2016 IEEE/ACM 38th IEEE International Conference on Software Engineering Companion

VisAr3D: an innovative 3D visualization of UML models

Claudia Susie C. Rodrigues
COPPE/UFRJ
Av. Athos da Silveira Ramos, 149/H318
Cidade Universitária,
Rio de Janeiro, Brazil
+55 21 3938-8785
susie@cos.ufrj.br

Cláudia M. L. Werner
COPPE/UFRJ
Av. Athos da Silveira Ramos, 149/H318
Cidade Universitária,
Rio de Janeiro, Brazil
+55 21 3938-8702
werner@cos.ufrj.br

Luiz Landau
COPPE/UFRJ
Av. Athos da Silveira Ramos, I 214
Cidade Universitária,
Rio de Janeiro, Brazil
+55 21 3938-8413
landau@lamce.coppe.ufrj.br

ABSTRACT
New challenges and demands on software engineering education are presented by rapid changes and increased complexity of software systems. This paper presents the VisAr3D environment that has been developed as an innovative proposal to be introduced in the classroom to provide a 3D visualization of UML models. The user is invited to intuitively understand the model elements in this 3D environment. It includes exploration and interaction to establish a practical and pleasant learning activity focusing on large scale systems. It provides a new way to visualize and understand UML models by combining the technologies of Virtual Reality and Augmented Reality. A 3D diagram is automatically generated from an existing 2D diagram and is able to provide richer semantics than its corresponding 2D diagram. The paper also presents some results of an experimental study conducted to investigate the feasibility of using the approach. The evaluation has provided positive evidence of its ability to improve the understanding of UML models in systems with many modeling elements. And also shows the contribution of the insertion of the third dimension on this understanding, in the interests of the students and in support to the practice of teaching.

CCS Concepts
• Social and professional topics~Software engineering education • Computing methodologies~Virtual reality • Software and its engineering~Software architectures

General Terms
Design, Documentation.

Keywords
Software Engineering Education; UML; 3D visualization; large scale system

1. INTRODUCTION
Software Engineering (SE) teaching has experienced pressures over the years as systems become larger and more complex with a great demand for quality. The academy teaches theory and concepts, and in practical terms, students write relatively small programs in a well-defined domain, sometimes in a small period of time. According to [1], many companies view people straight out of school as raw material to be shaped to the corporate culture. They are forced to supplement their education with training and preparation that provide them with knowledge that supplies this deficiency. Generally, students leave college without dealing with complex problems, changes in objectives during the project, problems with customers, pressure on the delivery time, demand for quality software or working with a big team, among others.

In [2], the author states that the distinguishing characteristic of industrial-strength software is that it is intensely difficult, if not impossible, for the individual developer to comprehend all the subtleties of its design. He also affirms that the complexity seems to be an essential property of all large software systems, and we should master this complexity. The study of complex systems deals with problems that are difficult to solve and often hard to understand because of their indirect effects. In particular, beginners, as students, for example, may face serious problems in understanding the details of such systems.

Given this situation, according to [2], software engineering has focused on reducing and managing complexity. In the past few years, we saw an increase in the effort put in searching new viable teaching methods that would help students acquire and retain knowledge. And the education community has invested much effort in developing new ways of teaching SE. New education proposals have been introduced in the classroom to meet these new demands, especially those that make teaching more attractive to students [3].

Based on these challenges, this research combines features of emerging technologies of 3D visualization technologies, such as virtual and augmented reality, recognizing that visual communication can be a key factor in the process of teaching and learning of future software architects due to their high expressiveness and power to be adapted to students' preferences. The tangible benefits of achieving this goal is to support the understanding of Software Models, including software architecture models, by developing the VisAr3D (Software Architecture Visualization in 3D) approach, which aims to provide the user the ability to manipulate and analyze a large amount of data from multiple perspectives through an attractive experience. UML is chosen for this research because it is a current and commercially accepted standard visual notation for visualizing software models.

Visualization in 3D space can significantly contribute in education by allowing to provide a process of exploration, discovery, observation and construction of a new vision of knowledge, helping the learner to better understand the object of study.

ICSE '16 Companion, May 14-22, 2016, Austin, TX, USA
© 2016 ACM. ISBN 978-1-4503-4205-6/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2889160.2889199

Therefore, this technology has the potential to collaborate in the student's cognitive process, providing not only the theory but also practical experience of the content in question.

To increase the comprehensibility of complex information systems and reduce the cognitive load of the user, McIntoch *et al.* solution is to also visualize the models in 3-dimensional diagrams [4]. 3D view provides substantial practical improvement to the way of working with large complex models. Alvergren and Granqvist say that one of the purposes of complementing 2D visualization with 3D is to be able to view more information in one view [5], allowing more complex information to be comprehended. And Marcus *et al.* add that when transformed into 3D, the semantic richness broadens enabling effects like objects having different elevation and transparency [6].

The remainder of this paper is structured as follows: Section 2 is dedicated to describe the pedagogical aspects that justified this research. Section 3 discusses 3d visualization. Section 4 describes the related work. Section 5 presents the VisAr3D approach. Section 6 describes the approach evaluation. Finally, Section 7 addresses some final considerations.

2. PEDAGOGICAL ASSUMPTIONS AND COMPREHENSION OF MODELS

This paper characterizes the need to support the teacher, the student and the software developer who face difficulties related to the description and understanding of software models of complex systems. The main objective is to enhance students' motivation by getting their whole personality involved in challenging activities.

What matters here is the students' access to the elaborated knowledge and thus making them receptive to this knowledge, but not passively. So they actively assimilate what is being exposed, that is, they receive the information, meditate on it, and make relationships with their experiences and knowledge.

Individuals are assumed to learn better when they are forced to discover things themselves rather than when they are instructed. The learner must have experience with hypothesizing and predicting, manipulating objects, posing questions, researching answer, imaging, investigating, in order to occur knowledge construction [7].

Large software system is very difficult for an individual developer to understand all the subtleties of the project. Even when the relationships between system elements are simpler, relatively static, and, in principle, understandable, there may be so many elements and relationships that understanding them is practically impossible. [2]. The study of complex systems is a challenging task, dealing with problems that are difficult to solve and, often, difficult to understand because they are open to varying interpretations. In particular, beginners, as students, for example, may face serious difficulties in understanding the details of such systems.

There are several research papers and projects that offer promising solutions as visualization techniques and applications. However, according to [8], these solutions are rarely marketed. He says that this is due to the difficulty of implementing comprehensive support for information visualization, having identified several problem areas related to understanding an information system model in general and integrating diagrams and objects within them, such as:

a) Having to understand a diagram and how it relates to other diagrams;

b) The need for providing an overview of the diagram;

c) The need for visual clues to help identify the interrelationships between diagrams;

d) Multiple views often lead to inconsistencies between them, or omissions, or errors in information system models;

e) The need for finding and showing reusable components, such as a property (like a name), an object (like a class), a diagram (like a class diagram), a view (like a static view), or even the entire model;

f) The large amount of data in a information system model, which can hide the main points of interest. Thus, a system must hide and show the details of the models when necessary, and shall allow zooming in and out;

g) The problem of managing different dimensions. A vertical dimension may be related to abstraction levels. A horizontal dimension may deal with different diagrams, but on the same level, e.g., use case diagram and class diagram.

3. 3D VISUALIZATION

Virtual Reality (VR) is defined by [9] as "a three-dimensional simulation of the real world or an imaginary world allowing the user to have a sense of physical presence and to manipulate 3D objects, in real time, inside three-dimensional computer-generated environments." The use of VR as means to support learning is not something new, having been suggested by [10]. Many studies have shown that VR can be used as a form to better understand abstract concepts [11]. [12] appoints the possibility of showing situations and concepts that the user would not be able to view otherwise and its immersive nature as the main advantages in using VR for education. The interest and enthusiasm caused by that technology is also mentioned as a positive point.

3D visualization can help in the production of an image to software, aiding at the process of software maintenance and evolution by improving the understanding of the software by the user. Visualization can help produce an image of the software. It creates a physical object (visual) that represents the software system, and provides software engineers with a gain of some early insight on how it is structured and composed. It helps them to use their perception skills in research and software comprehension, and also to materialize something that is abstract or logical.

[13] emphasize some important features to consider when designing a 3D visualization: visualizations should provide as much information as possible without overwhelming the user; 3D visualization helps to reduce visual complexity; the information content should vary to suit the interests of users; metaphors introduce familiar visual concepts to the user and provide a good starting point for obtaining an understanding of the visualization; the interface is user friendly and provides intuitive navigation, trying to avoid that users easily get lost; it correlates information between different possible visualizations; and, 3D interaction should provide mechanisms to obtain more information and also help maintain the user's interest.

According to [14], the great advantage of 2D drawings is that they require only paper and pencil. However, regarding some aspects, one can show where 3D overcomes 2D: it is possible to make a layout of objects in 3D environment more consistently than can be done in a single plane [15]; in a 3D environment, the order in space can be expressed and understood much better than in 2D; animation is important in the 3D environment due to the effects and highlights; 3D models allow a visualization semantically richer than 2D models [16]; regarding the perception of 2D orientation, 3D reduces the cognitive overload and improves the users' perception; in 2D view, only one diagram can be seen at a time, and in a 3D view, several diagrams may be displayed in a single scene; three-dimensional views are important for certain applications, especially for studying the relationships between various models or diagrams [17]; and, according to [15], 3D projects have a visual appeal and may be more attractive to students, therefore attractive to the learning process.

4. RELATED WORK

Although several approaches to diagram integration in three dimensions exist, they are not yet well established in software design. Previous research has shown that there are practical advantages in using 3D for UML visualization. For example, Mcintosh *et al.* [4] focused on the "visual" aspect of UML as the means of communicating information to the viewer. The main difference regarding our approach is how they present and explore the 3D environment to provide a means for understanding complex software system models. Geon [18] is a diagrammatic method for visualizing UML class diagrams. Their research has indicated that the Geon diagram is both easier to understand and easier to remember than traditional UML class diagram. This important requirement, i.e., to be easier to understand, is also present in the VisAr3D approach. Gil and Kent [19] 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 also intuitive from different perspectives. Radfelder and Gogolla's work [20] applies static and dynamic diagrams into a single view.

Dwyer [21] explores the use of a Force Directed Algorithm to layout three dimensional UML class diagrams. His prototype incorporated the construction and edition of UML models as well as simple visualization.

The GEF3D framework [17] builds on the Eclipse Graphical Editor Framework and as such enables existing Eclipse 2D UML editors to be extended into 3D. And in [22] the MetricView framework that supports task-oriented modeling and developed views and visualization techniques is proposed. The VisAr3D approach also uses diagram views to visualize the system from different perspectives.

VisAr3D, like most of the works cited above, also aims to facilitate understanding of a large scale system. Nevertheless none has the proposal to be an environment to support teaching and learning. It facilitates the teacher‘s task, the students' learning, and the communication among them, being an important resource.

5. VisAr3D APPROACH

The VisAr3D - Software Architecture Visualization in 3D - approach has been designed to provide a 3D visualization of a UML model automatically generated from an existing 2D diagram. This solution, detailed in [23], explores different perspectives, relationships, abstractions, and contextualized documentation, where the user should intuitively understand the modeling elements in this 3D environment. The idea is to reuse all the existing and relevant information related to the modeling elements of a system. These important data will be associated to the corresponding modeling elements and the user should explore and interact with them in the third dimension.

This approach allows students to participate and explore large scale models and later be prepared to build their own. For the purpose of this work, a large system, or complex system, is defined as any software system composed of a great number of parts with many interactions/associations.

VisAr3D is proposed to be a 3D viewing environment and, as a limitation of the approach, it does not allow editing of models in 3D. Its target audience is the teacher and graduate students.

By developing a 3D visualization system for the display of models in a new perspective, it is possible to deal with more elaborate and complex designs, similar to those developed by industry. An intuitive exploration and interaction by software modeling students and by the teacher, using resources and facilities such as the manipulation of models in a simulated learning environment by means of virtual elements, brings an attractive experience to the learning process.

The VisAr3D approach brings a new dimension to the practice of the Software Modeling discipline and proposes a different solution to support the understanding of models: (i) it provides support for interactive practical sessions, including the use of videos, animations, documentation etc.; (ii) it allows the creation of an environment of information exchange among students and teachers; (iii) it allows a new form of contextual information available to a 3D software model; (iv) it intends to surprise students with new ways to highlight existing and relevant information and relationships; and (v) it allows the student to intuitively interact in a 3D environment to understand a system with many modeling elements proposed in the classroom.

5.1 VisAr3D 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 small part of 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 model in study must had been created and documented within a software model editor. On the Architectural Module, all types of diagrams (class diagrams, use cases, sequence diagrams etc) must had been built using an external UML editor and exported as an XMI file [24]. VisAr3D is able to read this stored UML diagram. On the Augmented Reality Module, VisAr3D uses the 3D technology to capture and recognize the 2D projection (teacher's diagram), helping the teacher and students in identifying and quickly accessing the model. By using the ARToolKit libraries [25], a graphic pattern is captured by the webcam into the student's computer. VisAr3D uses this feature to identify the model being studied and its exact viewpoint.

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 developed prototype, this module needs tweaking and a few changes or new commands will need to be implemented to improve it. The idea is to use small notes on the graphic patterns containing relevant information to the student or teacher before they enter the three-dimensional environment.

After recognizing the software diagram displayed in 2D, the Virtual Reality Module automatically generates an equivalent 3D UML diagram from it (Figure 1 shows a snapshot of the automatic generation of a 3D model from a 2D model). This means that any

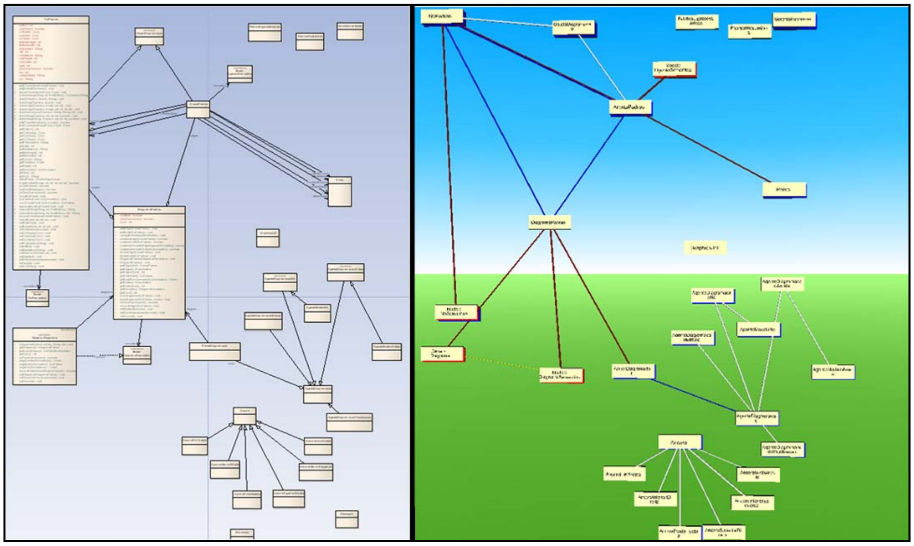

Figure 1: VisAr3D automatically generates a 3D diagram (on the right) from a 2D diagram (on the left).

2D diagram previously modeled in a UML tool and exported as an XMI file can be automatically displayed in 3D on VisAr3D. The resulted diagram is equivalent to the 2D diagram, displayed in an unlimited space with a slight depth.

5.2 VisAr3D Functionalities

VisAr3D, as a virtual environment, enables the exploitation of the environment by the student with the help of the mouse, moving in space to the right, left, up, down, staying away or getting closer to the plane and rotational movement. That, along with the ability to move the view around the diagram in a tridimensional world, not only allows the user to focus on specific parts of a diagram, but also enhances the experience, allowing a certain degree of immersion, and therefore increasing the chances of a better understanding.

The student is able to visualize the 3D diagram in different angles and distances with the aid of icons and colors. VisAr3D also allows 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 models from a new perspective, and thus it is possible to discover similarities between parts of its corresponding diagrams, to understand more complex relationships, different techniques, and architectural styles.

The ample space for data exploration in a 3D visualization environment facilitates the comprehension of models with different levels of abstractions, or versions that have been designed by different people. It helps to identify an important modeling element (a class, for example) that belongs at the same time to multiple diagrams and to learn more about it.

As previously mentioned, a corresponding UML diagram in three dimensions is generated from a 2D model by the VisAr3D tool. Its modeling elements are added with a depth that can contain, depending on the case, several types of information that will be explored by the student. They are relevant concepts that are highlighted and explicit to the student. An existing slight depth in a class, in red and blue color, indicates that it belongs to more than one diagram, and it contains documentation associated to it, respectively.

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

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 to quickly and easily find and access all kinds of information associated to the software model in study.

Zooming technique is also available, providing more details of the data shown on a certain view.

According to Shaw [26], practical and useful software projects do not just happen. They require student skills. She points out deficiencies in education and suggests solutions. In line with her work, views in the context of the present work that represent different forms of visualization are defined, intending to improve the quality of the course. They allow the navigation through diagrams. VisAr3D uses diagram views to help the user 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. The approach allows users to explore the different views, without forcing them to follow a specific path. The intention is to encourage greater creativity and also to make interesting observations. Initially, it was decided to use a second window for the menu of views and for displaying the information requested about the diagrams. Some of these views are:

One class diagram View: One class in a model can be presented in several diagrams. It is often difficult and tedious to find out on which place a class is in the model, because this relation is not explicitly presented in the diagram. To fully understand this class element, it might be necessary to explore this modeling 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. It is well-known that the comprehensibility of a system is mainly determined by the possibility of studying the system one part at a time. As the part of the system that is relevant for the user's purpose is showed, the target of the learning process is greatly reduced [27].

Package View: Package View gathers the modeling elements in one Package. If this package diagram is available, the user should access this level of abstraction by just pressing a key.

Metric View: VisAr3D combines the existing layout of UML class diagrams with the visualization of metrics. Since applying metrics to a UML model can result in an overwhelming amount of data, and these data are usually presented in tables, it is a hard work for a software engineer or a student to make the mapping between metrics values in a table and classes in UML diagrams manually. VisAr3D supports this activity by integrating the model and metric visualization, using color, size and shape to visualize values.

Attributes/Operations View: There is information that only appears when requested by the user through the Attributes/Operations View, to avoid pollution of information associated to large diagrams attributes, methods, and names of relationships. The user moves the cursor over the modeling element, and its attributes, methods and relationships names are displayed in a readable size in a window.

Author View: Likewise, VisAr3D allows visualization of the authorship of the model. In the distance, colorful icons indicate the authors of the modeling elements. In the Author view, the name of the author is presented on the output window when the user moves the cursor over the modeling elements.

Documentation View: The development of an architectural design often produces a large amount of documentation, as well as records of the initial decisions about the project to ease the communication between stakeholders. This information is very important for software modeling students, especially for the novice ones, to understand the whole process of creation and software development. While navigating through the model, 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 modeling element, associated with it, are marked. It can be associated to a video, animation or to an image. The intention here is to value good documentation and encourage students to always document.

Annotation View: VisAr3D contains information generated within the space. It serves to facilitate the understanding of software models, and consequently better communicate among team members who build the system. 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. The participants can add arbitrary annotations to particular artifacts.

Exercise View: This view allows the teacher to interact with students through exercises that can improve their theory knowledge. This view is similar to the annotation and documentation views but is categorized as exercises. The students can be asked to choose between design alternatives, be required to make choices to address customer needs, be asked to reuse components, to adhere to standards and also to correct erroneous elements, be encouraged to compete in using "Case Studies", or be challenged to refactor a piece of model.

The developed prototype only implements part of the features of the approach. Currently, VisAr3D provides a static view of a large scale system model, emphasizing the static structure of the system.

5.3 Implemented Prototype

The approach, which has in its title the words “Software Architecture Visualization in 3D”, is currently restricted to UML models visualization. UML has been chosen for this research because it is a current and commercially accepted standard notation for visualizing software models. The approach is ambitious, even though only a part of it has been implemented as a prototype. VisAr3D prototype was developed with the main objective to show the feasibility of the applied technology. The prototype has some interface and usability limitations, however it helps to test some important assumptions about the approach, in terms of functionality, technology and its contribution being a 3D environment. This prototype version only runs on Windows platform.

VisAr3D prototype was built using Java, X3D [28], Xj3D [29] and XMI (XML Metadata Interchange) [24] version 2.1. XMI is a standard from OMG (Object Management Group) that uses UML information to create XML documents. Through the use of XML, UML information can be more readily transferred between tools. The use of XML also allows the information to be more readily transformed into new forms, such as 3D.

X3D is an open standard and is a trademark of the Web3D Consortium. X3D has an XML encoding and allows users to download, display, and interact with 3D content via a web browser plug-in or a standalone viewer. X3D is the enhanced successor to Virtual Reality Modeling Language (VRML). The forms are described by geometrical shapes, and behaviors can be controlled from the scene, both internally and externally, by the X3D file for programming languages or script.

Xj3D is the Web3D Consortium’s open source project that provides a toolkit for writing X3D content and is developed in Java. By combining Xj3D and Java, one can easily add 3D capabilities to any application. Actually, they are implementing further improvements to Xj3D, considering how to stabilize, improve and accelerate.

The developed Augmented Reality Module uses the ARToolkit library, C++ and OpengGL [30], which is an API for creating 2D and 3D graphics applications.

Currently, to use VisAr3D, an XMI file is imported after being created in a UML tool and diagrams are generated in three dimensions, automatically from it. Following the information provided by the XMI file extracted using Java commands, the VisAr3D can: identify all modeling elements of each diagram created and their original positions in the 2D model; recognize all different types of associations, related to each class, or package, locating them in different diagrams; recognize all the attributes and operations of each class; recognize the packets, to which each class belongs; identify the different diagrams to which a class or package can belong; recognize the authoring and documentation associated with the elements modeling. After interpreting all of this information, VisAr3D positions the diagrams in the virtual world, adding a small thickness to the elements and adding icons and colors to them.

Figure 2 shows an example of the main screen of the prototype. It is divided into two parts: a virtual environment where the diagrams are displayed and a window that displays output information about the modeling elements.

Each 3D diagram created in the virtual environment is explained below: classes are virtual objects in 3D rectangular yellow, identified by name; relationships are colored tubes that connect the classes. In this version of the prototype, the directions of the relationships were not designed by interface decision reasons. They have different colors indicating the different types. The navigation buttons that allows movement in the virtual environment are located at the bottom of the output window. The output window provides the current diagram name and the menu that accesses the functionality of VisAr3D.

6. VISAR3D APPROACH EVALUATION

An experimental study was conducted to evaluate the feasibility of the VisAr3D approach (detailed in [23]). The focus of the study was to investigate the ability of the approach to contribute to the understanding of UML class model of a system with many modeling elements (in this study it was used approximately 240 classes divided into 13 diagrams). According to the participants' report, the advantage of using this type of system is to offer the learner the opportunity to work with a problem closer to reality, where most everyday situations can be explored and various modeling conditions may be exemplified. Thus, cases consistent with what he/she will handle in the job market are presented.

This study was used to test the following hypothesis: "Using the VisAr3D tool, students solve modeling tasks with more precision and more efficacy than using a traditional 2D modeling tool".

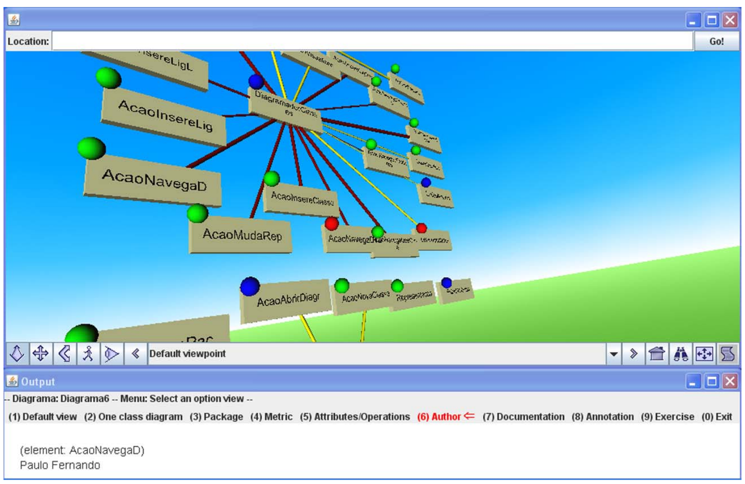

Figure 2. VisAr3D prototype screenshot.

The independent variables of the study were the tools used: VisAr3D and a traditional 2D tool. And the dependent variables were precision, efficacy and time. However, this study also helped to evaluate the following points: (1) if the third dimension contributed to support the understanding of UML models in systems with many elements, (2) if the environment in the third dimension was able to increase students' interest in relation to the 2D environment; and (3) if the third dimension gave more support to the practice of teaching in projects with many elements in relation to the 2D environment (considering the opinion of the participants, since the approach has not actualy been used in a teaching situation).

For the preparation of the experiment, firstly, a pilot study was executed, where two researchers reviewed the plan and the tools used, both with previous experience in teaching systems modeling. These researchers evaluated the difficulty of the tasks and the time required for the solution, and also assisted in the reformulation of the forms used in the study.

6.1 Planning

The goal of the study can be formalized using GQM – Goal/Question/Metric paradigm [31] as:

analyze	to analyze the use of the VisAr3D tool compared to using a traditional 2D modeling tool
for the purpose of	Characterize
with respect to	precision, recall, time and user perception of the use and adoption of new technologies
from the point of view	software engineers
in the context of	execution of tasks within the Systems Modeling discipline, using a system with many modeling elements

In this study, subjects played the role of students of the Systems Modeling discipline and answered some questions related to a system that has many modeling elements. They were divided randomly into two working groups, who performed the same tasks, but in different orders. They solved six tasks using a traditional 2D modeling tool (first step) and six similar tasks using the VisAr3D prototype tool in 3D (second step), and vice-versa.

Based on [32], the elaborated tasks were divided into three groups with different levels of difficulty: Filtering, Basic and Assimilation Tasks. The tasks dealt with the contents as follows: making the identification of a class diagram a simple task, to understand a system; showing the class, not only within a diagram, but within the system as a whole, analyzing relationships and inheritance; using examples of operations analysis, such as polymorphism; verifying the authorship of these classes, i.e., the team's participation in that diagram; exploring the given solution in the modeling form from a Java code etc.

Participants in the experiment were selected by convenience. They met the following prerequisites: (i) prior experience with UML modeling, (ii) academic formation (all participants were undergraduate and graduate students at COPPE/UFRJ, 7 doctoral, 5 masters students and 6 undergraduate students), and (iii) willingness to participate in the study. Of the total 18 participants, 28% had experience of teaching of Software Systems Modeling and 44,5% had relevant experience in UML modeling in industry.

50% of participants had experience with systems with more than 50 classes, a significant rate that contributed to the evaluation result. However, only 2 out of 18 participants (11%) used a system with more than 200 classes within a class.

6.2 Execution

All the necessary material for the study was installed in one computer, including two training videos on the use of the VisAr3D prototype and on the use of the modeling tool for 2D UML models. Participants were free to ask questions for clarification, while accompanied by the researcher, who took notes. They filled out four types of forms: Consent Form, Characterization of the Participant, Tasks (questions that had to be answered) and Evaluation Questionnaire.

6.2.1 Quantitative Analysis

In this study, precision can indicate exactness, which refers to the number of valid responses, in relation to the number of participant responses (it calculates the correct answers in relation to the number of responses of the participant).

Recall refers to an indicator of efficacy and completeness, and it determines the number of valid responses in relation to the expected number of correct answers (i.e., the total number of template responses).

Before using the statistical tests, an analysis of outliers for each metric was done. The new overall average of participants' results were recalculated per tool (the means and standard deviations are shown in Table 1).

The tasks of the study were prepared in such a way that could be resolved by the two tools, however the main objective was to observe how each participant performed each of them and what their needs and expectations were, using each tool. Therefore, it was expected that rates of precision and efficacy were not very different, as confirmed by the data analysis.

The participants were able to achieve their goals and managed to solve the tasks, with indicators of precision and recall (or efficacy), on average, near in the two tools, as shown in Table 1 (98.15%, using the 2D tool against 98.88%, using VisAr3D tool, in precision, and 96.94%, using the 2D tool, against 97.96%, using the VisAr3D tool, in recall). As might be expected, there was a slight advantage for VisAr3D.

Due to the characteristic of each one, it is not very useful to compare the average time spent on solving each exercise: in some cases, complex exercises led to short answers and easy exercises took too long to be answered. However, in a general context, it took longer, on average, to solve the tasks using the 3D tool compared to 2D tool, as shown in Table 1 (5.43 minutes versus 4.78 minutes). According to the participants, this can be explained by the fact that they are more familiar with 2D patterns. It was also observed that participants spent more time in the first step of solving the tasks than in the second, this is due to the learning effect during the execution of the study.

If there are close values in recall variables in the two tools means having the same performance in both tools, i.e., means being independent of the technology used. If the value of the precision variable is greater using the 3D tool, it means that students agreed with more questions using this tool.

After the statistical analysis performed, it was concluded that there is no statistically significant difference in precision, recall and time variables in the use of VisAr3D and 2D tools, i.e., the experimental study concludes that using the VisAr3D tool, students solve the tasks of modeling with the same precision, efficacy and time than using the 2D tool. Considering the fragility of a prototype with a new proposal for navigation and interaction, compared to a well-used commercial tool, this is still a very positive result for the quantitative analysis.

6.2.2 Qualitative Analysis

Some participants experienced difficulties in comparing such different viewing environments, they said, "it is unfair," as one is a commercial tool and the other an academic prototype. However, the VisAr3D prototype was highly rated, although being a new environment with a new navigation and exploration proposal, and having inherent limitations of a prototype. The traditional 2D visualization commercial tool adopts Windows standard that is already known, making it a comfortable navigation, even for those who never used this tool before.

Table 1. Average of the results of participants

	2D Tool		VisAr3D Tool	
Metric	Mean	Standard deviation	Mean	Standard deviation
Precision	98.15%	2.52%	98.88%	2.53%
Efficacy	96.94%	5.41%	97.96%	4.20%
Time (minutes)	4.78	1.33	5.43	1.25

Among the positive points of the VisAr3D tool cited by participants are: "Information about the authors, available in the diagram through the graphic elements, do not pollute the screen", "The depth of the class, which reports the existence of documentation and that it is present in more than one diagram", "It shows many detailed information contained in the model in a user-friendly visual representation", "It is easy to find information, it requires just a few mouse clicks to get to the necessary information, and it also has an intuitive navigation" (it is easy to find detailed information about the model, like attributes, operations etc.), "clarity of diagrams, new form of interaction with UML diagram (attractive), ease of commands (just move the mouse)", "the way to see the model is richer. We can see the capability of displaying more than one diagram in the same view (perspective), certainly facilitates learning".

The negative aspects, cited by participants are: "The mechanisms of navigability of the tool need to be improved", "Resolution of the diagram when zoomed out", "Lack of direction in relationships", and "The window that shows the information should be attached to the visualization tool".

Table 2 summarizes the best contribution by comparing the 2D and 3D visualizations, in the opinion of participants, according to some topics. This table presents the answers of participants in the Evaluation Questionnaire. Their answers were illustrated with helpful comments and relevant information to the qualitative analysis of the study. The participant chose which tool, in his/her perception, better contributed according to the topics shown in the table rows. Participants considered the column “not applicable” as a doubt between the two tools and in other situations as "neither option". In most cases, each choice was accompanied by a commentary. It was observed that almost in all topics VisAr3D was identified as the best choice.

The main purpose of a feasibility study, according to [33], is not to find a definitive answer, but create a body of knowledge on the application of technology. In this sense, it would be possible for the researcher to evaluate whether application of the technology is feasible, i.e., if it meets the goals initially set in order to justify or not to continue the research. Thus, the result of this feasibility study was positive, showing that the VisAr3D tool fulfills its purpose to support the understanding of UML models in systems with many modeling elements.

In summary, Table 2 shows that, qualitatively, the 3D tool provides positive evidence to solve tasks in the context of the Systems Modeling discipline. It also answers the main questions of the study with the aid of the participants’ responses to the Evaluation Questionnaire, which leads to the following conclusions: (1) the third dimension contributed to support the understanding of UML models in systems with many elements, (2) the environment in the third dimension increased students' interest in relation to the 2D environment, and (3) the third dimension gave more support to the practice of teaching (considering the opinion of the participants) in projects with many elements in relation to the 2D environment.

Table 2. Best contribution by tool according to some topics

	2D Tool	3D Tool	Not applicable*
Support the Understanding of UML in large systems	22%	56%	22%
Support to exploration of models	33%	56%	11%
Support for the resolution of tasks	17%	61%	22%
Complexity Reduction	17%	61%	17%
Decreased visual pollution	6%	83%	11%
Easy to read details of the information	22%	44%	11%
More intuitive environment	28%	44%	28%
Ease of obtaining information about authorship	0%	94%	6%
Ease of obtaining information about relationships	44%	33%	22%
Ease of obtaining information about a class on more than one diagram	6%	83%	11%
Ease of obtaining information on documentation	0%	94%	6%
Support teaching practice	11%	83%	6%

(*)"Not applicable" was considered by the participants as doubt between one tool and the other, and as neither of the two.

6.3 Threats to Validity

Some threats to validity [34], related to this study, can be summarized as:

- The study assumes that the subjects are representative for the population of system modeling students. It is not possible to confirm their previous experience with UML;
- The study was not performed in the same day by all subjects. This may have influenced the results;
- It is not possible to inform if the time measurement was accurate;
- We tried to control the risk of training effect providing the same training videos for all subjects;
- We tried to minimize the learning effect by asking the subjects to solve the same tasks using the two tools in alternate orders;
- It was not possible to ensure that subjects would not release any information or results to another subject before the end of the study, although they were asked to;
- Participants were allocated to groups randomly, because they did not have the same ability to solve the tasks;
- Although the sample size was limited from a statistical point of view, it is considered reasonable in experiments in software engineering;
- The 2D tool uses similar graphical interface to any other 2D UML tools, and it is not guaranteed that the measured time and performance obtained with other 2D UML tools have the same magnitude;
- The researcher who conducted the experiment was the same who developed the prototype;
- A small number of exercises was selected in order to not tire and discourage participants during the resolution of tasks;
- The participants' perceptions about the three different levels of difficulty of adopted tasks may not be the same;
- Some participants were already familiar with the 2D tool, but they have never used the VisAr3D prototype. And yet, the 2D tool uses the Windows standard, which all participants are already familiar with.

7. FINAL CONSIDERATIONS

This paper presented the VisAr3D approach - Software Architecture Visualization in 3D. General aspects of the approach were described in this work, including the results of a experimental study. This evaluation of the approach aimed to ensure that it exploits the advantages of 3D space for understanding of UML models in large-scale systems, including their level of precision and efficacy in teaching practice, as well as its attractiveness.

VisAr3D developed a new way to visualize and understand UML models, by combining the technologies of Virtual Reality and Augmented Reality. With this approach, teachers and students have the opportunity to explore a 3D environment, which emphasizes an intuitive understanding of the learning context. There is a need to focus on practical experience in the construction of collective reasoning in understanding concrete situations, and investment in the implementation of graphical user interfaces and documentation to encourage students and reduce their learning curve.

The approach allows users to explore different views, without forcing them to follow a specific path. The intention is to encourage greater creativity and also to make interesting observations. They may find errors, analyze and learn from them, and even share them with colleagues. By using the VisAr3D approach the student may regain the taste for mastering knowledge in a pleasant way. The idea is to bring greater dynamism to the classroom, resulting in a more significant participation of students.

From the confirmation of the contribution of the third dimension, we intend to explore the third dimension through: the combination of static and dynamic diagrams in the same environment, ease of exploration of relationships not explicit in conventional diagrams, highlight of polymorphic operations, comparison of models, visualization of the evolution of models, and the access to other available types of diagrams, such as use case diagrams or sequence diagrams. By improving the navigation performance, we plan to invest in stereoscopic projection in order to make user interaction even more flexible, dynamic and interesting to the learning process.

This approach provides an opportunity to deepen the understanding of practice and all matters connected to the Systems Modeling discipline. It allows the operation and interaction of students of this discipline, using the resources, and present facilities in 3D virtual environment, such as: access to multimedia documentation in audio format, images and videos, combined with the modeling elements, enabling the visualization of models at various angles and interaction thereof in an unlimited space with all modeled diagrams available.

7.1 Contributions

The main contributions of the research, are:

1) Specification of a new environment for teaching and learning the Systems Modeling discipline based on 3D technologies. This environment presents important resources that allow the user to:

a) Support the understanding of UML models, enabling communication between users, and exploring complex systems;

b) Support the practice of the Systems Modeling discipline;

c) Provide a dynamic, modern and intuitive environment, which attempts to increase the interest of students;

d) Strengthen UML diagrams by adding relevant information about the project, organizing and overlaying the diagrams and making them available to users;

e) Analyze diagrams in different details and abstractions;

f) Support both students and teacher without requiring a learning curve as a prerequisite, which provides support to their target audience;

g) Invest in teamwork and group discussion and to support the instructor;

2) Development of the VisAr3D prototype to demonstrate the viability of the technology.

3) Planning and execution of the experimental study, which allowed the analysis of important issues about the approach.

The VisAr3D approach is an innovative proposal that can be used in the classroom, in order to provide the operation and interaction of UML models through 3D visualization. It also represents a framework of an environment to develop and evaluate new research proposals in teaching software engineering. It encourages the use of different methods, techniques and virtual and augmented reality devices that can help students to acquire and retain knowledge.

7.2 Limitations

From a critical analysis of the proposed approach and its implementation some limitations were identified. It had only one developer, and not a multidisciplinary team, as it is recommended for a 3D environment that supports teaching and learning.

Participants pointed out limitations of the prototype related to its performance and usability during the evaluation. Another limitation of the current work is that the prototype only displays static diagrams.

There are also limitations of the evaluation undertaken as the sample size is considered small from a statistical point of view, though acceptable for Software Engineering studies.

7.3 Ongoing Research

New researches are already underway. One aims to expand VisAr3D to support the understanding of concurrency in oriented object systems through Virtual Reality. Another proposes a tool that uses interaction techniques of motion control as a method of immersion, to support the learning of system modeling.

And even new researches invest in using other interactive devices such as Oculus Rift, CAVEs, data-gloves, allowing new experiences to students and the teacher.

7.4 Future Works

Future work will adopt Unity3D as the primary development platform of virtual elements. One of the factors that led to the exchange of technology was the development environment, as well as support by the community and the easy integration with virtual reality devices that could allow a more immersive and attractive experience.

And then, a new evaluation can provide analysis of objective evidence related to the approach.

8. ACKNOWLEDGMENTS

This research was partly supported by CAPES, CNPq and FAPERJ.

9. REFERENCES

[1] Thompson, J. B. and Edwards, H. 2009. Preparing Graduate Students for Industry and Life Long Learning: - A Project Based Approach. In *Education and Technology for a Better World: 9th World Conference on Computers in Education-WCCE 2009* (Bento Gonçalves, Brazil, July 27-31, 2009).

[2] Sommerville, I. *et al*. 2012. Large-scale Complex IT Systems. *Commun ACM* 55, 7 (July. 2012), 71–77.

[3] Emam, A. and Mostafa, M. G. 2012. Using Game Level Design as an Applied Method for Software Engineering Education. In *17th International Conference on Computer Games - CGAMES* (Louisville, USA, July 30 - August 1, 2012). http://dx.doi.org/10.1109/CGames.2012.6314583.

[4] Mcintosh, P., Hamilton, M. and Van Schyndel, R. 2005. X3D-UML: enabling advanced UML visualization through X3D. In *Proceedings of the 10th International Conference on 3D Web Technology - Web3D 2005* (Bangor, U.K, March 29 - April 01, 2005).

[5] Alvergren, J. and Granqvist, J. 2009.3D Visualization of Complex Software Models: Practical Use and Limitations. *Bachelor of Software Engineering and Management Thesis*. University of Gothenbur. Sweden, May.

[6] Marcus, A., Feng, L., and Maletic, J. 2003. 3D Representations for Software Visualization. In *Proceedings of the 2003 ACM symposium on Software visualization* (San Diego, USA, June 11-13 , 2003).

[7] O'Loughlin, M. 1992. Rethinking science education: beyond piagetian constructivism toward a sociocultural model of teaching and learning. *Journal of Research in Science Teaching*, 29 (8) (Oct. 1992), 791–820.

[8] Huotari, J. 2004. Integrating UML views with visual cues. In Second International Conference on Coordinated and Multiple Views in Exploratory Visualization (London, England, July 15, 2004).

[9] Milgram, P. and Kishino, F. 1994. A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information Systems*, 77(12) (Dec. 1994), 1321-1329.

[10] Hsu, H. J. 2011. The Potential of Kinect in Education. *International Journal of Information and Education Technology*, 1 (5) (Dec. 2011), 365-370.

[11] Yellowlees, P. M. and Cook, J. N. 2006. Education About Hallucinations Using an Internet Virtual Reality System: A Qualitative Survey. *Academic Psychiatry*, 30(6) (Nov. 2006), 534-539.

[12] Bell, J. and Foglerl, H. S. 1995. The Investigation and Application of Virtual Reality as an Educational Tool. In *Proceedings of the American Society for Engineering Education Annual Conference* (Anheim, USA, jun 25-29, 1995).

[13] Young, P. and Munro, M. 1998. Visualising Software in Virtual Reality. In *Proceedings of the IEEE the International Workshop on Program Comprehension* (Ischia, Italy, June 24-26 1998).

[14] Alfert, K. and Fronk, A. 2000. 3-dimensional visualization of Java class relations. In *Proceedings of the The Fifth World Conference on Integrated Design & Process Technology* (Dallas, EUA, June 4-8, 2000).

[15] Feijs, L. and Jong, R. D. 1998. 3D Visualization of Software Architectures. *Communication of the ACM*, 41(12) (Dec. 1998), 73-78.

[16] Sanatnama, H. and Brahimi, F. 2010. Graph Drawing Algorithms: Using in Software Tools. *Journal of Applied Sciences*, 10(17) (2010), 1894-1901.

[17] Pilgrim J. V. and Duske, K. 2008. Gef3D: a framework for two-, two-and-a-half-, and three-dimensional graphical editors. In *Proceedings of the 4th ACM symposium on Software visualization* (Ammersee, Germany, September 16 - 17, 2008, 2008).

[18] Casey, K. and Exton, C. 2003. A Java 3D implementation of a geon based visualisation tool for UML. In *Proceedings of the 2nd international conference on Principles and Practice of Programming in Java* (Kilkenny City, Ireland, June 16-18, 2003).

[19] Gil, J. and Kent, S. 1998. Three dimensional software modeling. In *Proceedings of the 20th international conference on Software engineering - ICSE 1998* (Kyoto, Japan, April 19 - 25, 1998).

[20] Radfelder, O. and Gogolla, M. 2000. On better understanding UML diagrams through interactive three-dimensional visualization and animation. In *Proceedings of the working conference on Advanced Visual Interfaces - AVI'00* (ACM Press, Palermo, Italy (Palermo, Italy, May 24 - 26, 2000).

[21] Dwyer, T. 2001. Three dimensional UML using force directed layout. In *Proceedings of the Australian symposium on Information Visualisation* (Australia (Sydney, Australia, December 3 - 4, 2001).

[22] Lange, C. F. J., Wijns, M. A. M., and Chaudron, M.R.V. 2007. A visualization framework for task-oriented modeling using UML. In *Proceedings of the 40th Hawaii International Conference on System Sciences* (Waikoloa, USA, Jan 3-6, 2007).

[23] Rodrigues, C.S.C. (2012). VisAr3D - An Approach Based on Emerging Technologies in 3D for Supporting the Comprehension of UML Models. 2012. PhD. Thesis (in Portuguese). Universidade Federal do Rio de Janeiro. Rio de Janeiro. Brazil.

[24] XMI - Object Management Group: http://www.omg.org/spec/XMI/. Accessed in: October, 2015.

[25] Kato, H. 2011. ARToolkit, Human Interface Technology Laboratory, University of Washington, http://www.hitl.washington.edu/artoolkit. Accessed in: October, 2015.

[26] Shaw, M. 2000. Software Engineering Education: A Roadmap. In *Proceedings of the 22nd Conference on the Future of Software Engineering - ICSE* (Limerick, Ireland — June 04 - 11, 2000).

[27] I. Hammouda, O. Guldogan, K. Koskimies, and T. Systä. 2004. Tool-supported customization of UML class diagrams for learning complex systems, In *Proceedings of the 12th International Workshop on Program Comprehension - IWPC 2004* (Bari, Italy, Jun 24-26, 2004).

[28] Web3D: http://www.web3d.org/. Accessed in: October, 2015.

[29] Xj3D: http://www.xj3d.org. Accessed in: October, 2015.

[30] OPENGL. http://www.opengl.org/. Accessed in: October, 2015.

[31] Basili, V., Caldiera, G., and Rombach, H. 1994. Goal question metric paradigm. John J. Marciniak (ed.) *Encyclopedia of Software Engineering* (1), John Wiley & Sons, New York, 528-532.

[32] Knodel, J., Muthig, D., and Naab, M. 2006. Understanding software architectures by visualization - an experiment with graphical elements. In *Proceedings of the 13th Working Conference on Reverse Engineering - WCRE '06* (Benevento, Italy, October 23-27, 2006).

[33] Shull, F., Mendonça, M., Basili, V., Carver, J., Maldonado, J., Fabbri, S., Travassos, G., and Ferreira, M. 2004. Knowledge-Sharing Issues in Experimental Software Engineering. *Empirical Software Engineering*, 9(1-2) (Mar. 2004), 111-137.

[34] Wohlin, C., Runeson, P., Höst, M. *et al.* 2000. Experimentation in Software Engineering: An Introduction. 1 ed. Norwell, USA, Kluwer Academic Publishers.