

Effects of Model Composition Techniques on Effort and Affective States: A Controlled Experiment

Mateus Manica, Kleinner Farias, Lucian J. Gonçalves, Vinícius Bischoff
PPGCA, Unisinos, RS, Brazil
mateusman@edu.unisinos.br
kleinnerfarias@unisinos.br,
{lucianj,viniciusbischof}@edu.unisinos.br

Bruno C. da Silva
Department of Computer Science,
Software Engineering, Cal Poly.
San Luis Obispo, USA
cbcda Silv@calpoly.edu

Everton Guimarães
Department of Computer Science
Drexel University, Philadelphia, USA
etg38@drexel.edu

Abstract—Even though existing heuristics and specification-based techniques support composing design models, it is still considered a time-consuming and highly intensive task. In addition, there is a lack of studies exploring the effects of composition techniques on software developers’ affective state and development effort. This study reports a pilot study to investigate these effects while developers apply composition techniques to detect and resolve inconsistencies in output-composed models. In this sense, a widely known wearable EEG headset, namely Emotiv EPOC, with 14 channels was used, while developers made use of heuristic-based and specification-based composition techniques to evolve design models. Our results suggest that using heuristic-based techniques produced a higher effect on the developers’ affectivity, compared to specification-based techniques. Moreover, the higher the effects on the developers’ affectivity, the higher the odds to invest less effort and produce correctly composed design models.

Keywords- UML; Cognitive Effort; Empirical Evaluation; EEG.

I. INTRODUCTION

Model composition has been used for supporting many software-development activities, e.g., adding new features to evolve design models. The term *model composition* may be briefly seen as a set of tasks that should be realized over two (or more) input models, M_A and M_B , so that an output-intended model, M_{AB} , can be produced [1]. Given the input model elements of M_A and M_B may conflict with each other, developers must often spend some effort to solve such conflicts before producing, M_{AB} . In recent years, researchers and practitioners have focused on producing composition techniques [2][3] as means to obtain M_{AB} from M_A and M_B , by investing as little effort as possible.

These techniques could be classified into two categories: (1) *specification-based techniques* (e.g., Epsilon [2]), where developers explicitly specify both correspondence and composition relations between the elements of M_A and M_B to produce M_{AB} ; and (2) *heuristic-based techniques* (e.g., override, merge and union algorithms [4]), in which developers use a set of predefined heuristics for “guessing” the correspondence and composition relations between the elements of M_A and M_B . However, in fact, both categories may lead *composition conflict problems*. Composition conflicts consist of contradictions between the values assigned to the properties of design models [7][8]. For example, a UML class *Researcher* assigned as a concrete class (i.e., *Researcher.isAbstract* = false), whereas in

the M_B delta model the class *Researcher* is set as an abstract one (i.e., *Researcher.isAbstract* = true). These contradicting values assigned to *isAbstract* represent a conflict that must be solved by developers. However, if this issue is not properly addressed, inconsistencies are inserted into the output-composed model M_{CM} . For example, *Researcher.isAbstract* = false represents an inconsistency as the expected value would be true. To sum up, composition techniques cannot guarantee an optimal solution, and usually developers produce an output-composed model, M_{CM} , with inconsistencies—typically after solving conflicts between M_A and M_B improperly. That is, usually M_{CM} and M_{AB} do not match ($M_{CM} \neq M_{AB}$) [1][5]. In this sense, developers often need to invest some extra effort to detect and resolve such inconsistencies.

For this reason, composing design models is still considered as being an error-prone and time-consuming task [5]. Recent studies have shown the relationship between development practices and their effects on the developers’ cognitive activities [13]. However, nothing has been done to reveal to what extent the use of composition techniques might influence the developers’ affective states, including frustration, excitement, and meditation. Therefore, this paper seeks to apply neuroscience methods to analyse and understand affective states, while software developers perform compositions to support the evolution of design models. This study is important to identify cognitive factors that may affect the practice of software modelling. Using too much effort to manipulate models may explain the selective use of UML models [12]. For this, a pilot controlled experiment was performed for grasping the effects of composition techniques on the developers’ affective states by monitoring their cognitive processes. We have used Emotiv EPOC wireless EEG headset [6] with 14 channels to collect and process affective states in 288 integration scenarios of model elements of UML class diagrams [13] in the context of 18 evolution scenarios.

II. EXPERIMENTAL DESIGN

A. Objective and Research Questions

The goal of this study is to **analyze** compositional techniques **for the purpose of** investigating their effects **with respect to** effort, correctness and affectivity **from the perspective of** software developers **in the context of** evolution of software

design models. The goal of this work was formalized using the GQM template [9]. Therefore, three Research Questions (RQ) emerged from this objective:

- **RQ1:** What is the relative effort of composing two input models using specification-based composition techniques with respect to heuristic-based composition techniques?
- **RQ2:** Is the number of correctly composed models higher using specification-based techniques with respect to heuristic-based ones?
- **RQ3:** Does the use of heuristic-based technique cause a higher effect on the developer's affectivity than technical based Specification?

B. Hypotheses

These research questions seek to explore how cognitive processes triggered to perform composition techniques may influence (1) the effort invested by developers to integrate design models, (2) the correctness of the compositions, and (3) the affective states generated during composition tasks respectively. For this, to answer these questions the following null-hypothesis were formulated.

H1: Null Hypothesis 1, H_{1-0} : the specification-based composition technique requires less effort (or equal to) than heuristic-based technique to produce M_{AB} model from M_A and M_B by the developer.

H2: Null Hypothesis 2, H_{2-0} : The specification-based composition technique produces a smaller or equal number of models properly composed than the heuristic-based composition technique.

H3: Null Hypothesis 3, H_{3-0} : The use of the specification-based composition technique causes a lower (or equal) impact on the affectivity than the heuristic-based technique.

C. Study Variables

The dependent variables of H1 are associated with each part of the model composition effort equation: $effort(M_A, M_B)$, representing the general effort for composing two models; $f(M_A, M_B)$, the effort to apply composition techniques; $diff(M_{CM}, M_{AB})$, the effort required to detect inconsistencies; and $g(M_{CM})$, the effort required to resolve the inconsistencies. All these variables are measured in minutes.

The dependent variable of H2 is the correctness (Cor) of the output-composed models. If $M_{CM} = M_{AB}$, then the composition was correct (Cor = 1); otherwise $M_{CM} \neq M_{AB}$, then the composition was incorrect (Cor = 0). The hypothesis regarding the correctness (H2₁) evaluates which technique produced more correctly composed models.

The dependent variables of H3 are affective indicators: (i) *engagement* (Engaj(M_{CM})) is the result of experience in the alert, attention to the task status consciously. When its levels are negative mean a high surveillance or even boredom; (ii) *meditation* (Medit(M_{CM})) represents how much calm or relaxed the user has been during the experimental task; (iii) *excitement* (Excit(M_{CM})) determines a condition of sensorial alert and response readiness; and (iv) *frustration* (Frustr(M_{CM})) reflects the reaction from the feedback obtained in practice, or can even express stress. These variables are quantified using the Emotiv EPOC. Finally, the independent variables of H1, H2, and H3 are

the heuristic-based and specification-based composition techniques. We observed the use of these techniques in the proposed scenarios to evaluate the impact on dependent variables.

D. Context and Subject Selection

Our pilot study had three subjects, which used two techniques (i.e., Epsilon and traditional algorithms) to perform six evolution scenarios of design models. These models were used because they were already validated in a previous empirical study [1]. In this way, we can also reduce threats to validity of the study results. All the subjects of this study have experience on software modeling and programming. Each participant was exposed to the same level of training about the modeling and composition techniques so we can make sure they will share the same knowledge related to model composition.

E. Experimental Process and Study Setup

The controlled experiment had been organized into 4 different activities as depicted in Figure 1. We assigned the participants to treatments randomly and equally distributed following the within-subjects design, in which all participants in the study will run all activities related to the treatments [10]. In each treatment, participants used a modeling composition technique to perform the 6 experimental tasks.

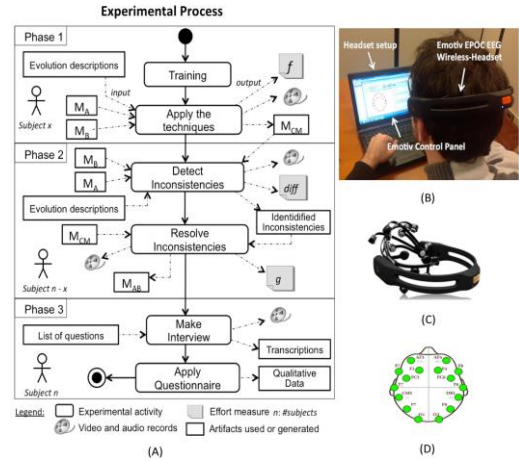


Figure 1. Experimental process (A), Experimental setup (B), Emotiv EPOC (C) and its electrode positions (D).

The activities performed by the participants are described as follows: (1) *training*, each subject received training to assure that all participants had familiarity with the composition techniques; (2) *apply the techniques*, the subjects used the techniques Epsilon and traditional algorithms to integrate two input models, M_A and M_B ; (3) *detect inconsistencies*, the next step was to read the composed model produced to detect inconsistencies; (4) *resolve inconsistencies*: Having identified inconsistencies, the participants were encouraged to solve them, seeking to produce the M_{AB} ; and (5) *make interview and Answer questionnaire*, the authors inquired the participants to think aloud about their experience throughout the experiment.

III. STUDY RESULTS

A. RQ1: Effort and Composition Techniques

Figure 5 highlights the effort invested in each required

activity to combine two input design models. We observed developers tend to invest less effort to produce the output-intended model (M_{AB}) using heuristic-based techniques rather than the specification-based technique. That is, when participants used Epsilon language they ended up investing more than twice effort to produce the output model. On average, developers invested by about 28 min to run the experimental tasks using a heuristic-based technique. Thus, they spent about 63 min to perform similar tasks using the specification-based techniques. For example, the first participant (a) invested twice more effort using the specification-based technique than using the heuristic-based one.

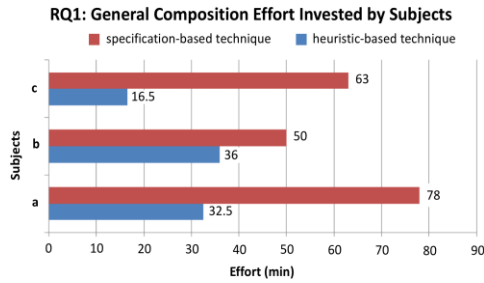


Figure 2. The invested effort to compose design models.

This upward trend was also observed with the other participants. In the second participant (b), the effort increased from 36 min using the heuristic-based technique to 50 min, representing a rise of 38%. Similarly, the third participant (c) also spent a superior effort to compose design models using the specification-based technique. The participant invested 16.5 min using the heuristic-based technique, and 63 min using the specification-based technique. When using heuristic-based technique the participant invested 41% of the effort to compose the design models, compared to the specification-based technique. The results confirmed the findings reported in [6].

Finally, the general result of the H1 is that for all participants applied more effort using the specification-based technique, which suggests the rejection of the null hypothesis ($H1_{1-0}$), and the possible confirmation of the alternative hypothesis ($H1_{1-1}$). We also observed the specification-based technique required more effort to detect and resolve inconsistencies (Figure 5), also confirming the alternative hypothesis H1.

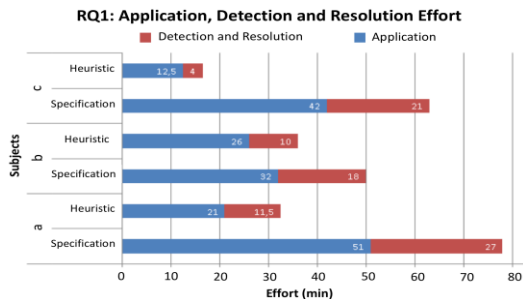


Figure 3. Effort invested in each step of the composition process.

Conclusion of RQ1: Developers tend to invest more effort to combine two input models, detect and resolve inconsistencies using a specification-based technique, compared to a heuristic-based technique.

B. RQ2: Correctness and Composition Techniques

Figure 6 shows the obtained results. The general correctness, i.e., the number of correctly composed model, was 5 using the heuristic-based technique, while 2 using the specification-based technique.

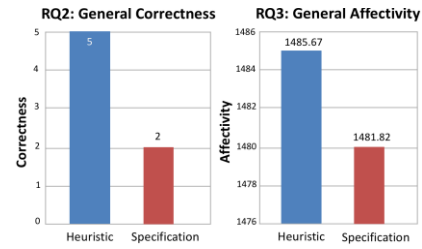


Figure 4. The correctness of the output-composed model (left), and the general affectivity of subjects using heuristic- and specification-based technique (right).

We observed the heuristic-based technique produced a higher number of correctly composed model compared to the specification-based one. As previously mentioned, when using a specification-based technique, the expectation is that the number of models produced correctly, $M_{CM} = M_{AB}$, is enhanced, given its flexibility to elaborate the composition rules. However, the results showed the opposite. The specification-based technique has not produced a higher number of correctly composed models, neither generated models with a lower inconsistency rate. Thus, the result suggests the rejection of the null hypothesis ($H2_{1-0}$) and confirmation of the alternative hypothesis ($H2_{1-1}$).

Conclusion of RQ2: Developers tend to produce a higher number of correctly composed models using heuristic-based technique, compared to specification-based technique.

C. RQ3: Affectivity and Composition Techniques

Figure 7 presents the results about the impact of the composition techniques on the developers' general affectivity. The main finding is that the heuristic-based technique produced a higher effect on the developers' affectivity than the specification-based technique. Thus, the result suggests the rejection of the null hypothesis ($H3_{1-0}$) and confirmation of the alternative hypothesis ($H3_{1-1}$).

To better explore this issue, we have examined five facets of developers' affective state, including frustration, engagement, mediation, long-term excitement, and instantaneous excitement. Figure 7 shows the measures the affectivity using heuristic-based technique, while Figure 8 presents the results regarding the specification-based technique. The results show developers' affectivity using the heuristic-based technique outnumbered the ones produced using the specification-based technique.

The results revealed a pattern regarding the measures of affectivity, since the two participants (a and c) obtained very similar indicators, and the participant (b) had different measures. On the other hand, Figure 7 does not present the same pattern regarding the use of the specification-based technique. However, participant (b) had still the higher engagement and frustration indicators. By examining the average engagement of the participants, we observed the value is 5% higher using the heuristic than specification-based technique.

Due to the results presented by the participant (b), we questioned the influence of the frustration indicator on the

quality of the compositions. If the relationship between high levels of frustration and engagement and low excitement would not reflect the participant (b) was under pressure. We point out the pressure can have many different causes, such as personal competitiveness, time available for executing activities, social pressure, among others. Based on the initial results, it is still necessary to carry out further studies to improve knowledge, understanding, as well as to translate it into more realistic results about the influence of affectivity of the participants in the quality of the compositions.

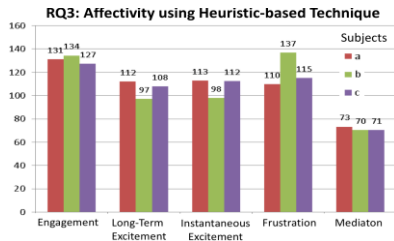


Figure 5. The impact on affectivity using heuristic-based technique.

The specification-based technique (Figure 8) had a minor impact on developers’ affectivity compared heuristic-based approach. Therefore, the data suggest that null hypothesis H3 (H3₁₋₀) is confirmed as the specification-based technique showed a lower impact on affectivity. While the specification-based technique requires a greater effort and composition produces a lower amount of correctly composed model, it had a minor impact on the variable affectivity of developers. We also observed the participants had a higher engagement and less frustration using the heuristic techniques.

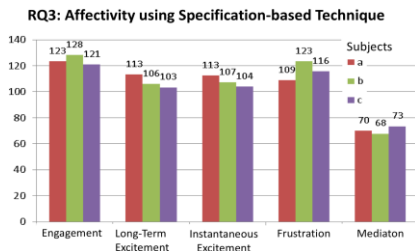


Figure 6. The impact on affectivity using specification-based technique.

Conclusion of RQ3: Specification-based technique tends to cause a lower impact on the affectivity of the developers, compared to heuristic-based technique.

IV. RELATED WORK

A controlled experiment for exploring the benefits of UML models on the comprehensibility of Java source-code deprived of comments is presented in [14]. The authors concluded that an additional effort was observed to read UML models, but this was paid back in the form of an improved comprehension of source code. In [11], the author mentions program comprehension as the main activity of the software developers. They emphasize that even though a huge amount of research to support the programmer has been done, the high amount of time developers has to grasp source code remained constant over thirty years. The author mentions that the use of EEG could be a reliable way

to measure cognitive load of programmers. However, nothing is presented in this sense. To the best of our knowledge, this work is the first to (1) explore the influence of composition techniques on the developers’ effort, correctness, affective states in the current literature, and (2) provide an initial discussion on the interplay between composition techniques and affective states.

V. CONCLUSIONS AND FUTURE WORK

The results suggest the specification-based technique required a higher composition effort, produce a lower amount of output-intended model, as well as caused a lower impact on the developers’ affectivity, compared to its counterpart. As future work, we seek to replicate the study so that a larger sample of data can be produced, allowing hypotheses testing using statistical methods. Finally, this study showed that it is possible to explore affective states to mitigate their impacts on the correctness of composed models. Our expectation is that the issues outlined throughout the paper can encourage other researchers to replicate our study in the future under different circumstances. Finally, we see this paper as a first step in a more ambitious agenda to support empirical assessment of model composition techniques, as well as understanding cognitive and emotional aspects of software developers.

REFERENCES

- [1] K. Farias, A. Garcia, J. Whittle, J., C. Chavez, C. Lucena. “Evaluating the effort of composing design models: a controlled experiment”, *Journal on Software & Systems Modeling*, vol. 14, n. 4, pp. 1349-1365, 2015.
- [2] The Epsilon Book, <http://www.eclipse.org/epsilon/doc/book/>, Accessed in 10 March 2018.
- [3] S. Clarke. “Composition of Object-Oriented Software Design Models”, PhD Thesis. Dublin City University, 2001.
- [4] IBM Rational Software Architect, <https://www.ibm.com/developerworks/downloads/r/architect/>, accessed 10 March 2018.
- [5] M. La Rosa, M. Dumas, R. Uba, R. Dijkman. “Business Process Model Merging: An Approach to Business Process Consolidation”, *ACM Transactions on Software Engineering Methodology*, vol. 22, num. 2, pp. 1-42, 2013.
- [6] EMOTIV user manual, <http://emotiv.com/developer/SDK/UserManual.pdf>, accessed 16 March 2018.
- [7] K. Farias. “Empirical evaluation of effort on composing design models”. Ph.D. thesis, Department of Informatics, PUC-Rio, 2012.
- [8] T. Mens “A State-of-the-Art Survey on Software Merging”, *IEEE Transactions on Software Engineering*, vol. 28, num. 5, pp 449-562, 2002.
- [9] V. Basili, G. Caldiera, H. Rombach. “The Goal Question Metric Paradigm”, *Encyclopedia of Software Engineering*, vol. 2, (John Wiley and Sons, 1994), pp. 528–532.
- [10] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, A. Wesslén. “Experimentation in software engineering”, Kluwer Academic Publishers, 2012.
- [11] J. Siegmund. “Program Comprehension: Past, Present, and Future”, *IEEE 23rd Int. Conf. Software Analysis, Evolution, and Reengineering*, Suita, Japan, pp. 13-20, 2016.
- [12] M. Petre. “UML in practice”. *Int. Conf. Software Engineering*, San Francisco, USA , 2013, pp. 18–26.
- [13] “UML: Infrastructure specification, version 2.5”, <http://www.omg.org/spec/UML/2.5/PDF>, accessed 10 March 2018.
- [14] G. Scanniello et al. “Do software models based on the UML aid in source-code comprehensibility? Aggregating evidence from 12 controlled experiments.” *Empirical Software Engineering*, pp 1-39, 2018.