A Study on the Application of the PREViA Approach in Modeling Education

Marcelo Schots, Claudia Susie C. Rodrigues and Cláudia Werner
System Engineering and Computer Science
COPPE/UFFRJ
Rio de Janeiro, Brazil
{schots, susie, werner}@cos.uffrj.br

Leonardo Murta
Instituto de Computação
Universidade Federal Fluminense (UFF)
Rio de Janeiro, Brazil
leomurta@ic.uff.br

Abstract—Creation, interpretation, and evolution of models and diagrams are activities commonly performed in software projects, and their teaching is fundamental for the student education. However, it is difficult to see the changes made in model corrections, evolutions, or maintenance done by the student, as well as by the teacher, comparing templates and different solutions. In this regard, this paper presents the use of the PREViA approach, which aims at visualizing software architecture models. This work focuses on supporting teaching and learning in the modeling education context. A feasibility study was conducted and its results provided positive evidences of the use of the approach in the correction of diagrams.

Keywords—Software Architecture/Modeling Education, Software Visualization

I. INTRODUCTION

Our society is increasingly supported by a software based technology, with a great demand for quality. As systems become more sophisticated and larger, software engineering (SE), as well as its teaching, becomes vital. This is part of the pressure experienced by professionals of SE education, which over the years has mobilized the community to meet these demands.

Software architecture (including its models and architectural views) has received special attention from the software engineering (SE) academic community [1]. The study of the possible achieved benefits and the direct consequences of its application have leveraged software architecture to the status of discipline, becoming one of the main topics of SE [2] [3]. As software-based systems reach a larger scope and complexity, the importance of specification and design increases. Aiming at systematizing the software design process, the architecture provides a solid basis for reuse, allowing a greater standardization of structural systems representations.

The Unified Modeling Language (UML) provides a standard way for the creation, visualization and documentation of architectural design [4], including conceptual aspects (e.g., business processes and system functions) and concrete items (e.g., classes, database schemas and reusable components) [5]. In this sense, UML class diagrams can be considered one of the architectural views of a system [6].

A systematic literature review of the initiatives undertaken in teaching Software Architecture [7] has reaffirmed the needs for an education program that allows practical experience and attempts to deepen the understanding of skills in the formation of a good software architect.

Visual communication is a key factor in the teaching and learning process of software architecture and systems modeling [8] [9]. Understanding architectures is not a trivial activity, which consumes time and effort, due to factors such as [10]: (i) the high complexity of systems, (ii) the large amount of data, and (iii) the high number of dependencies among their architectural elements. For these reasons, the process of acquiring knowledge should be supported by visualization approaches and tools to reduce the effort of understanding by improving awareness of the behavior, structure, and evolution of the system.

In this sense, software visualization can be benefited from and applied to many different areas. In education, students can use visualization to facilitate the understanding and learning of new concepts in software development.

Considering this scenario, this paper presents the first results of a feasibility study regarding the use of PREViA approach to support activities related to model differencing, aiming at improving comprehension in learning. The study has provided positive evidences of the use of the approach in the context of diagram correction.

Besides this introductory section, this paper is divided as follows: Section II describes the PREViA approach; Section III presents the use of PREViA in the context of modeling education. Section IV describes the evaluation of the approach in this context. Section V details some related work and Section VI presents the conclusions, some limitations, and future work.

II. THE PREViA APPROACH

Software architectures evolve over time due to several reasons, and it is important to keep the consistency between the developed applications (represented by emerging architectures) and their conceptual architecture anytime in the software development lifecycle. The PREviA approach (Procedure for Representing the Evolution through Visualization of Architectures) aims at visualizing the evolution of software architectural models via the comparison of model versions (stored in version control systems). This is done by using concepts and techniques of software visualization to improve comprehension and perception of the evolution [11].
The architectural differences obtained by diff engines are not intuitive (sometimes restricted to textual representations) and can require a large time and effort to be comprehended [12]. To address this problem, software visualization techniques can provide a better representation of such information by allowing a simpler, more intuitive way of understanding the underlying data [13], making software artifacts and behavior “visible” by using visual representations. According to Mukherjea & Foley [14], the visualization of such information is particularly important for allowing people to use the perceptive reasoning (instead of using cognitive reasoning) when performing these tasks, which facilitates detection of patterns that reveal the underlying structure in the data.

The approach originally includes two perspectives: (i) the adherence of the emerging architecture to the conceptual architecture over time, and (ii) the differences between two distinct versions in the architectural evolution process. However, as the aim of this work does not include evolution, the time dimension was not taken into account and only the first visualization perspective was adopted, making the model template (i.e., the model expected as the result of the exercise) to be the conceptual model and each model developed by the students is considered as a distinct emerging architecture (which does not compound an evolutionary cycle).

Regarding the implementation of PREViA, an Eclipse plug-in was built on top of EvoTrack [15], a mechanism for capturing and visualizing the evolution cycle of a software project considering its timeline. UML was used as proof-of-concept for representing the low-level module view (Eclipse provides plug-ins for working with UML diagrams); with PREViA, the elements (in terms of the metrics obtained) are marked by using specific UML profiles, with customized stereotypes according to the metric (e.g., the kind of difference between the models).

III. APPLICATION OF PREViA IN THE MODELING EDUCATION

The detection of divergences among models is an important activity in terms of teaching and learning. From the students’ perspective, the comparison between their model and the teacher’s model allows the interpretation of the successes and mistakes made during a modeling exercise (as scientists have long known, mistakes are conducive to learning). However, the understanding of divergences between the models is not a trivial activity. Likewise, the teacher may have difficulties in correcting modeling exercises, especially considering that the elements that differ from the template model may mean different design alternatives. These exercises may contain many elements, and the comparison is usually done by using handmade models.

Thus, it is believed that PREViA can provide support for comparison and understanding activities, facilitating this type of interpretation, especially when such activities occur repeatedly, which can cause them to become tiresome and inefficient. The use of such an approach stems from the importance of discussing and investigating aspects of knowledge construction by the teaching of architecture and modeling systems disciplines.

Regarding the visual perception of differences between architectures, the elements in which any difference kind is detected are highlighted by the PREViA approach. Models elaborated by each student are overlapped to the template, and the latter is shown in the form of a watermark that becomes darker when a match between elements from both models occurs. So, for this use, the highlights made with the red color indicate elements that were inserted in the answer, but do not exist in the template; the gray color indicates elements that appear on the template, but were not included in the answer. Elements that were found in both models (i.e., that represents a match between them) are represented in black.

IV. EVALUATION

A feasibility study was conducted in order to evaluate the application of the PREViA approach in the modeling education. The focus of this study was the use of the PREViA approach in the evaluation of high-level models created by students and corrected by the teacher. The main purpose of this study was to explore how teachers consider divergences between models; due to that, there was no previous training in model/architectural conformance or inspection.

A. Planning

Following the GQM template [16], the goal of this study can be described as follows: analyze the use of PREViA approach (object) for the purpose of exploring (purpose) with respect to facility to interpret and understand divergences between models (quality focus), from the viewpoint of teachers and trainers (viewpoint) in the context of software architecture and systems modeling disciplines (environment).

Two exercises (A and B) were used for the generation of the models. It is noteworthy that these models are real and have been prepared in the context of a Brazilian discipline of systems modeling and design (thus, the models are presented in Portuguese language). The topics used for the exercises (scholar and enterprise domain) were simple and well known by both students and subjects. This exercise focused specifically on design at the analysis level (the attribute and methods were not part of the exercise).

The study sample was selected for convenience. Participants were selected by meeting the following prerequisites: (i) previous experience in teaching modeling systems and/or software architectures, (ii) academic formation (all participants are graduate students: one master student, two PhD students and a PhD professor) and (iii) availability to participate in the study. All the material that was needed for the study was sent via email. There was no presence support to participants.

B. Execution

The study was divided into two stages. In the first stage, participants received an exercise template A (i.e., a model that was expected as the result of the exercise A) and
answers given by two students (i.e., models generated by students in the modeling exercise) to compare, without the application of the PREViA approach. Figure 1 shows an example of a template and an answer from a student without the application of the approach.

In the second stage, participants received an exercise template B and answers from two other students, with the indications of convergences and divergences identified by the PREViA approach (as described in Section III). In order to decrease learning bias, the domain topic was changed between the stages (the group that worked with the scholar domain in the first stage worked with enterprise on the second one, and vice-versa). Figure 2 shows the output of the PREViA approach over a student’s answer (the input for this result were the models presented in Figure 1).

In this figure, the “Cargo” class represents an expected element (as it is defined in the exercise template), but was not found in the student’s answer. Furthermore, the “Empresa” class was found in the student’s answer, but was not planned, according to the exercise template. Another example is the relationship between “Funcionário” and “Dependente” classes: Besides the difference in names, a composite association was expected, but the student’s response contains a simple association.

For this exercise, there were seven possible divergences:
- missing class (i.e., a class that was expected, but was not present in the answer);
- unintended class (i.e., a class that was not expected, but was present in the answer);
- different association multiplicity (for example, [0..*] instead of [1..*]);
- different association name (for example, “tem” instead of “possui”);
- missing association (i.e., an association that was expected, but was not present in the answer);
- unintended association (i.e., an association that was not expected, but was present in the answer);
- different association kind (related to aggregation or composite associations – for example, an aggregation instead of a composite association).

In both stages, it was requested for participants to complete a form with the following information: (i) time spent (ii) elements identified as divergences, and (iii) a grade evaluating the exercise. The grade was used as means of verifying how correct the given exercise is (in the subject’s opinion) and whether correctness of exercises can somehow influence the evaluation). The remaining information was used for analyzing the level of detail in each description of divergences and for calculating efficiency, efficacy, precision and recall (these metrics are described in Section III.C).

It is important to note that the PREViA approach provides only highlights of the differences between the models, leaving to the user the task of verifying whether each difference characterizes an error or a design alternative. An example of such situation can be found in Figure 2 (the words “possui” and “tem” are synonyms in Portuguese).
Such information was shared with the participants during the study. After that, a follow-up questionnaire with eight questions was distributed for collecting some qualitative data about the study and the PREViA approach, in order to verify strengths, deficiencies and improvement opportunities.

C. Quantitative Analysis

Quantitative data were used for checking whether the use of PREViA was useful for the participants to find the divergences. In most cases, the use of PREViA allowed the participants to achieve better results.

Regarding the possible divergences that could be found in this study, the missing classes, the unintended classes, and the different association kinds were the most easily identified by the participants (100% of the occurrences in the exercise were found), followed by the missing associations (67% of the occurrences were found) and the different association multiplicities (59% of the occurrences were found).

Table I shows the averages of the participants’ results.

<table>
<thead>
<tr>
<th>TABLE I. AVERAGES OF PARTICIPANTS’ RESULTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without PREViA</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>No. of divergences</td>
</tr>
<tr>
<td>No. of valid divergences</td>
</tr>
<tr>
<td>Time spent (minutes)</td>
</tr>
<tr>
<td>Efficiency (*)</td>
</tr>
<tr>
<td>Efficacy</td>
</tr>
<tr>
<td>Precision</td>
</tr>
</tbody>
</table>

(*) divergences per minute

As the exercises selected for the study were simple, the participants found similar amounts of valid divergences, with or without PREViA (7.25 and 6.25 divergences on average, respectively). However, less time was spent when using PREViA (a difference of 2.5 minutes, on average). Although in a more complex exercise this difference could possibly increase, there is not enough evidence for such assertion.

In this study, efficiency refers to the time spent during the divergences identification, while efficacy refers to the correct identification of divergences. These metrics can be calculated by the following equations:

\[
\text{Efficiency} = \frac{\text{number of divergences found}}{\text{time spent}}
\]

\[
\text{Efficacy} = \frac{\text{number of valid divergences found}}{\text{total number of existing divergences}}
\]

For measuring efficiency, it is necessary to consider all the divergences found (including incorrect and duplicated ones). With the approach, the divergences were detected at a higher average efficiency (0.83 divergences per minute against 0.60 divergences per minute on average, without PREViA).

The efficacy has also reached higher values with the approach (0.65, against 0.63 without the use of PREViA). After analyzing the issue, this could be explained by the meaning of divergence for one of the participants – for example, he did not consider some different association names as divergences. However, one participant stood out and has reached 100% rate of success. Although he took a long period of time performing the exercises (13.25 minutes on average), this was the highest rate for valid divergences.

This can be explained by his past experience with inspections: due to this profile, he probably took special care in describing the differences, and may have been meticulous (that is, with conscientious attention to details).

Precision and recall metrics [17] can indicate, respectively, the exactness and completeness of the answers regarding the exercise template (recall and efficacy measures are calculated in the same way, so the former is not shown on the table, as it has the same values). In terms of these measures, no significant difference was noticed. Especially regarding recall measure, the participants had, on average, almost the same results with and without PREViA (as described for the efficacy metric).

D. Qualitative Analysis

All the participants found it easy to use PREViA for comparing models and interpret the data. Also, none of them had difficulties while interpreting information marked by PREViA. These are evidences for one of the goals aimed by using the approach (i.e., support for comparison and understanding activities, facilitating this kind of interpretation). Some of them mentioned that it would be difficult to perform model differencing activities in large-scale models, and indicated the need to automatically obtain information about the types of divergence among the models. This can be achieved by the use of the PREViA tool, which uses EMF Compare’s algorithm [18] in order to obtain differences between the models.

A positive feedback was given by the participants: “there is no need to look at the model template all the time”; “the model assessor does not bother with what is not relevant”; “[the approach] has greatly facilitated the examination of differences in the multiplicities of relationships, which is not so easy to see without this feature”.

One downside noted by participants in the use of PREViA is that the elements that have the same semantics, but different names, are considered as a divergence by the approach. This point becomes more critical when the number of elements increases. However, the approach does not intend to do a semantic check, restricting itself to the syntactical aspect (a semantic analysis would involve the processing of natural language, which is not the main focus of the approach, but can be a future work).

Moreover, it is possible to observe that the use of PREViA may hamper the correction when the number of divergences is much higher than the number of similarities, because it generates a very large “visual pollution” and the perception that the approach aims to provide is greatly affected. A comment made by one of the participants corroborates this assertion.
Another negative remark was the indication of the differences between relationships. In the current version, the tool that implements the PREViA approach is not able to highlight only the divergent part of the relationship (i.e., only the difference of name, type, or cardinality) – for instance, the only difference between the relationships “está lotado” and “lotado” (Figure 2) is the name of the relationship – the type and multiplicity have matched the exercise template. This is a limitation of the UML plug-in used by the tool. Finally, one participant has found the use of gray color difficult to read in the diagram; this can be handled through the PREViA tool, which allows the customization of colors.

As suggestions for improvement, two participants indicated that a list of all divergences could be generated automatically, leaving to the teacher the task of deciding whether or not these divergences represent an error. They argued that this may facilitate the work of the final correction of the model. This list is already generated automatically, but was not used during the study. It will help to check divergences, when needed, and in the improvement of this rate.

E. Learned Lessons

Recalling Section II, the PREViA approach was planned to deal with evolutionary scenarios, where the elements usually does not differ too much from the previous version (or are intended to become less divergent as the software is coded). This is not true in the educational scenario, because the compared models are results of a creative design process, where each designer can achieve a solution in a different way (but not necessarily incorrect).

Regarding what the participants considered divergence, it was possible to note that, when they indicated a missing class, they did not indicate a missing association between this class and another one. So, when a class is missing, they do not seem to consider the missing association as a divergence. This is not true for an unintended class (both the class and the association between this class and another one were considered as divergences, in most cases).

By analyzing the grade given by the participants to each exercise answer, an interesting point is that the exercises with lower grades had the worst results using PREViA. The approach did not lead to improvements in efficiency and efficacy in such cases, as some of the participants using PREViA had lower values of these metrics when compared to participants without PREViA. A hypothesis is that exercises with more errors can be harder to evaluate; still, the meaning of divergence was not formalized in the study for each participant. Although they had their own strategy for punctuating and grading the exercises, this is an issue that should be studied further.

This study with few participants represents a very important step for the evaluation of PREViA both in the context of education and in its original format. The answers, the indicators, and comments were very interesting and helpful, and were considered for improvements in the studies related to PREViA.

V. Related Work

Through literature review and the systematic literature review mentioned in Section I, some approaches that somehow relate to this study were found.

Some works such as [19], [20], [21], and [22] are examples of approaches that were directed to support SE teaching (by using games, for instance), but they are not geared specifically for systems modeling and software architecture.

The modeling tool proposed by Wang & Scannell [23] helps students to learn and conduct analysis to evaluate the quality of architectural designs alternative. It resembles the use of the PREViA approach for two reasons: (i) it invests in visualization features to encourage students in order to decrease the learning curve, and (ii) it uses metrics to increase awareness of the activities results. However, the visual representation used in their work is restricted, with coarse-grained elements, in order to meet the focus of the authors. The PREViA approach gives a richer, more detailed representation, and allows navigation between different levels of abstraction, including grouping of diagram packages.

The ArchE tool [24] works as an assistant to the software architect. Like PREViA, both students (who interact from suggestions made by the tool as the architectural design is created) and teachers (for teaching software architecture) can use it. One of the differences between the ArchE tool and PREViA is that the former aims at demonstrating the meeting of required quality attributes; another difference is that the PREViA approach supports the perception of students and teachers via visualization techniques (including zoom, clustering and drill-down). Also, the main view of PREViA displays diagrams that represent architectural model elements.

VI. Conclusions

This paper presented a study of the application of the PREViA approach in the modeling education context. The main benefits of the PREViA approach are to facilitate the comparison of models and the interpretation of data, which was confirmed after the study results. This indicates that the use of PREViA may contribute in these software engineering education activities.

A limitation of this study is the impossibility of generalizing results, because it is a feasibility study with few participants. Some threats to validity include: (i) the proximity between the participants and the researchers, (ii) the small number of exercises used, and (iii) the choice of exercises, which took place in a non-random form.

The evaluation described in this paper had an exploratory nature and, as such, has raised some interesting issues and suggested the potential of the PREViA approach in the educational context. Nevertheless, it is intended to perform future studies with a larger sample of participants, for the purpose of providing more conclusive evidences. A study using the approach with the students is ongoing. There are also ongoing improvements in performance, features, and functionality of the PREViA implementation.
ACKNOWLEDGMENT

The authors would like to thank the subjects for their valuable contributions to the study. Furthermore, we thank CNPq, CAPES, and FAPERJ for their financial support.

REFERENCES


