display.

One practice program and one actual program were used. Both programs were short Java programs, consisting of one class and a main method. During the experiment, participants performed one of the two tasks. *Task D* (data flow task) was to predict the values that variables get during program execution. Required information to perform the task is shown explicitly in the visualizations that show the values of the variables and assignments. *Task C* (control flow task) was to give the program such inputs that a given variable will have a given value at the end of the program execution. Information about the program’s control flow, explicitly shown in the program code, is needed to perform the task correctly.

Program summaries of the participants were collected using a program summary form, which asked the participants to “Describe the functionality of the program in your own words, or tell what happened in the program during execution”. Participants’ opinions of PlanAni were gathered using an evaluation form that included Likert scale questions and open questions about the tool and its use.

Procedure The participants were run individually, and individual experimental situations lasted between 1,5 and 2 hours. In the first phase, the participant was given a pre-test on role knowledge. The time was limited to 15 minutes. At the point of ten minutes, the participant was informed that she had five minutes left.

In the second phase, the participant was seated in front of an eye-tracking equipment, Tobii eye-tracking device 501, where the camera is embedded in the panels of

Table 1. Scores of pre- and post-tests on role knowledge.

	Group			
	ID	IC	TD	TC
Pre-test score & SD	8.50 2.35	8.17 3.19	9.00 3.46	8.33 2.88
Post-test Score & SD	15.50 2.43	12.33 1.86	16.00 2.83	14.83 2.64

Table 2. Mean viewing times (minutes, seconds).

	Group			
	ID	IC	TD	TC
Time & SD	17.29 2.03	18.48 6.46	15.39 1.02	16.52 3.55

the monitor, and the camera was calibrated. During the calibration, the pre-test score was analyzed, and the participant was assigned to an appropriate group. After this, the practice program was shown to the participant. Then, the participant was given one of the two tasks to perform, and the actual program was shown to the participant while the locations of participant's gaze were recorded. Both programs were run one time, and the participant was allowed to adjust the speed of the execution by clicking the Ok button of the notifications. The time to view the programs was not limited. Depending on the task given to the participant, the inputs were given (task D) or were chosen freely by the participant (task C).

In the third phase, after viewing the actual program, the program was dismissed from the screen, and the participant was given a program summary form to fill; time to fill the form was not limited.

In the fourth phase, the participant was asked to perform a post-test on role knowledge. The time to perform the test was limited to ten minutes. At the point of eight minutes, the participant was informed that she had two minutes left.

In the fifth phase, the participant was given an unlimited time to fill the tool evaluation form.

3.2 Results

The results of the pre-test on role knowledge were used to divide the participants to the four groups. The maximum score in the pre-test was 14 points. The mean scores and standard deviation for the groups are presented in Table 1. The differences between the groups were analyzed using two way between-subjects ANOVA, and they were found to be statistically non-significant.

The maximum score in the post-test on role knowledge was 19. The mean scores and standard deviation for the groups are presented in Table 1. The differences between the groups were analyzed using again two way between-subjects ANOVA. In post-test, the difference between the two tasks (groups ID & TD versus groups IC & TC) was found to be statistically significant ($F(1,\ 20) = 4.630$, $p\ <\ 0.05$).

The mean times used to study the actual program for each group are presented in Table 2. The differences between the groups were analyzed using two way between-subjects ANOVA, and were found to be statistically non-significant.

Table 3. Proportional mean viewing times and standard deviations on the different areas of the screen.

Code	Screen Area	Group							
		ID		IC		TD		TC	
		Time	SD	Time	SD	Time	SD	Time	SD
COD	Code	56.94	8.20	60.76	9.12	43.57	20.39	57.86	13.34
VAR	Variables	12.23	4.57	13.79	9.47	16.54	9.57	7.57	2.40
NOT	Notifications	21.85	4.38	18.99	5.52	29.35	10.44	22.64	8.18
IO	Input and Output	6.23	2.98	4.5	2.69	8.8	5.85	8.21	4.04
OTH	Other	2.75	2.79	1.97	0.57	1.74	2.01	3.73	3.56

The visual attention was analyzed by dividing the screen into five areas; code area, variable area, IO-area, notifications area, and other. The first three areas were formed by taking the smallest bounding box that includes all symbols within the areas. The notifications area was formed by the bounding box that surrounds largest notifications message box. The other area consists of all other parts of the screen. Table 3 presents the mean proportions of viewing times on these five areas in the four groups. Differences in the distribution of visual attention between the groups were analyzed with three way mixed design ANOVA using Greenhouse-Geisser correction, and only the main effect of screen area was found to be statistically significant ($F(1.538,\ 30.758) = 132.471$, $p < 0.001$).

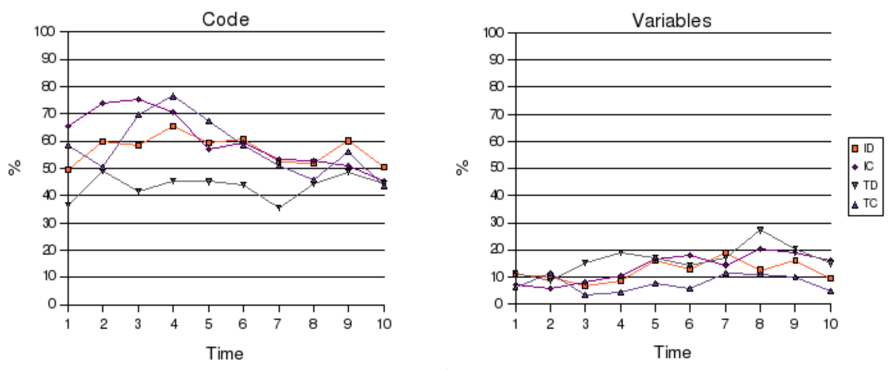

Fig. 3. The proportions of visual attention on code and variable area during different intervals in time.

To get a more detailed view of the changes in the distribution of visual attention of participants during viewing, we divided the viewing time to ten intervals. Figure 3 presents the proportions of visual attention on code and variable areas during the intervals of time in the four groups. The proportions of visual attention on the other three areas during the intervals did not differ significantly between the groups, and are not treated in detail. Differences between the groups were analyzed with three way mixed design ANOVA using Greenhouse-Geisser correction. In distributions related to variable area, only the main effect of interval was found to be statistically significant ($F(2.930, 58.598) = 3.245$, $p < 0.05$). In distributions related to code area, both the main effect of interval ($F(4.275, 85.505) = 5.487$, $p < 0.001$), and the interaction between interval and task type ($F(4.275, 85.505) = 2.921$, $p < 0.05$) were found to be statistically significant.

Table 4. Mean proportions of IT categories used in program summaries.

Code	Information Type	Group							
		ID		IC		TD		TC	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
FUN	Function	3.05	4.76	3.33	9.16	1.18	2.90	5.35	10.04
ACT	Actions	23.63	15.30	35.95	9.70	26.85	12.64	23.97	16.30
OPE	Operations	3.63	5.66	4.30	5.18	6.75	6.78	18.28	15.65
SHI	State-high	2.80	4.84	3.27	5.34	4.80	5.89	5.42	6.50
SLO	State-low	0.83	2.04	2.22	3.45	2.08	3.48	2.02	3.19
DAT	Data	40.87	22.52	32.88	14.90	36.78	15.93	33.13	14.89
CON	Control	0.98	2.41	4.05	6.42	1.88	3.06	7.401	4.19
ELA	Elaborate	9.75	8.34	11.25	10.22	7.63	11.90	3.20	3.59
MET	Meta	0.00	0.00	1.03	2.53	0.70	1.71	0.00	0.00
IRR	Irrelevant	10.00	20.00	1.67	4.08	6.45	7.42	1.18	2.90
UNC	Unclear	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
INC	Incomplete	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CUT	Continuation	4.47	5.38	0.00	0.00	4.87	7.66	0.00	0.00
HIG	FUN+ACT+SHI+DAT	70.35	27.46	75.43	14.42	69.65	23.43	67.88	23.50
LOW	OPE+SLO+CON	5.43	8.46	10.60	10.55	10.72	9.52	27.72	19.39
OTH	100-HIG-LOW	24.20	24.95	13.95	12.35	19.65	20.82	4.40	5.72
HIP	HIG / (HIG+LOW) * 100	92.30	11.93	87.73	11.02	85.52	11.98	70.25	21.25
DAT	SLO+SHI+DAT	44.50	20.83	38.37	16.12	43.67	14.56	40.57	16.44
CON	ACT+OPE+CON	28.25	15.35	44.30	11.53	35.48	11.46	49.65	10.18
OTH	100-DAT-CON	27.25	27.74	17.33	16.37	20.85	19.60	9.78	10.86
DAP	DAT / (DAT+CON) * 100	61.55	12.78	45.47	14.16	54.56	11.82	44.03	13.61

Good's program summary analysis scheme [5] was used to study participants' mental models of the studied programs. For our analysis, we included additional categories presented in [2]. The scheme contains two classifications; The information types classification (IT) is used to code summary statements on the basis of the information types they contain, while the object descriptions classification (ODC) looks at the way in which objects are described. The distribution of information type statements in each group is presented in Table 4, and the distribution of object description statements in each group is presented in Table 5.

The distributions in information types were analyzed with three way mixed design ANOVA using Greenhouse-Geisser correction, and the main effect of IT statements

Table 5. Mean proportions of ODC categories used in program summaries.

Code	Object Description Category	Group							
		ID		IC		TD		TC	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
PON	Program only	2.17	3.44	8.32	10.99	0.98	2.41	7.25	6.74
PRO	Program	0.00	0.00	5.08	8.78	0.00	0.00	2.38	5.84
PRR	Program—real-world	9.77	10.48	13.95	8.91	11.63	10.21	18.38	22.21
PRD	Program—domain	1.28	3.14	0.00	0.00	0.72	1.76	0.00	0.00
DOM	Domain	78.02	19.56	65.73	29.78	82.22	14.04	68.40	27.39
IND	Indirect reference	8.78	11.15	6.93	11.50	4.45	8.08	3.55	6.27
UNO	Unclear	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

was found to be statistically significant ($F(3.480,\ 69.590) = 39.058$, $p\ <\ 0.001$). We further grouped individual information types using two criterias; high versus low, and data versus control. Both distributions are presented in Table 4. The distributions were analyzed with three way mixed design ANOVAs using Greenhouse-Geisser correction. The main effect of information type statements was found to be statistically significant in both high versus low -grouping ($F(1.562,\ 31.234) = 84.189$, $p\ <\ 0.001$), and in data versus control -grouping ($F(1.796,\ 35.915) = 14.124$, $p\ <\ 0.001$).

The distributions in object description categories were analyzed with three way mixed design ANOVA using Greenhouse-Geisser correction. The main effect of object description categories was found to be statistically significant ($F(1.523,\ 30.463) = 119.285$, $p\ <\ 0.001$).

Table 6. Participants' evaluation of the visualisation tool (scale 1-5); the best is 5.

Charasteristic	Group							
	ID		IC		TD		TC	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Originality	4.17	0.75	3.50	0.55	3.50	0.55	3.33	0.82
Pleasure	2.67	0.82	3.17	1.17	2.83	0.98	3.17	0.74
Salience	3.50	0.84	3.50	0.84	4.00	0.89	4.67	0.52
Understandability	4.33	0.82	4.00	0.89	4.67	0.52	4.33	0.52
Usefulness	2.50	0.55	2.50	1.22	2.83	1.33	2.33	1.03

Table 6 presents the participants' evaluations of the visualization tools. Participants were asked to evaluate each charasteristic with one question. For example, the understandability of the visualizations was evaluated by proposition "I found this representation easy to understand". The differences in scores of different evaluation categories between the groups were analyzed using three way mixed design ANOVA using Greenhouse-Geisser correction. The main effect of different evaluation categories was found to be statistically significant ($F(2.285,\ 44.705) = 21.961$, $p\ <\ 0.001$).

4 Discussion

In the experiment, we gave the participants either a task which requires information of program' s data flow (D), or a task which requires information of program' s control flow (C). We also used two versions of PlanAni. First version (I) contained the full functionality of PlanAni. In the second version (T), role images and role-based animations were removed. We measured the locations of viewer's gaze on the screen, and gathered program summaries from the participants, which we further analyzed using Good's program summary analysis scheme. We also measured the participants' role knowledge with a post-test. This way, we were looking to find out what effects the type of engagement and the representation of the visualization tool have on visual attention, short-term program knowledge and short-term programming knowledge.

When participants were asked to evaluate PlanAni, answers did not differ significantly between the two versions. The version containing images was found to be more original, while the textual version was more understandable and in a more salient role for learning the program. Both versions were judged equally pleasurable and useful. The mixed results indicate that it is probably not useful to look for differences between the effects of the tools from the participants' conscious grading of the tool. Instead, we focused on the measured differences between the groups in their interaction with the tool.

Regardless of the given task or the visualization tool used, the participants in different groups used approximately the same **time to view the program** (15 minutes 39 seconds – 17 minutes 29 seconds). Visual attention was focused most of the time on the code area (43.57 – 60.76%), whereas variable visualizations were looked at only between 7,57 – 16,54% of the total time. The overall distribution of **visual attention** did not differ significantly between the groups.

These observations are similar to our previous experiment in which we compared animated and static variable visualizations in PlanAni. Both results are, however, radically different from the experiment in which PlanAni was compared with Turbo Pascal debugger [11], in which variable visualizations differ in location and size from that of any of the PlanAni versions used in the experiments. This seems to suggest that differences in the location and size of the visualizations area have more influence on the distribution of visual attention than the content of visualization area itself.

When the distribution of visual attention is examined at different intervals in time (Figure 3), differences between the groups emerge. When the distribution of visual attention of the participants performing task D (groups ID and TD) remains similar throughout the time, the distribution of visual attention of the participants performing task C (groups IC and TC) changes notably between the first and second half of the viewing. These participants focus their visual attention at the beginning mostly on the code, and increase their visual attention to the other parts of the visualization on the second half of the viewing, while decreasing visual attention on the code area at the same time.

Task C was control flow task and asked the participants to enter such inputs that a given variable will have a given value at the end of the execution. The participants seemed to search the necessary information from the program code first, and focused their visual attention on the other parts of the screen only after that. Many of the participants performing task C also spent some time simply viewing the program code while animation was stopped. This happened either before starting the animation or alternatively before providing the first input to the program, and was probably due to the fact that students felt it more comfortable to determine the correct inputs with pencil and paper before animation than during it. These students seemed to use pencil and paper as a problem solving tool, and the animator as a verification tool for their answer.

No clear differences between the groups in the content of the **mental models of the studied programs (i.e., program knowledge)** were found. The different styles of engagement did not produce significant differences; neither did the variation in the graphical richness of the visualizations. The results are similar to [11], and [12]. Even though PlanAni has been found to have positive long-term effects on programming skills and mental models of (new) programs when used for a longer period [17, 3], the differences in mental models of programs do not seem to manifest themselves in individual occasions of using the tool.

The different styles of engagement did, however, produce significant differences in the amount of **role knowledge (i.e., programming knowledge)** measured after the viewing, while the variation in the representation of the visualizations did not. This is somewhat contradictory to the experiment by Stützle and Sajaniemi [19], where role images did produce significant differences in role knowledge when compared with neutral control images. One possible explanation is that the absence of role images in itself does not have a negative impact on the adoption of role knowledge, at least when notifications provide similar information in textual form. Instead, neutral control images, that do not convey role information, may introduce interference that manifests itself in scores of the role knowledge post-test.

The participants performing task D performed clearly better in the post-test on role knowledge. These participants also divided their visual attention more evenly during the viewing. Moreover, the group scoring highest in the post-test (group TD), also had by far most even distribution of visual attention between the five screen areas. The participants who focused their attention most evenly throughout the whole viewing time and between all the screen areas, benefited most from the tool, at least regarding role knowledge.

The signifigance of the style of engagement for the beneficiality of a visualization tool has been speculated for example by Hundhausen and Douglas [7], Hundhausen et al. [8], and Naps et al. [10]. The differences between the two tasks in our results provide support for the notion that the style of engagement is an important factor in the overall influence of a visualization tool on learning. Naps et al.[10] have presented a taxonomy, in which our two tasks seem to fall into the same category (responding), that is, even inside this category, different tasks produce different effects. Our results indicate that the beneficiality of the visualization tool for learning dependents not only on the depth of the engagement (for example viewing versus constructing visualizations), but also on the nature of the engagement (for example tasks requiring data flow versus control flow oriented information).

5 Conclusion

We are carrying out a series of experiments based on a model of cognitive phenomena taking placing during visualization viewing sessions. In the presented experiment, we gave the participants either a data flow task (D) or a control flow task (C). We also varied representation by using two versions of program animator; one with role images and animation, and one without. We then measured how this variation affects visual attention, short-term program knowledge, and short-term programming knowledge.

The results showed that the overall distribution of visual attention did not differ significantly between the groups. However, when distribution of visual attention was analyzed during different intervals in time, differences between the groups emerged. When the distribution of visual attention of the participants performing data-related task remained similar throughout the time, the participants performing control-related task focused their visual attention at the beginning almost solely on the code, and increased their visual attention to the other parts of the visualization on the second half of the viewing. The participants performing data-related task also scored higher in the post-test on role knowledge, benefiting most from the tool.

Analysis of the results together with the results from earlier experiments on visualizing roles [11, 12, 19] indicates that the location and size of visualization area influences visual attention more clearly than presence or absence of images or animation. Engagement changes the viewing patterns of the participants even when the overall distribution remains the same, and affects the role knowledge (programming knowledge), which is also affected by the content of role images. Differences in program knowledge between different conditions or groups in any of the experiments are modest, which may be due to the large amount of time spent on viewing the available code in all conditions and groups.

Results of the current experiment indicate that engagement plays an important role in beneficiality of the visualization tool for learning. Benefits are not dependent only on the depth of the engagement (for example viewing versus constructing visualizations), but also on the nature of the engagement (for example task requiring data flow versus control flow oriented information).

6 Acknowledgments

This work was supported by the Academy of Finland under grant number 206574.

References

1. J. Bertin. *Semiology of Graphics*. University of Wisconsin Press, 1983.
2. P. Byckling, M. Kuittinen, S. Nevalainen, and J. Sajaniemi. An inter-rater reliability analysis of good's program summary analysis scheme. In *Proceedings of the 16th Annual Workshop of the Psychology of Programming Interest Group (PPIG 2004)*, pages 170–184, 2004.

3. P. Byckling and J. Sajaniemi. Roles of variables and programming skills improvement. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2006)*, pages 413–417, New York, NY, USA, 2006. Association for Computing Machinery.
4. K. Ehrlich and E. Soloway. An empirical investigation of the tacit plan knowledge in programming. In J. C. Thomas and M. L. Schneider, editors, *Human Factors in Computer Systems*, pages 113–133. Norwood, NJ: Ablex Publishing Company, 1984.
5. J. Good. *Programming Paradigms, Information Types and Graphical Representations: Empirical Investigations of Novice Program Comprehension*. PhD thesis, University of Edinburgh, 1999.
6. J. Good and P. Brna. Program comprehension and authentic measurement: A scheme for analysing descriptions of programs. *International Journal of Human-Computer Studies*, 61:169–185, 2004.
7. C. D. Hundhausen and S. A. Douglas. Using visualizations to learn algorithms: Should students construct their own, or view an expert's? In *the 2000 IEEE International Symposium on Visual Languages (VL'00)*, pages 21–28, 2000.
8. C. D. Hundhausen, S. A. Douglas, and J. T. Stasko. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages and Computing*, 13:259–290, 2002.
9. A. W. Lawrence. *Empirical studies of the value of algorithm animation in algorithm understanding*. PhD thesis, Georgia Institute of Technology, 1993.
10. T. L. Naps, G. Rling, V. Almstrum, W. Dann, R. Fleischer, C., Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger, and J. . Velzquez-Iturbide. A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. In *Working Group Reports From ITiCSE on innovation and Technology in Computer Science Education (Aarhus, Denmark, June 24 - 28, 2002). E-WGR '02*, pages 131–152. Association for Computer Machinery, 2002.
11. S. Nevalainen and J. Sajaniemi. Short-term effects of graphical versus textual visualization of variables on program perception. In P. Romero, J. Good, S. Bryant, and E. A. Chaparro, editors, *Proceedings of the 17th Annual Workshop of the Psychology of Programming Interest Group (PPIG 2005)*, pages 77–91, 2005.
12. S. Nevalainen and J. Sajaniemi. An experiment on short-term effects of animated versus static visualization of operations on program perception. In *Proceedings of the 2nd International Computing Education Research Workshop (ICER 2006)*, pages 7–16. Association for Computer Machinery, 2006.
13. N. Pennington. Comprehension strategies in programming. In G. M. Olson, S. Sheppard, and E. Soloway, editors, *Empirical Studies of Programmers: Second Workshop*, pages 100–113. Norwood, NJ: Ablex Publishing Company, 1987.
14. R. S. Rist. Schema creation in programming. *Cognitive Science*, 13:389–414, 1989.
15. J. Sajaniemi. An empirical analysis of roles of variables in novice-level procedural programs. In *Proceedings of IEEE 2002 Symposia on Human Centric Computing Languages and Environments (HCC'02)*, pages 37–39, Washington, DC, USA, 2002. IEEE Computer Society.
16. J. Sajaniemi and M. Kuittinen. Visualizing roles of variables in program animation. *Information Visualization*, 3:137–153, 2004.
17. J. Sajaniemi and M. Kuittinen. An experiment on using roles of variables in teaching introductory programming. *Computer Science Education*, 15(1):59–82, 2005.
18. J. T. Stasko, A. Badre, and C. Lewis. Do algorithm animation assist learning? an empirical study and analysis. In *Proceedings of the SIGCHI conference on Human factors in computing systems (ACM INTERCHI'93)*, pages 61–66. ACM Press, 1993.
19. T. Stützle and J. Sajaniemi. An empirical evaluation of visual metaphors in the animation of roles of variables. *Informing Science Journal*, 8:87–100, 2005.

20. Tobii. *User Manual - Tobii Eye Tracker, Clearview Analysis Software*. Tobii Technology AB, 2004.