Making the Comprehension of Software Architecture Attractive

Claudia Susie C. Rodrigues
System Eng. Comp. Science
COPPE/UFRJ
susie@cos.ufrj.br

Cláudia M. L. Werner
System Eng. Comp. Science
COPPE/UFRJ
werner@cos.ufrj.br

Abstract

Visualization stimulates the cognitive capacity of humans and facilitates the understanding of a subject. It performs a crucial role in teaching software architecture. As systems become more complex, new education proposals have been introduced in the classroom, especially those that make teaching more attractive to students. This paper presents the VisAr3D approach which was designed to provide a 3D visualization of UML models, where the user should intuitively understand architectural elements in this 3D environment. It includes exploration, interaction and simulation resources to establish a practical and pleasant learning activity, focusing in large scale systems.

1. Introduction

This research exploits the emerging technologies of 3D visualization, such as virtual reality and augmented reality, recognizing that visual communication can be a key factor in the process of teaching and learning of future software architects. Recently, interest in these technologies has increased both in entertainment and training applications (e.g., games and theater industry). By the increasing popularity of such technologies, that people not only currently complete complex tasks in such environment, but also enjoy the experience.

Software can be incredibly complex with a large demand for quality. Software Engineering (SE), as well as its teaching, becomes vital in this scenario. Based on these demands, new proposals for SE education have been introduced in the classroom, especially those that make teaching more attractive to students.

According to Feijs and Jong [1], 3D designs can be attractive and fun to walk through. Such visual appeal can be used for presentation purposes, explaining software design to potential customers. As Ware and Hui [2] state, there is an increasing evidence that it is possible to perceive and understand complex information systems if they are displayed as graphical objects in a three dimensional space.

Although the question of what are the benefits that a 3D representation offers over 2D still remains to be answered, some experiments have shown some optimistic results.

The main purpose of this work is to use 3D visualization technologies to help the understanding of UML models of a complex system. The tangible benefits of achieving this goal is to support the understanding of software architectures, by developing Visar3D (Software Architecture Visualization in 3D), which aims to provide the user the ability to manipulate and analyze a large amount of data from multiple perspectives through an attractive experience. This approach allows students to first participate and explore large scale systems and later be prepared to build their own diagrams. For a course to be successful, the students' level of interest, as well as their teacher's, is an important factor to be aimed.

978-1-4577-0348-5/11/$26.00 © 2011 IEEE CSEE&T 2011, Waikiki, Honolulu, HI, USA

The remainder of this paper is structured as follows: Section 2 presents some related work. Section 3 describes the VisAr3D approach. Section 4 addresses some final considerations, and, finally, references are presented in Section 5.

2. Related Work

Previous research has shown that there are practical advantages to use 3D for UML visualization. For example, Mcintosh *et al.* [3] focused on the "visual" aspect of UML as the means of communicating information to the viewer. The main difference with regards to the VisAr3D approach is how they explore the 3D environment to provide means for understanding complex software system architecture. Gil and Kent [4] describe a 3D graphical notation that combines several familiar 2D diagrams. In VisAr3D, the basic idea is to make 3D visualization as familiar as 2D and intuitive. As it happens in VisAr3D, Radfelder and Gogolla's work [5] applies static diagrams (their work also incorporates dynamic diagrams).

The GEF3D framework [6] builds on the Eclipse Graphical Editor Framework and as such enables existing Eclipse 2D UML editors to be extended into 3D. And Lange *et al.* [7] proposed a framework that supports task-oriented modeling and developed views and visualization techniques. The VisAr3D approach uses diagram views to visualize the system from different perspectives.

3. The VisAr3D Approach

Currently, VisAr3D provides a static view of a large scale system model, emphasizing the static structure of the system, although there is a plan to also represent behavior in the future.

3.1. Basic System Principles

Jacobson *et al.* describe [8] a set of general design principles for creating learning environments and tools, based on recent constructivist models of learning and on a consideration of the successes and challenges identified in recent education and complex systems projects. Those principles confirm the following eight VisAr3D Basic System Principles in several points:

(1) It must support the development and participation of students in complex projects. The designs found in the labor market are generally large systems, with requirements such as performance, reliability, low cost and high quality. Meanwhile, students leave college without dealing with complex problems. This approach should enhance the modeling of a system for large groups of students and support the understanding of the parties, to facilitate understanding of the whole. Students often learn theory best when it appears in concrete examples.

(2) It must reduce the gap between theory and practice. Allied to theory, supporting practice brings a greater dynamism to the classroom and results in a more meaningful participation of students.

(3) It must support the activities of students in the process of assimilation of knowledge and skills. It is intended to help improve communication between group members, help in organization and division of tasks within the team, allowing the integration of individual results as a group activity at work, and collaborate with the resolution of complex problems, through cooperation. These are some issues identified in the systematic review conducted during this research [9].

(4) It must be attractive to students. It must provide an exploratory environment, an intuitive environment that is conducive to discoveries and challenges, and responds to their interaction.

(5) The simplest is the best. The main goals for this visualization tool were simplicity, ease-of-use and effectiveness, due to the fact that both software engineering teachers and students have limited time to learn new concepts and will most probably lose interest in using the tool, if it imposes a steep learning curve as a prerequisite for getting familiar with it.

(6) The target 3D diagrams must have a similar 2D source diagram vision, so that it becomes a familiar landscape over extended use. The user must easily recognize his/her own diagram, even in a different guise.

(7) Visual enhancement is necessary to differentiate additional information to be explored and interacted. Using depth and color represent new views provided and presented in Section 3.3.

(8) Details like attributes, methods and association names are hidden to reduce visual pollution. With very large diagrams, it is essential to obtain clarity. However, they appear when requested by the user.

3.2. System Overview

VisAr3d is composed of three main modules: Architectural Module, Augmented Reality Module and Virtual Reality Module.

On a regular course, the teacher exhibits a large and complex system diagram projected on the wall or on a printed folder. It details the features, components and connectors, in different levels of abstraction, possibly using various architectural styles. Before using VisAr3D, the software architecture in study must had been created and documented within a software architecture editor. On the Architectural Module, the diagrams must had been built using an external UML editor and exported as an XMI file. VisAr3D is able to read this stored UML diagram.

VisAr3D uses the 3D technology, by using the ARToolKit libraries [10], to capture and recognize the 2D projection (teacher’s diagram), helping the teacher and students in identifying and quickly accessing this architecture on the Augmented Reality Module. A graphic pattern is captured by the webcam into the student’s computer. VisAr3D uses this feature to identify the architecture that is being studied and the exact viewpoint on this architecture. The Augmented Reality Module turns the object of study more accessible to the user and integrates it in a natural way in the classroom activities, bringing benefits to the student’s learning. Besides being easily usable, it ensures the display of models in the latest version in the computer network.

After recognition of the software architecture displayed in 2D, the Virtual Reality Module automatically generates an equivalent 3D UML diagram from it. This resulted diagram should be equivalent to the 2D diagram, displayed in an unlimited space with a slight depth.

3.3. VisAr3D Functionalities

VisAr3D, as a virtual environment, should enable the exploitation of the environment by the student with the help of the mouse, moving in space to the right, left, up, down, stay away or get closer to the plane and rotational movement.

The student should be able to visualize the 3D architecture in different angles and distances with the aid of icons and colors. VisAr3D must also allow students to get closer and realize the presence of more contextual information, and to go away until this information disappears. This feature allows the analysis of software architectures from a new perspective, and thus it is possible to discover similarities between parts of that diagram, to understand more complex relationships, different techniques, and architectural styles.

The amply space for data exploration in a 3D visualization environment facilitates the comprehension of diagrams with different levels of abstractions, or versions that have been designed by different people.

The functionality of viewpoint facilitates the movement of students at pre-defined points of interest. The teacher or the student can select (define) a position in the architecture as a new viewpoint and the user will be able to quickly move between these points, even when they are far away in the architecture.

Another facility provided by VisAr3D is a search agent that allows looking for documents by keywords or filters. The result can be a link list or a graphical visualization through colors or flags. This sub-system allows fast and easy access to all kinds of information associated to the software architecture in study. Zooming technique is also available, providing more details of the data shown on a certain view.

Besides allowing the navigation through all diagrams, there are defined views in the context of this work that represent different forms of visualization, intending to improve the quality of the course, in line with what is discussed in [11]. VisAr3D uses diagram views to visualize the system from different perspectives. No complex system can be understood in its entirety from only one perspective. The main idea is that different aspects of the system can be independently focused by choosing the right set of views. Some of these views are:

One class diagram View: To fully understand a class element in the model, it might be necessary to explore this architectural element: its attributes, methods, and associations that can be found in different diagrams. In this kind of visualization, the selected class element is centered on the screen as the user changes from one diagram to another.

Package View: Package View gathers the architectural elements in one Package.

Metric View: VisAr3D combines the existing layout of UML class diagrams with the visualization of metrics, using color, size and shape to visualize values.

Properties View: The user moves the cursor on the architectural element, and its attributes, methods and relationships names are displayed in a readable size in a window, when they were requested.

Author View: In the distance, colorful icons indicate the authors of the architectural elements.

Documentation View: While navigating through the architecture, students will have access to this superimposed information, as they move through the model space. It should be easy to perceive its correspondence to its content, identified by icons, colors or flags. As the user chooses to read one document, more than one architectural element, associated with it, are marked.

Annotation View: Through spontaneous and sometimes even playful situations, a student can explore, create, and especially communicate with other students or the teacher to obtain new information and to build his/her own learning process by sharing findings, doubts, or simple messages using notes.

Exercise View: This view allows the teacher to interact with students through exercises that can improve their theory knowledge.

4. Final Considerations

The paper presented the VisAr3D approach. Several important aspects were described in this paper to exploit the advantages of the 3D space to support the understanding of complex systems through a more attractive experience for the user.

VisAr3D uses 3D visualization technologies, such as Virtual Reality and Augmented Reality. It is a novel way to visualize and understand UML models of a complex system. With this approach both teachers and learners have a great opportunity to explore a 3D environment that emphasizes the intuitive understanding in a didactical context. There is a focus on the need of a practical experience (students feel greater motivation to see practical results in operation), bridging the gap between practice and theory; on the construction of a collective reasoning to understand concrete situations; and on the investment in the application of graphical interfaces and documentations to encourage students and to reduce the learning curve.

The approach lets users explore the various views without forcing them to follow a specified path. This should allow for more creativity and therefore interesting observations. They can find errors, examine them and learn through them.

To make the software architecture discipline more interesting and stimulating, VisAr3D aims that students regain the taste of mastering the knowledge in a pleasant way. The idea is to bring a greater dynamism to the classroom, resulting in a more meaningful participation of students.

However, it is still a work in progress and no evaluation has been carried out so far. The next steps are to complete the implementation of the prototype, which includes the integration with the other researches in the context of our research group, and to make the planning and execution of an evaluation to check the advantages and disadvantages of VisAr3D in concrete situations, including its level of efficiency and attractiveness. Currently, an observation experiment planning is being prepared.

5. References

[1] L. Feijs, R. D. Jong, "3D visualization of software architectures", Communications of the ACM, 41(12), 1998, pp. 73-78.

[2] C. Ware, D. Hui, et al., “Visualizing object oriented software in three dimension”, In: Proceedings of the 1993 Conference of the Centre for Advanced Studies on Collaborative research, IBM Press, Toronto, Ontario, Canada, 1993, pp. 612 - 620.

[3] P. Mcintosh, M. Hamilton, R. Van Schyndel, “X3D-UML: enabling advanced UML visualization through X3D”, In: Proceedings of the 10th International Conference on 3D Web Technology (Web3D 2005), ACM Press, Bangor, U.K., 2005, pp. 135-142.

[4] J. Gil, S. Kent, “Three dimensional software modeling”, In: Proceedings of the 20th International Conference on Software Engineering, IEEE Computer Society, Kyoto, Japan, 1998, pp. 105-114.

[5] O. Radfelder, M. Gogolla, "On better understanding UML diagrams through interactive three-dimensional visualization and animation", In: Proceedings of the Working Conference on Advanced Visual Interfaces, ACM Press, New York, USA, 2000, pp. 292-295.

[6] GEF3D, "GEF3D Eclipse Project Website", Available at: http://www.eclipse.org/gef3d/, Accessed in: October 2010.

[7] C. F. J. Lange, M. A. M. Wijns, M. R. V. Chaudron, "A Visualization Framework for Task-Oriented Modeling Using UML", In: Proceedings of the 40th Hawaii International Conference on System Sciences, IEEE Computer Society, Los Alamitos, CA, USA, 2007, pp. 289a-298a.

[8] Jacobson, M. and Working Group 2 Collaborators, Complex Systems and Education: Cognitive, Learning, and Pedagogical Perspectives, Available at: http://www.necsi.edu/events/cxedk16/cxedk16_2.html, Accessed in: Jan, 2011.

[9] C. S. C. Rodrigues, C. M. L. Werner, “Software Architecture Teaching: A Systematic Review”, In: Proceedings of the 9th World Conference on Computers in Education (WCCE 2009), Bento Gonçalves, Porto Alegre, Brazil, 2009, pp. 1 – 10.

[10] H. Kato, “ARToolkit”, Human Interface Technology Laboratory, University of Washington, Available at: http://www.hitl.washington.edu/artoolkit, Accessed in: October 2010.

[11] M. Shaw, “Software Engineering Education: A Roadmap”, In: Proceedings of the Conference on the Future of Software Engineering, 22nd ICSE, Limerick, Ireland, 2000, pp. 371-380.