Are Visualization Tools Used in Programming Education?

By Whom, How, Why, and Why Not?

Essi Isohanni
Department of Pervasive Computing
Tampere University of Technology
Tampere, Finland
essi.isohanni@tut.fi

Hannu-Matti Järvinen
Department of Pervasive Computing
Tampere University of Technology
Tampere, Finland
hannu-matti.jarvinen@tut.fi

ABSTRACT

Visualization tool developers and researchers deliberate on the future directions of tool design and research. Sometimes, it is argued that visualization tools are not used widely. However, there is no factual knowledge about the current rate of visualization tool usage in programming education. How widely are visualization tools used in classes? Who uses them? How, why or why not?

This work studies the current usage of visualizations in order to answer the above mentioned questions. The aim is to provide facts to lay the ground for discussion on the future directions on visualization tool development and research.

A worldwide survey was organized for over 250 teachers of programming. We targetted the teachers of programming in general, not only those interested in visualization technologies, and asked if they use visualization tools in their class, how they use them and their reasons for and against using them. This paper discusses the results related to almost 500 programming courses giving a current overview of the state of the field.

Regular use of visualization tools in class is rather rare: approximately 20% of programming courses use software visualizations regularly. Contrary to the recommendations on visualization tool usage, most commonly they are used by the teacher, not by the students. The most often mentioned reason for not using visualization tools was that the teacher prefers to create his/her own visualizations in some other way, e.g., using the blackboard.

Categories and Subject Descriptors

K.3.2 [**Computing Milieux**]: Computers and Education—*Computer and Information Science Education*

Keywords

Programming, Software visualization, Computing education

Koli Calling '14, November 20–23, 2014, Koli, Finland.
Copyright 2014 ACM 978-1-4503-3065-7/14/11 ...$15.00.
http://dx.doi.org/10.1145/2674683.2674688.

1. INTRODUCTION

Software visualization in programming education is a widely studied topic [1, 4, 5, 9, 13, 12]. There are multiple software visualization tools available for educational purposes and numerous studies on their usage. In 2002, Ross and Grindner [7] stated that "Although appealing and often useful, the anecdotal evidence is that these visualizations are seldom used in the classroom." Several other authors have also expressed similar views, for instance Hundhausen et al. [1] as follows: "visualization technology has failed to catch on in the mainstream computer science education". Is this "anecdotal evidence" valid? Does it hold nowadays?

There are new state of the field reports on algorithm visualizations analysing the quality of the visualizations and giving recommendations on how they should be improved [9, 10]. However, even new work related to the state of classroom usage of visualization tools does not reveal accurately how widely visualization tools are used in programming education [8].

There are also current openings to bring theoretical approaches into the field of software visualization [2, 3, 11]. It is debated, how research on software visualizations in programming education should be developed. However, before this it would be good to know what the state of the field is at the moment: to what extend are visualization tools used at the moment, how they are used and what are the reasons for or against using them.

In this work, we conducted a survey that reached a large group of programming teachers world wide. We asked them about the visualization tool usage in their programming courses and gained insights into almost 500 programming courses.

The paper is organized as follows: Section 2 presents the design of our survey in relation to some related work. Results are presented in Section 3 and discussed in Section 4. Section 5 concludes with answers to the questions.

2. SURVEY DESIGN & METHODOLOGY

The earlier questionnaires in this area typically concentrated only on the use of visualizations [4, Section 2.2]. We suppose that for a respondent who is not interested in visualization technologies, it is more likely to decide not to respond if the questionnaire specifically handles the use of visualizations than when the questionnaire handles teaching programming in general. Our questionnaire addressed the use of programming languages in programming courses and after this we had questions about the use of visualization tools. The questions related to the use of programming languages are relevant to all teachers of programming. Therefore, we think that our questionnaire reached all kinds of programming teachers, not just the ones interested in the use of visualization tools.

The survey was implemented as a web form. We distributed the link to the survey firstly at Koli Calling conference in 2012, and then using both international and national mailing lists by the help of several colleagues. The survey was open from November 2012 until April 2013, when the number of new responses per week dropped to zero.

The questionnaire specified that by the term *software visualization tools* we mean all kinds of educational software tools for displaying and visualizing the execution of programs or algorithms. It also defined that the time scale of interest is over the years 2009-2012.

Software visualizations are commonly divided into two main categories: program visualizations (PV) that are typically used in introductory programming courses, and algorithm visualizations (AV) that are used in teaching datastructures and algorithms. However, there are also other types of visualizations, for instance, some visualization tools specifically illustrate concurrency. Therefore, the questionnaire divided courses roughly into five categories: basic programming, advanced programming, datastructures and algorithms, concurrent programming, and finally other programming courses to let the respondents indicate anything that is relevant.

The questionnaire started with a section related to the background information about the respondent and the courses he/she is teaching. For the age groups of students, we gave the options “under 13”, “13-18” (later referred to as “teenagers”), and “over 18” (later referred to as “adults”) and asked the teacher to choose to age group he/she is primarily teaching.

Next, there were separate sections for different types of programming courses. The respondent was only inquired about the types of courses he/she indicated in the background information. For example, if a teacher indicated he teaches introductory programming and concurrent programming, he/she was given the same set of questions first regarding introductory programming and then regarding concurrent programming. Therefore, the number of responses in the tables presenting the results is higher than the number of respondents.

In the section related to a certain course category there were two multiple choice questions: “Were software visualization tools used in the programming course(s) either by teachers, students or both?” and “How were software visualization tools used during the programming course(s)?”. The answer options of the questions can be seen in the left columns of Tables 1 and 3. After multiple choice questions, there was an open field for writing the names of the visualization tools that were used. At the end of each section, we asked the teachers reasons for using and not using a visualization tool. We provided a list of possible reasons where one can tick all that apply (options listed in the left column of Tables 4 and 5) and then an open field to list any reasons he/she has in mind. The predefined options were collected from earlier surveys and other literature related to visualizations, e.g., [4, Section 2.2].

3. RESULTS

Altogether 255 teachers from 33 countries answered the questionnaire. The respondents represent both genders and different age groups. Most of the teachers responded regarding adult students (181 respondents) and almost all the rest regarding teenagers (57 respondents), but we also had 8 responses from teachers teaching programming for younger children. We think that this represents quite a normal distribution of the teachers of programming. Since the last student age group is so much smaller than the other groups, it was not possible to compare it with the other groups. However, in the comparisons between the different course categories, all the responses are included.

3.1 Rate of Visualization Usage

The results to the question “Were software visualization tools used in the programming course(s) either by teachers, students or both?” are presented in Tables 1 and 2. Visualizations are not used at all in almost half (45.6%) of the programming courses. Slightly more than 20% of programming courses use visualizations regularly. There is a statistically significant difference ($p<.01$) between courses targeted for adult students and younger ones. The use of visualizations is more common with younger students.

As we expected in the survey design, the course categories using visualizations the most are basic programming courses (PVs) and datastructures and algorithms courses (AVs). Also advanced programming courses and other programming courses take advantage of visualizations but in courses related to concurrent programming the use of visualizations is clearly rarer.

3.2 Ways of Visualization Usage

The percentages of different ways of using visualizations are presented in Table 3. There was a similar trend in all the course categories and age groups: the most common ways of using visualizations were “Teacher(s) demonstrated the usage” and “Teacher(s) used them in teaching”. It seems that visualizations are most often used as a means of communication when the teacher explains things for students.

As the use of visualizations is more common when teaching teenagers also all the ways of using them were more common in this age group.

3.3 Reasons for Using Visualization Tools

Table 4 presents the percentages of respondents who ticked the predefined answer options in the question regarding their reasons for using visualization tools in programming courses. There is a clear difference between the age groups: the teachers teaching teenagers have higher percentage values in most of the options, which is natural because they also more often used visualizations.

The most often marked reason for using visualizations was that they find visualization tools educationally effective. At least half of the teachers teaching teenage students had marked it in all the course categories.

In the open-ended question for specifying the other reasons, respondents had mentioned reasons like:

- The tool was already there (part of the programming environment, installed by the administrator, etc.).
- The tools provide something useful (more or instant feedback to students, automatic assessment, etc.).

Table 1: Usage of visualizations grouped by different course categories. The question in the web form was: "Were SV tools used in the programming course(s) either by teachers, students or both?"

	Basic programming		Advanced programming		Datastructures and algorithms		Concurrent programming		Other progr. courses		All	
	N	%	N	%	N	%	N	%	N	%	N	%
Not at all	72	37.5	67	53.2	41	39.4	19	76.0	25	56.8	224	45.6
Occasionally	65	33.9	41	32.5	40	38.5	5	20.0	10	22.7	161	32.8
Regularly	55	28.6	18	14.3	23	22.1	1	4.0	9	20.5	106	21.6
	192	100.0	126	100.0	104	100.0	25	100.0	44	100.0	491	100.0

Table 2: Usage of visualizations compared between different age groups. The groups that contain five or less responses are left out from the table.

	Students over 18 years										Students 13-18 years					
	Basic progr.		Advanced progr.		Datastr. & alg.		Concurrent progr.		Other progr.		Basic progr.		Advanced progr.		Data str. & alg.	
	N	%	N	%	N	%	N	%	N	%	N	%	N	%	N	%
Not at all	57	42.9	60	57.7	30	44.1	17	73.9	24	64.9	14	25.9	7	31.8	11	30.6
Occasionally	48	36.1	34	32.7	26	38.2	5	21.7	8	21.6	14	25.9	7	31.8	14	38.9
Regularly	28	21.1	10	9.6	12	17.6	1	4.3	5	13.5	26	48.1	8	36.4	11	30.6
	133	100.0	104	100.0	68	100.0	23	100.0	37	100.0	54	100.0	22	100.0	36	100.0

- The tools are free.
- The students are used to them in other courses.

These reasons are mostly not related to the content of the visualization but rather other features or properties of the visualization tool.

3.4 Reasons Against Use of Visualization Tools

Table 5 presents the percentages of respondents who ticked the predefined answer options in the question regarding their reasons for not using visualization tools in programming courses. In all course categories and age groups, clearly the most often ticked reason for not using visualization tools in teaching was "I prefer to create my own visualizations, e.g. by drawing pictures on the black board." About a fifth (21.4%) of the respondents had marked it. The same content was written also in the open field in many different wordings.

This question seemed more thought-provoking than the question regarding the reasons for using visualizations since the amount of open answers was much higher. However, the answers specifying the other reasons partly repeated the predefined options in other wordings. New reasons written by the respondents were:

- The teacher prefers some other means (learning by doing, concretizing/visualizing in some other way, etc.).
- Tools' limitations.
- There is no need for visualizations (students grasp things otherwise, teacher can demonstrate things otherwise, visualizations are not relevant to the topic, etc.).
- There is no time to revise teaching materials.
- I have not thought about it.
- Never knew visualization tools exist.

The teachers in advanced programming courses had ticked the reasons "I do not believe that visualizations are educationally effective in this level of programming courses" and "There are no visualization tools available for this topic" almost twice as often as the teachers of basic programming courses and courses on datastructures and algorithms. The teachers of advanced programming courses also ticked the reason "I prefer to create my own visualizations, e.g. by drawing pictures on the black board" clearly more often than the teachers of basic programming and datastructures and algorithms.

4. DISCUSSION

Our questionnaire reached a relatively large group of teachers from 33 countries. We also assume that the respondents of our survey represent the teachers of programming rather generally, because the questionnaire was targeted to all teachers (included the section with common programming question). Therefore, the visualization tool usage rate we report, should be close to the real usage of visualization tools between 2009-2012.

4.1 Comparison with Other Surveys

Table 6 shows a comparison of the visualization usage rate with the results that Naps et al. [4, Section 2.2] reported in 2002. Our questionnaire did not use exactly the same wording as theirs, but we see that our phrase "regularly" compares with their phrases "Almost every day" and "Once per week" and our phrase "Occasionally" to their phrase "A few times per term". The usage rate that we report is notably lower than the rate reported by Naps et al. We think that the reason causing this difference is that our questionnaire was targeted for all teachers of programming and their questionnaire addressed specifically the use of visualizations and therefore reached mainly respondents interested in visualization technologies.

Table 3: Distribution of different ways of using visualizations in programming courses.

How were SV tools used during the programming course(s)?	All	Teaching adults	Teaching teenagers
Teacher(s) demonstrated the usage	36.3%	30.4%	51.7%
Teacher(s) used them in teaching	38.5%	33.4%	53.4%
Students in class, obligatory use	22.0%	14.2%	43.2%
Students in class by their choice	13.8%	11.0%	22.9%
Students at their own, obligatory use	14.5%	11.8%	22.9%
Students at their own by their choice	21.4%	20.8%	23.7%

Table 4: Distribution of answers to the reasons for using visualizations in programming courses.

Why were these tools selected for use in the programming course(s)?	All	Teaching adults	Teaching teenagers
I've found the tools educationally effective.	40.3%	35.1%	55.9%
They are easy to use.	34.6%	28.2%	51.7%
It's easy to make own visualization examples using these tools.	18.9%	14.0%	33.1%
There were good visualization examples available for these tools.	14.3%	12.6%	18.6%
Tools were easy to adapt to my teaching approach and/or course content.	20.2%	19.5%	22.0%
We developed the tools.	9.8%	12.6%	1.7%
Someone else told me to use them.	3.9%	3.6%	4.2%
Other reasons.	5.3%	5.2%	4.2%

Hence, we do not conclude that the usage rate has decreased during a decade. However, we can conclude that the rate of visualization tool usage has certainly not increased even though the visualization tools have changed and the technologies developed. This result confirms the "anecdotal evidence" of Ross and Grindner [7].

4.2 Ways of Visualization Usage

The students' engagement with the visualization tool has been strongly emphasized by visualization tool developers and researchers ever since Hundhausen et al. [1] reported that it is most important *how* the visualizations are used. Naps et al. [4] instruct that instead of just viewing the visualization the student should respond to it or interact with the visualization in some other way.

As for our results, the survey shows that visualization tools are most often used by the teacher, not the students. Is this contradictory to the instructions on student engagement with the visualization? Or is it just one way of engaging the student with the visualization that the teacher presents the visualization to the students? Either way, as the teacher is the most common user of the visualization tools it should also be taken into account in the tool design, development of features, and instructions on how to use them. Currently, the instructions in this field strongly emphasize the student engagement but in light of this result maybe also teachers' use of visualization tools should be regarded.

Naps et al. [5] acknowledge teachers as being in central user role of visualization tools. They look at teachers' use of visualizations from a practical point of view and see that the key impediment to the adoption of visualizations by teachers is the time required to learn, install, and develop visualizations and then integrate them into a course. The results of our survey suggest that the teacher should also be regarded as a user of the visualization tool that explicates the content of the visualization to the learners.

4.3 Usage in Different Kinds of Courses

The comparison between the course categories shows that visualizations are still most often used in introductory programming courses and courses on algorithms and datastructures.

There is a statistically significant difference between teachers working with adults and teachers working with teenagers: the teachers of teenagers use visualization tools more often than the teachers of adults. The development of abstract thinking skills starts at the teenage [6], which may explain why teachers working with teenagers find it more important to support their students with visualization tools. Visualizations make the program behaviour visible and more concrete, and therefore aid in abstract thinking.

The reported statistical difference also affects the results of the other questions: the group that used visualizations more often naturally resulted in higher percentage values also on the questions regarding how they used visualizations and why they used them and lower on why visualizations were not used.

4.4 Reasons to Use or Not to Use Visualizations

Earlier surveys clarifying the reasons why teachers did not use visualizations have found practical problems the main reasons that hinder the usage [4, Section 2.2]. For example, lack of time was related to all of their top three reasons. In our results, the main restraint was that the teachers prefer to create their own visualizations instead of using visualization tools.

Knobelsdorf et al. [3] suggest activity theory as a framework for understanding the use of visualization tools in learning programming. We introduce the approach briefly, and then use it for discussing the results of the survey. Knobelsdorf et al. [3] explain how the visualization acts as a symbolic tool that mediates the teachers' and students' activi-

Table 5: Distribution of answers to the reasons agains using visualizations in programming courses.

Why were no SV tools used in the programming course(s)?	All	Teaching adults	Teaching teenagers	Basic prog.	Advanced prog.	Data str. & alg.
I do not believe that visualizations are educationally effective in this level of programming courses.	8.8%	10.1%	5.1%	6.8%	12.7%	6.7%
I prefer to create my own visualizations, e.g. by drawing pictures on the black board.	21.4%	25.8%	9.3%	18.2%	26.2%	19.2%
There are no visualization tools available for this topic.	6.3%	7.9%	1.7%	3.6%	7.1%	3.8%
I don't think the visualization tools available are educationally effective.	4.5%	4.1%	5.9%	5.7%	4.8%	3.8%
The available visualization tools illustrate the programming concepts in a way that I don't like.	2.4%	2.5%	2.5%	0.5%	2.4%	1.9%
It requires too much work to search for good visualization tools.	8.1%	9.3%	5.1%	8.9%	8.7%	8.7%
It requires too much work to learn to use new tools.	5.9%	6.8%	2.5%	5.2%	4.0%	5.8%
It requires too much work to find good visualization examples.	5.5%	6.3%	3.4%	6.8%	5.6%	2.9%
It requires too much work to develop new visualization examples.	4.3%	5.2%	1.7%	4.2%	5.6%	2.9%
It requires too much work to adapt the visualizations to my teaching approach or course content.	4.1%	4.9%	1.7%	3.6%	6.3%	2.9%
There are technical limitations in the class rooms where I teach, e.g. no data projectors available.	2.9%	3.0%	2.5%	3.1%	2.4%	1.9%
I'm lacking necessary technical skills.	1.2%	0.5%	3.4%	1.0%	0.8%	1.9%
Other reasons.	15.7%	18.6%	6.8%	14.1%	15.1%	10.6%

Table 6: Comparison of our resuts with the result of an earlier survey.

Our results		Result of Naps et al.		
Were SV tools used in the programming course(s) either by teachers, students or both?		Were dynamic visualizations used	in class	outside class
Not at all	45.6%	Never	13%	23%
Occasionally	32.8%	A few times per term	54%	53%
Regularly	21.6%	Almost every day/Once per week	33%	24%

ties. The student who is learning programming goes through an internalization process of programming concepts. This process is mediated by all the tools that are used, for example by the programming language but also the visualization tool. The visualization tool represents an externalization of the tool developer's understanding of programming concepts. The teacher who uses the tool has to acknowledge these externalized concepts and adopt the tool as a supportive educational tool. However, the teacher also has his/her individual internalization of programming concepts, which may be different. Therefore, the use of the visualization tool may complicate the activities by bringing in the tool developer's internal representations in the form of the visualization tool.

In the results of our survey, practical problems were indicated by a notably smaller group of respondents than the preference of creating one's own visualizations. We see that this lends support to the theoretical explanation given by Knobelsdorf et al. [3] that it is difficult for the teacher to acknowledge the tool developers externalized concepts. Instead, the teacher prefers to present his/her personal internalization of the programming concepts in a personal format. The practical problems do of course also exist, but we should not underestimate the meaning of the psychological hinders related to the use of visualizations.

At the same time, the main reason why the teachers used visualizations was that they found visualizations educationally effective. However, many of the other reasons for using visualization tools were related to features of tools that have nothing to do with the actual visualization. This reflects that visualization tools are getting integrated with course management systems and becoming parts of other kinds of large software systems.

The educational effectiveness of software visualizations has been debated among teachers of programming. As the educational effectiveness of the visualization tools was the main reason for using them, apparently there is a group of teachers who believe in software visualization. These teachers have found their own visualization tools whose externalized representations they can adopt in their teaching.

5. CONCLUSIONS

We presented the results of a survey that reached a reasonably large group of programming teachers and found that the "anecdotal evidence" holds: visualization tools are not really used commonly in programming courses. Approximately 20% of programming courses use visualization tools regularly and slightly less than half do not use them at all. Visualizations are more often used by teachers working with younger students. The topics where visualizations are used most often are basic programming and datastructures and algorithms.

Especially noteworthy was that most often the visualization tool is used by the teacher, not the students. This is contradictory with the predominant understanding that software visualization tools are educationally effective mainly when the student is engaged with the use of the tool. As the teachers use the visualization tools more often than the students, we recommend that this way of using them should also be taken into account in the tool and material design and instructions on their usage.

The main reason why teachers use visualization tools is that they found the tools educationally effective. Earlier surveys regarding the hinders of visualization usage in programming courses listed the practical reasons and difficulties as the main problems. Our survey revealed that the primary reason for not using visualization tools is that the teacher prefers to create his/her own visualizations in some other way, e.g. using the blackboard. This lends support to Knobelsdorf et al. [3] approach of using activity theory as a theoretical framework to explain how and why visualizations are used and why they are not used in learning programming. Knobelsdorf et al. [3] theoretically explain the psychological hinders of adopting the visualization tools as supportive educational tools in teaching programming. Such psychological hinders should not be underestimated in tool development.

This study revealed new interesting topics to be discussed by visualization tool developers and researchers. Despite the large body of knowledge on software visualization tools, there are still new points to be taken into account in this field.

6. ACKNOWLEDGEMENTS

We are very grateful to all the respondents of the survey and want to thank everyone who helped us to spread the survey link by posting it on their national mailing lists.

7. REFERENCES

[1] C. D. Hundhausen, S. A. Douglas, and J. T. Stasko. A Meta-Study of Algorithm Visualization Effectiveness. *Journal of Visual Languages & Computing*, 13(3):259–290, 2002.

[2] E. Isohanni. *Visualizations in Learning Programming — Building a Theory of Student Engagement.* Doctoral Thesis. Tampere Univerisity of Technology Doctoral Dissertations, Tampere University of Technology, Finland, 2013.

[3] M. Knobelsdorf, E. Isohanni, and J. Tenenberg. The reasons might be different: Why students and teachers do not use visualization tools. In *Proceedings of the 12th Koli Calling International Conference on Computing Education Research*, Koli Calling '12, pages 1–10, New York, NY, USA, 2012. ACM.

[4] T. Naps, G. Rössling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger, and J. Velazquez-Iturbide. Exploring the Role of Visualization and Engagement in Computer Science Education. *SIGCSE Bulletin*, 35(2):131–152, June 2003.

[5] T. L. Naps, G. Rössling, J. Anderson, S. Cooper, W. Dann, R. Fleischer, B. Koldehofe, A. Korhonen, M. Kuittinen, C. Leska, L. Malmi, M. McNally, J. Rantakokko, and R. J. Ross. Evaluating the Educatiocal Impact of Visualization. *SIGCSE Bulletin*, 35:124–136, June 2003.

[6] J. Piaget and B. Inhelder. *The Growth of Logical Thinking from Childhood to Adolescence.* Basic Books, New York, 1958.

[7] R. Ross and M. Grinder. Hypertextbooks: Animated, active learning, comprehensive teaching and learning resources for the web1. In S. Diehl, editor, *Software Visualization*, volume 2269 of *Lecture Notes in Computer Science*, pages 269–283. Springer Berlin Heidelberg, 2002.

[8] C. A. Shaffer, M. Akbar, A. J. D. Alon, M. Stewart, and S. H. Edwards. Getting Algorithm Visualizations into the Classroom. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, SIGCSE '11, pages 129–134, New York, NY, USA, 2011. ACM.

[9] C. A. Shaffer, M. Cooper, and S. H. Edwards. Algorithm Visualization: a Report on the State of the Field. In *SIGCSE '07: Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*, pages 150–154, New York, NY, USA, 2007. ACM.

[10] C. A. Shaffer, M. L. Cooper, A. J. D. Alon, M. Akbar, M. Stewart, S. Ponce, and S. H. Edwards. Algorithm Visualization: The State of the Field. *Transactions on Computing Education*, 10(3):9:1–9:22, Aug. 2010.

[11] J. Sorva. *Visual Program Simulation in Introductory Programming Education.* Doctoral Thesis. Aalto University Publication Series Doctoral Dissertations 61/2012, Aalto University, Finland, 2012.

[12] J. Stasko, J. Domingue, M. H. Brown, and B. A. Price. *Software Visualization: Programming as a Multimedia Experience.* MIT Press, 1998.

[13] J. T. Stasko and C. D. Hundhausen. Algorithm Visualization. In S. Fincher and M. Petre, editors, *Computer Science Education Research*, pages 199–228. Taylor and Francis, The Netherlands, Lisse, 2004.