El problema de la escalabilidad en visualizaciones complejas de la información y del *software*.  
TIN2004-07568

J. Ángel Velázquez Iturbide*  
Departamento de Lenguajes y Sistemas Informáticos  
Universidad Rey Juan Carlos

Abstract

A visualization is an abstract representation of an object that improves our understanding of it. The term information visualization is used for the emerging discipline that studies the design and implementation of computer-generated visualizations. One of the main problems of visualization, named the scalability problem, is given when the number of elements of compound information is so large that the visualization is unable to represent it adequately. The main aim of the project consists in developing techniques to effectively visualize information of large size. The techniques will be developed for and applied to three domains: functional programs, object-oriented programs and students’ activity in e-learning platforms. Each domain deserves a different treatment, since the scalability problem has received different degrees of attention and the effort necessary to construct visualizations also greatly varies.

**Keywords:** information visualization, software visualization, scaling problem, effortlessness, functional programming, object-oriented programming, e-learning platforms.

1 Aims of the Project

A visualization is an abstract representation of an object that improves our understanding of it. The term information visualization is used for the emerging discipline that studies the design and implementation of computer-generated visualizations. One of the main problems of visualization, named the scalability problem, is given when the number of elements of compound information is so large that the visualization is unable to represent it adequately.

The main aim of the project consists in developing techniques to effectively visualize large-scale information. We seek domain-independent techniques and techniques for three specific domains: functional programming, object-oriented (OO) programming, and e-learning platforms.

The proposal is grounded on experience of the applicant team, mainly in program visualization and educational applications. Previous developments that were substantially used for the project are the functional programming environment WinHIPE and a web-based e-learning platform.

The project proposes to enhance visualization techniques mainly by means of three approaches. Firstly, the combined use of different techniques will have a synergic effect. We propose to use, at least, the multiple views, navigation, and simplification techniques. Secondly,
porting a technique from a given domain to another one will require enhancing it. Thirdly, new visualization techniques will probably be necessary. A key element in the three approaches is effectiveness evaluation by means of controlled experiments.

Each domain has specific features that must be taken into account. For instance, software is often visualized dynamically (as an animation) in educational contexts. Another important difference lies in the role of the user. Information visualizations are completely generated by computer programs, so the user only interacts with them. However, software animation systems typically provide the infrastructure to the teacher, who must build the animation with great effort.

The initial scheduling of the project is shown in Table 1. For the sake of brevity, we omit tasks concerning mundane activities (task A) or bibliographic study (subtasks *.1). Tasks correspond to the domains of study: functional programming (task A), OO programming (task B) and e-learning platforms (task C). Roughly, all of the tasks consist in a sequence of design, implementation and tuning of a visualization model. Task B contains an intermediate subtask (B.3) to implement a language processor. Task D contains twice such a sequence: first for multiple, partial visualizations of students' activity, and then for a comprehensive visualization model. There is no specific task to develop general visualization techniques, but they will be a side effect of research in the 3 domains.

<table>
<thead>
<tr>
<th>Task</th>
<th>First year</th>
<th>Second year</th>
<th>Third year</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D.7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Chronogram of tasks

2 Technical Results

2.1 Key Decisions

Task B was altered so that a new functional programming processor was not developed: we considered more productive to make use of a processor already available by the group. Task C also was modified so that a single approach was not addressed, but several exploratory works were performed. Task D is being performed pretty similar to the initial schedule. Despite these deviations from the initial schedule, contributions obtained so far justify these decisions.

With respect to visualization of programs, we focused in educational contexts. As explained above, a major source of difficulty in education is the effort required to generate animations. This workload is posed on educators. Consequently, our research efforts on software visualization are performed within effortless frameworks, typically based on program visualization.
2.2 Visualization of Functional Programs

Objectives Fulfilled to Date
In summary, we extended the WinHIPE functional programming environment to address in depth the effort and the scaling issues.

We refined the effortless approach of WinHIPE to build discrete animations of the evaluation of functional expressions [35]. Further support was given to customize and maintain web animations [26][27]. Extensions to integrate tests also were considered [24]. The approach was assessed with students of a programming languages course. Statistical evidence was obtained about the fact that students who construct an animation perform better than those just watching it [28].

The scale problem was addressed in three aspects: size of an expression, number of expressions, and number of animations.

An intermediate expression can be too large to fit a window, therefore being difficult to understand it. We designed a technique to filter the parts of an expression that are less interesting. It is a logical fisheye view [32] based on five design requirements.

The selection of the visualizations that form an animation is difficult when their number is large, even if they are displayed as miniatures, because they do not fit a window. We designed a focus+context technique to deal with miniatures, called r-zoom [25]. The technique satisfies specific requirements of program visualizations and provides enhanced user interaction. It was empirically evaluated and compared to an overview+detail interface [30]. Experienced users of r-zoom completed tasks faster and with fewer errors than users of the overview+detail interface. Furthermore, users felt more comfortable with r-zoom.

When an educator develops animations, she will probably want to reuse them for different courses. WinHIPE allows constructing and maintaining hierarchical collections of web animations [26][27]. The user may handle the hierarchy, the animations and their look. A collection is visualized as a tree of folders with expansion/contraction facilities.

Current and Future Work
We are working on four issues. Firstly, we made a long-term evaluation of the educational effectiveness of WinHIPE during two months. We still have to statistically analyze collected data. Secondly, we are developing a pedagogical guide for educators to design animations. Thirdly, a general API for r-zoom, amenable to be used by any Java container and visualizer, is under development. Finally, we plan to evaluate the facilities for collections of web animations.

2.3 Visualization of Object-Oriented and Imperative Programs

Objectives Fulfilled to Date
We conducted different exploratory works on visualization of object-oriented and imperative programs. The goal was to explore several domains, to identify the role of visualization in them, and to study the importance of the scaling problem. The domains are: OO debugging, tutors and assistants on OO programming, symbol tables, and algorithm design techniques.

We identified the limitations of line-oriented debuggers for the practitioner programmer and developed a structural debugger [5]. It solved those problems with structural actions interactively invoked with a graphical interface. It is based on JavaMod, an API developed by us.

We addressed the design of a series of tutors for Java OO concepts. They are designed within the framework provided by Bloom’s taxonomy [8]. Three tutors, aimed at the 3 lowest levels of the taxonomy, were developed. Firstly, we developed a tutor on inheritance [6][7]. It contained components for theory, Flash animations and adaptive tests. The tutor was evaluated, yielding
enhanced learning at the comprehension level and students’ positive attitude. Other tutors [9] were: GeCom, that automatically generates comments in programs, and CreOO, that randomly generates and corrects problems on object creation.

We also developed TextOO, an assistant aimed at the application and upper levels of Bloom’s taxonomy [13][14]. It assists in developing class, object and interaction UML diagrams and linking them to parts of a natural language specification. The assistant also corrects students’ diagrams.

Visualization of other kinds of imperative programs was also addressed. SOTA is a tool aimed to graphically demonstrate the behaviour of symbol tables [2][3]. Its educational effectiveness was empirically evaluated [4][29]. Students using SOTA performed the same as students not using it, but in less time. They also had a positive attitude towards using SOTA for self-study.

A final line of research is visualization of algorithms based on design techniques. We performed a comprehensive study on visualizations for four algorithm design techniques [1]. In order to implement their visualization, we developed a generic framework.

Current and Future Work
The work described so far continues in several lines. Two works are in progress to explore the extension of our tutors to other programming domains: programming syntax and greedy algorithms. The TextOO assistant also is being enhanced to better assess UML diagrams. Two works on SOTA are in progress: a new evaluation of its educational effectiveness, and a new version to deal with any source language; most importantly, this new version will include scaling facilities. With respect to the visualization of algorithm design techniques, we are currently extending the generic framework to visualize recursion using multiple views.

2.4 Visualization of Students’ Activity in e-Learning Platforms

Objectives Fulfilled to Date
Our first contribution to visualization in e-learning platforms was focused on timetable and exam schedule [33]. Tabular representations were designed, where collisions were detected graphically.

Next efforts were devoted to visualize students’ activity. We followed two approaches. For the first one, we used an e-learning platform developed by our team ([http://trancos.escet.urjc.es/urjc](http://trancos.escet.urjc.es/urjc)). We identified issues that educators are typically interested in about their students’ activity, we studied 10 tools for analyzing web sites and we concluded that they fail in tracking individual behaviour [10]. Consequently, we developed two extensions for our platform, respectively based on a database of students’ actions and on cookies. A comparison of functionality provided by commercial analysis tools, our prototype and surveys was conducted [11], as well as a comparison with other e-learning platforms [12].

A different approach was concurrently used to analyze students’ activity of an on-line course offered within the ADA-Madrid project ([http://www.upm.es/adamadrid](http://www.upm.es/adamadrid)), where a different e-learning platform was used. Data collected by the platform were used for exploratory data analysis and data mining [21][22]. Surprising conclusions were obtained about university registration procedures, students’ skills in computer usage, interaction patterns, descents in performance, students' evaluations, or self-confidence.

Current and Future Work
The two platform extensions are collecting access data from 6 class groups. In February, we will be able to analyze and compare data obtained by both tools. A questionnaire will also be submitted to instructors. Based on our experiences, we will design a more comprehensive visualization model of students’ interaction. However, evaluation will be done after the end of the project.
2.5 Other Contributions

We delivered several states of the art [16][18][19] and WG reports [17][20] on visualization. Other contributions were produced as a side effect of research on visualization. Animations of functional programs were used in a freshmen course on programming languages [31]. We restated algorithm design by transformations [15]. An on-line course on visualization was offered [21][23].

3 Indicators of Success

3.1 Personnel in Training

Three PhD Thesis are being conducted. The first one is finished and will be presented in 2007:

- Jaime Urquiza. “Generación semiautomática de animaciones de programas funcionales con fines educativos”.
- Isidoro Hernán. “Taxonomía de Bloom como marco de diseño de tutores programación”.
- Raquel Hijón. “Análisis visual de actividad de los alumnos en plataformas de e-learning”.

The project has also been a framework for research grants and works:

- Micael Gallego, URJC doctoral grant.
- Antonio Pérez, project grant.
- Research projects of graduate students (Luis Fernández, Raquel Hijón, David Redondo).
- Capstone projects of undergraduate students (11 projects).

3.2 Publications

The project has produced so far 35 publications. They are here summarized by categories. The complete list of these publications can be found in the references section.

- Edited books: 1
- Book chapters: 4
- Articles in international journals or magazines: 4
- Papers in international conferences: 17
- Papers in international workshops: 1
- Posters and sw demos in international conferences: 3
- Papers in national conferences: 2
- Papers in national workshops: 2
- Technical reports: 1

3.3 System Implementations

We have delivered several applications, already mentioned above:

- Extensions of the WinHIPE IDE to generate and manage animations.
- The JavaMod API to model Java programs and a Java structural debugger.
- Three OOP tutors (anonymous tutor on inheritance, GeCom and CreOO).
• The TextOO assistant to develop UML designs based on natural language specifications.
• The SOTA system to visualize symbol tables.
• A framework to visualize algorithm design techniques and a prototype for recursion.
• A web application to schedule timetables and exams.
• Two extensions of our e-learning platform (based on databases and cookies).

3.4 Collaboration with National and Foreign Groups

Our research team has collaborated with several groups. Firstly, we participated in working groups of the ACM Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE). In particular, Jaime Urquiza participated in a group at ITiCSE 2005, Lisbon, and Jaime Urquiza and Ángel Velázquez participated at ITiCSE 2006, Bologna. Jaime Urquiza participated at the 2006 GIDagstuhl-Seminar “Human-Centered Visualization Environments” held in Wadern, Germany.

Secondly, visits were made to other universities. Jaime Urquiza visited the EdTech group at Joensuu University, Finland, from July to August 2006 (2 months). Maximiliano Paredes visited the CHICO group at the Universidad de Castilla – La Mancha from June to September 2006 (3 months), as well as the Universidad Privada Antenor Orrego, Peru, from November to December 2006 (one month). Manuel Rubio visited the COSMAL group at University of California, San Diego, on September 2006, and has been accepted to perform a 6-month research visit in 2007.

3.5 Other Activities

We organized a seminar in 2006 [34], co-supported by the URJC. A second edition is being organized for 2007. Manuel Rubio and Ángel Velázquez received the Best Poster Award at the I Jornada de Innovación Pedagógica del Proyecto ADA-Madrid for their analysis of students’ activity [22][23].

4 References


[34] Á. Velázquez, M. Paredes (eds.). Tecnologías del Software. Dyckinson, in press.
El problema de la escalabilidad en visualizaciones complejas de la información y del software. TIN2004-07568. Article. Full-text available. Se describen los diferentes tipos de información que componen las animaciones de programas, su proceso de construcción y los diseños alternativos para su publicación en la web. Se ha evaluado su efectividad pedagógica así como el grado de satisfacción de los usuarios. Los resultados obtenidos son prometedores, ya que los usuarios opinan que el proceso de construcción de las animaciones es fácil de aprender y usar, también opinan que las animaciones basadas en la web son útiles como herramientas educativas.