

INTEGRATING THE USE OF VISUALIZATIONS TO TEACHING PROGRAMMING

Essi Lahtinen

Institute of Software Systems

Tampere University of Technology

P.O.Box 553, FIN-33101 Tampere Finland

email: essi.lahtinen@tut.fi

ABSTRACT

Both students and teachers have problems when trying to reach the learning goals of the introductory programming courses. One attempt to aid learning and teaching was the development of visualization materials about programming concepts. Even though there are visualization tools and examples available for free use, the teachers have problems in providing the visualizations for the students in a usable way and the students have problems in using them effectively in learning. Improving the way the visualizations are put available for the students would increase their usage and the benefits gained by them. Also the use situations of the visualizations should be considered to find the most beneficial ones.

The aim of this article is to facilitate the usage of visualizations by providing tips on how to integrate the visualization materials to the teaching. The article consists of a model on how to integrate the visualizations to the course materials and ideas on how to use the visualizations in different learning and teaching situations. The integration of the visualizations in the course materials has been tested on a programming course and the experiment shows that it has increased the use of visualizations in learning.

1. INTRODUCTION

Programming is an abstract subject that is difficult to learn for most of novices. The biggest learning problems are that the students have to handle concepts to which they do not have a concrete model in their everyday life [1], that they tend to approach programs line by line [2, 3], and that they are not able to handle the larger wholes of the programs [4].

In addition to the learning problems there are usually also problems in teaching programming. Due to the lack of resources causes the teachers can not give personal instruction to all the students and the weaker students suffer. One way to aid the students that would need revision of the taught subject is to provide visualizations for them. Numerous visualization tools e.g. VIP [5] and Jeliot 3 [6] are already available for the use in introductory programming. These visualizations are capable of providing an easier perspective to approach programming if they are used effectively.

Visualizations can ease these problems. First of all good visualizations can provide a different kind of a perspective for the student and make the student understand the behaviour of the program more comprehensive. Secondly visualizations can be available for the students always when they need the support but the teacher is not available. Thus, using visualizations as voluntary extra material for the students that find the course challenging is an idea worth trying.

Even if the use of visualizations has increased in introductory programming courses, the visualizations are still often not integrated in the teaching, the course content and the learning situations. If there are just a few visualizations available every now and then during the course, the students do not develop a habit of using them regularly in learning and they are easily forgotten. On the other hand even if the teacher had prepared visualizations continuously of all the subjects included in the course, it can happen, that the students do not find them beneficial in later parts of the course, because the visualizations are biased only on the lower levels of cognitive development [7]. The teacher needs to plan the usage of visualizations carefully and actively guide the students towards using the visualizations to gain the maximum benefit out of them.

This article begins by introducing the organization of programming courses and the uses of visualizations there in Section 2. Section 3 introduces a model on how to integrate the use of visualizations to the course materials and Section 4 deals with the integration of the visualizations to the learning and teaching situation. The tips and ideas are concluded in Section 5.

2. VISUALIZATIONS AND THE COURSE SETTINGS ON INTRODUCTORY PROGRAMMING COURSES

When programming, the student needs to use a computer and it is a part of his learning environment. Thus, in addition to the use of books and other more traditional teaching materials, e.g. copies of the lecture notes, it is very natural to put the materials of a programming course on a web page. The web pages of programming courses often contain links to learning materials and assignment descriptions for exercise sessions and programming assignments. This makes the course web site an important and frequently visited source of information. Thus, the use of the visualization tool can be made easy on a programming course.

On introductory level of programming, visualizations are typically used as illustrative visualizations for presenting new concepts for the students [7]. However, this is only one use of visualizations and the scope can be widened to other parts of the course content too.

Programming courses typically consist of lectures, exercise sessions (either in normal class rooms or computer rooms) and programming assignments. Besides these there can be various kinds of extra tasks e.g. homework essays. The teacher has to choose in which teaching situations the use of visualizations is most beneficial for the students. For instance, showing the visualization in front of a class room is usually not beneficial, because the amount of student participation is significant for the engagement in learning [8]. Thus integrating the visualizations to the learning and teaching situations is important.

Visualizations can be divided into five different levels according to the level of the cognitive development of Bloom's taxonomy [9] they support [7]. These different kinds of visualization categories suit for different parts of the course and for supporting different kinds of activities. Table 1 shows the division of the visualizations and ideas on the situations in which they could be used. For example an illustrative visualization would suit also for showing on a lecture since it just offers knowledge on how the concept works but a problem-solving visualization is better for situation where the student uses it independently, since it requires participation from the user.

Table 1. Usage of visualizations from different categories of Bloom's taxonomy.

| Levels of Bloom's taxonomy | Visualization category | Usage |
|-------------------------------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 knowledge & 2 comprehension | Illustrative visualizations | Lecture materials, revising after lectures, revising before the exercise sessions |
| 3 application | Utilizing visualizations | Homework assignments, assignments in exercise sessions, small programming assignments |
| 4 analysis | Problem-solving visualizations | Homework assignments, assignments in exercise sessions, revising the subject |
| 5 synthesis | Productive visualizations | Homework assignments, assignments in exercise sessions, assignment descriptions for programming assignment, supporting the programming assignments |
| 6 evaluation | Discerning visualizations | Lecture materials, homework assignments, assignments in exercise sessions, basis for essays, revising the subject |

3. INTEGRATING THE VISUALIZATIONS TO THE COURSE MATERIALS

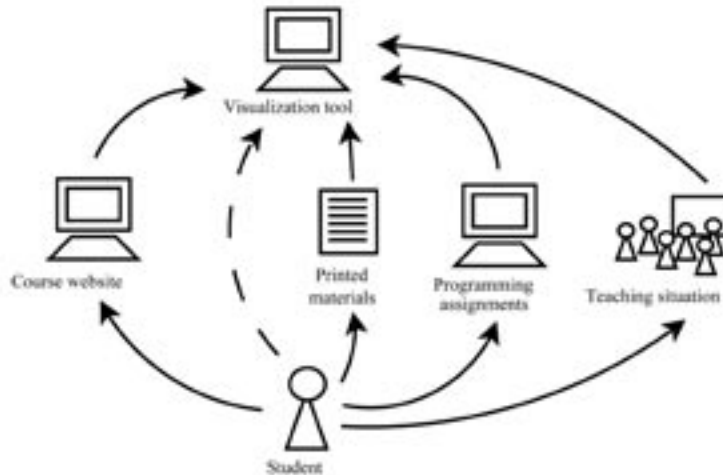


Fig. 1. Visualization-directed model of the organization of course materials.

One flaw in the use of the visualizations is that the students do not find them or do not recognize to use them at the right moment. This occurs when the visualization tool is in a separate learning environment, e.g. on a web page that is not connected to the course material. Other little details such as the need of typing a pass word for the web page may also hinder the use of the visualizations.

To increase the use of the visualization tool it should be integrated to the actual course materials. We suggest that the teacher includes links and references to the examples in the visualization tool to all the content of the course material. This does not only mean the printed materials, but we can also take advantage on the multiple purposes of the course web pages of programming courses when promoting the visualizations.

There can be students that are simply fond of the idea of using visualizations and use the tool as it is. Most of the students, however, need to be guided to use it. Figure 1 illustrates how the students who are not that spontaneous can be reminded of the visualization tool. The idea is to exploit the materials that they already know how to use or that they need to use. The printed course materials contain references to the visualization tool. The course web page contains links to the visualization tool. The programming assignment descriptions and the programming environment are attached a possibility to gain support by using visualization. The teaching situations also use visualizations when possible. This way also the students who do not plan their learning situations and the use of the materials will benefit of the visualization tool in right situations.

The base of the course material is normally a book or copies of the lecture slides. Students search for information about all the subjects there. It is very important to integrate visualizations into these materials.

There are different ways to do this:

- an example program from the material can be presented as an illustrative visualization as it is
- the theory sections of the material can be supplemented with extra visualization from the same area (more example programs, only in a different format)
- the assignments in the material can be provided alternatively as utilizing, problem-solving, productive or discerning visualization

In all the cases the visualizations need to be marked with identification numbers that also have to be clearly mentioned in the printed material so that the student seeking for extra advise or material can easily find them.

We have measured the use of a visualization tool -- VIP [5] -- on the introductory programming courses in TUT. The visualization tool automatically collects a log file about all the actions done in the visualization tool. This way we can follow how the visualizations have been used. On the first year the visualization tool was provided on the course web page and introduced on the lectures but it was not integrated to the materials properly. On the second year all the materials included links to it. The result was not only that the students used the visualization tool more when it was integrated to the course materials. Also the use of the visualizations concentrated better on the subjects that were handled at the time. I.e. the students used the visualization tool in their studies.

Figure 2 shows how the visualizations on a specific subject are downloaded on the website. The course started on week 42. The use of visualization examples on weeks 42 and 43 is ca. the amount they are normally used during the semester all the time. On week 44 there was the first lecture on loop structures and the number of downloads of the loop structure examples has increased because of the references to them in the course materials. On week 45 there was another lecture on loop structures and in addition the weekly exercise session handled loop structures. This doubled the use of the loop structure visualizations. Week 46 was an exam week so the use is slightly smaller but there were students using the visualization tool most likely because on the week 47 there was a deadline for a programming assignment that handles loop structures. The deadline week was almost as busy in the use of visualizations as the exercise session week. After the deadline of the assignment there are again very few downloads of the loop structure visualizations but on week 50 there are again students revising the use of the visualizations because there is a deadline of an assignment about the new subject: arrays.

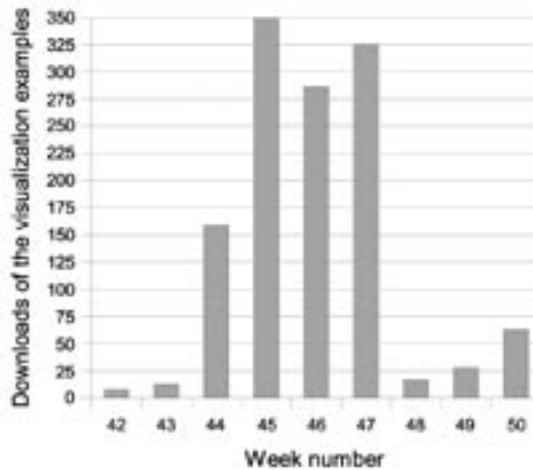


Fig. 2. Weekly usage of the visualization examples about loop structures on a course where the visualizations were integrated to the course materials.

So it can be clearly observed from Figure 2 that the students' usage of the visualization examples follows their course work when the material contains links and references to the visualization examples.

4. INTEGRATING THE VISUALIZATIONS TO THE LEARNING SITUATIONS

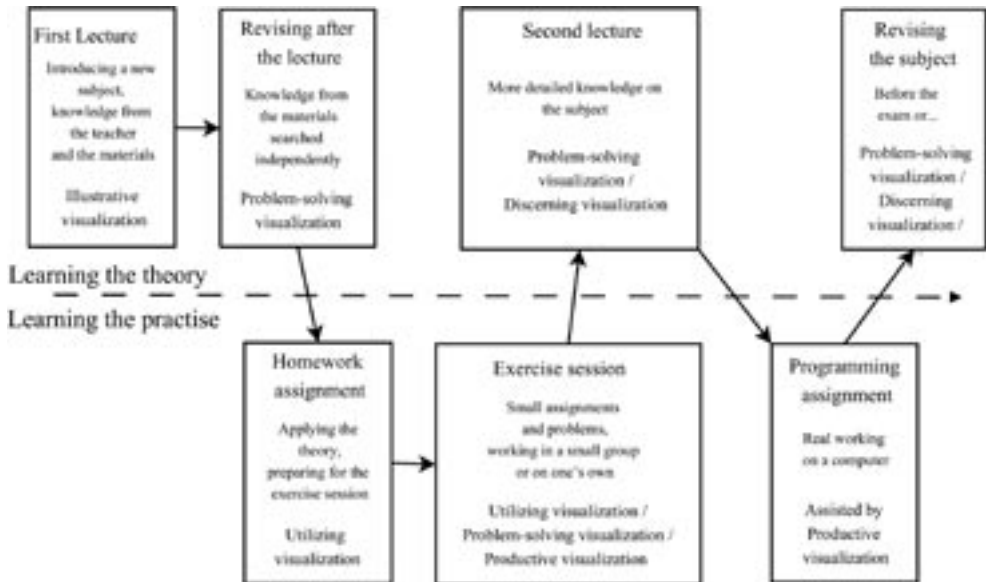



Fig. 3. Visualizations to support different learning and teaching situations

Figure 3 introduces a plan on how the visualizations can be used when the course proceeds. The content of the Figure shows how one specific subject, e.g. loop structures, is taught to the students. The course repeats these phases for all the subject. For instance an introductory programming course can start by introducing variables, then proceed to selection structures and then to loop structures. The phases in Figure 3 are first done for the variables then again for the selection structures and again for loop structures etc.

The figure is divided into learning the theory and learning the practise. These parts do not need to be divided in teaching, so both can happen in same teaching sessions. However, most of the programming courses have lectures and exercise sessions separately so the figure is built this way.

A new subject is typically first introduced on a lecture. The teacher gives an overall picture of the subject to the students. Sometimes showing a visualization can help the teacher in his explanation so the visualizations can also be used on lectures even if it is not the most effective use situation for them. Since the student participation to controlling the visualization is small on a lecture, most likely the used type is an illustrative visualization.

After the lecture the student can revise the content of the lecture by reading the book or the copies of the lecture slides. With the isualization identification numbers included in the materials the revisioning includes also use of the visualization tool. If the course web site contains a schedule on the lecture contents the teacher can add links to the visualization examples on the topic there and easen the students work for searching the visualizations by the reference numbers. At this time the visualizations provided do not need to be only illustrative visualizations. Since the participation of the student is desired, a problem-solving visualization would actually suit much better. If the teacher wants to provide extra help for the students that missed a lecture he can also add visualizations about examples that are presented on the lecture from outside the lecture material in the visualization environment.



It has been researched that the use of visualizations in preparing for the exercise sessions improves the students preparation for the exercise session \cite{ahoniemi}. Thus it is a good idea to provide a visualization task for the students as a homework assignment to be done prior to the exercise session. We recommend a utilizing visualization for this since they are typically not too taxing but do engage the student to work himself, which increases the benefits of the visualization.

In the exercise session there are numerous ways of using visualizations and many different types can be used since the teaching situation in a smaller teaching group is usually easier to organize than in a lecture. We suggest that the used type of visualization here would be something else than a utilizing visualization because the other types engage the student in the learning process better. At least some of the assignments every week could be available in the visualization tool in addition to other ways of handling the assignment. This way the weaker students can choose to do the assignment in the visualization tool and get started with the new subject. The students that do not have that many difficulties can choose the alternative to do the assignments in a more traditional way without the visualization tool.

The use of productive visualizations (visualizing the students solution) in the programming assignments the students need to return in the beginning of the course can also be helpful for the weakest students. This way the students use the visualization tool to replace their normal program development environment. It is called productive visualization. However, working in a visualization tool can be frustrating for the students that can already program well so it should not be obligatory for everyone.

Sometimes it is a good idea, that using a visualization for at least in one task in the beginning of the course is obligatory, so all the students will have at least one try with them. This way the students who recognize them useful will be able to continue with using them. All of the students do not want to participate in new ways of learning so the use of visualizations should not be obligatory all the time.

5. CONCLUSIONS

This paper provided a set of instructions and ideas for the teachers of introductory programming courses on how to use visualizations. The instructions included a visualization directed model on integrating the visualizations to the course materials and ideas on how to use different types of visualizations in different learning and teaching situations.

To achieve a wide usage of the visualizations it is important that the teacher links them to all the teaching materials of the course. This way the student does not need to put effort on finding the visualizations and the usage will be easier and more frequent for the students. The teacher also needs to take care that the different categories of visualizations are used versatily and that the visualizations are used in different kinds of learning situations. This way the student will be supported also on higher cognitive levels than just providing knowledge of programming concepts.

These instructions should increase the benefits gained from providing the visualizations for the students and easen the work done both by the students and the teachers.

6. REFERENCES

- [1] A. Robins, J. Rountree, and N. Rountree, "Learning and teaching programming: A review and discussion," *Computer Science Education*, vol. 13, no. 2, pp. 137–172, 2003.
- [2] E. Soloway and J. Spohrer, *Studying the Novice Programmer*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1989.
- [3] L. E. Winslow, "Programming pedagogy – a psychological overview," *SIGCSE Bulletin*, vol. 28, no. 3, September 1996.
- [4] E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, "A study of the difficulties of novice programmers," *ITiCSE 2005, Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, pp. 14–18, June 2005.
- [5] A. T. Virtanen, E. Lahtinen, and H.-M. Järvinen, "VIP, a visual interpreter for learning introductory programming with C++," *Proceedings of the Fifth Finnish/Baltic Sea Conference on Computer Science Education*, November 2005.
- [6] A. Moreno, N. Myller, E. Sutinen, and M. Ben-Ari, "Visualizing programs with Jeliot 3," *Proceedings of the International Working Conference on Advanced Visual Interfaces AVI 2004*, May 2004.
- [7] E. Lahtinen and T. Ahoniemi, "Visualizations to support different levels of cognitive development," *Proceedings of the Fifth Finnish/Baltic Sea Conference on Computer Science Education*, November 2005.
- [8] T. Naps, G. Rössling, V. Almström, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger, and J. Velázquez-Iturbide, "Exploring the role of visualization and engagement in computer science education," *SIGCSE Bulletin*, vol. 35, no. 2, pp. 131–152, June 2003.
- [9] B. S. Bloom, M. D. Engelhart, E. J. Furst, W. H. Hill, and D. Krahwohl, *Taxonomy of Educational Objectives, Hand-book I: Cognitive Domain*. New York: David McKay Company, Inc., 1956.