Analyzing the Effort on Composing Design Models in Industrial Case Studies

Kleinner Farias
OPUS Research Group, Department of Informatics,
Pontifical Catholic University of Rio de Janeiro,
Rio de Janeiro – RJ, Brazil
kfarias@inf.puc-rio.br

Categories and Subject Descriptors
D.2.2 [**Software Engineering**]: Design tools and Techniques – *object-oriented design methods, computer-aided software engineering.*

General Terms
Experimentation, Design, Documentation, Measurement

Keywords
Model Composition, UML, Empirical Studies

1. BACKGROUND
Even though model composition plays a central role on collaborative software modeling, there is little empirical knowledge about the effort on composing design models [1] [2] [3] [4] [5]. There is also a growing body of work on model composition techniques and supporting tools; however, most of the research on the interplay of effort and model composition rest on subjective assessment criteria [1].

1.1 Model Composition and Conflicts
Model composition refers to a set of activities that should be performed over two *input models*, M_A and M_B, to produce a *composed model*, M_{CM}. The latter often needs to be reviewed and changed to become complaint to an *output intended model*, M_{AB}. We use M_{CM} and M_{AB} to differentiate ($M_{CM} \neq M_{AB}$) between the output models produced by a composition technique and the model desired by developers, respectively; because the input models conflict in a some way, so usually M_{CM} is produced rather than the M_{AB}. Conflicts consist of contradicting changes assigned to model element's properties e.g., the property *isAbstract* receives conflicting values *false* and *true*. When conflicts are improperly tamed, *inconsistencies* emerge in the M_{CM} e.g. *isAbstract* = false instead of the expected value *true*, thereby impairing key model quality attributes, such as its correctness.

Copyright is held by the author/owner(s).
AOSD'11, March 21–25, 2011, Pernambuco, Brazil.
ACM 978-1-4503-0605-8/11/03.

1.2 Model Composition Effort
The effort (*time*) to produce the M_{AB} involves three variables (see Figure 1) as follows: (1) the *effort to apply* a model composition technique: $f(M_A,M_B)$; (2) the *effort to detect* undesirable inconsistencies in the M_{CM}: $diff(M_{CM},M_{AB})$; and (3) the *effort to resolve* inconsistencies: $g(M_{CM})$. Once a M_{CM} has been produced, the next step is to measure the effort to transform M_{CM} into the intended model (M_{AB}).

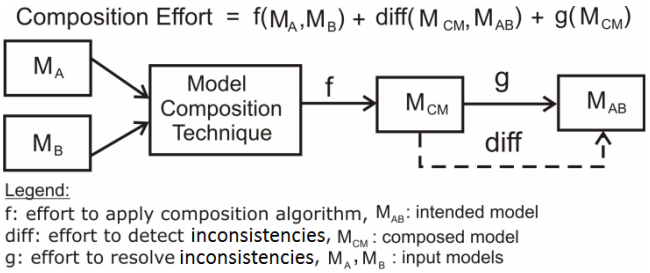

Figure 1. Overview of model composition effort: an equation.

2. RESEARCH PROBLEM AND MOTIVATION
Model composition claims to entail many potential benefits, such as facilitating the collaborative design and evolution of software models (e.g., OO and AO design models). However, nothing has been done to systematically *quantify* the effort variables from real-world settings [1]. Even worse, there is no attempt to *characterize* the factors that influence these composition effort variables and yield success or failure in real-life software projects

Composition Effort Measures. If the effort for composing models is high—i.e., M_{AB} is produced with an effort beyond the expected, so the potential gains in productivity in collaborative software modeling can be compromised. The reason is manifold. First, current model composition techniques are not lightweight and intuitive for developers to apply them properly. Second, the detection of semantic and syntactical conflicts is non-trivial. It requires an interpretation of the model semantics and a deep knowledge about the complex syntax of the modeling language, respectively. Third, conflict resolution requires that many static and dynamic aspects of a software system are understood so that conflicts can be properly grasped and tamed. Even worse, in practice, decisions whether $M_{CM} = M_{AB}$ are strictly based on ad hoc feedback from experts rather than measures and empirically-driven guidance.

Influential Factors on Collaborative Model Composition. There is a growing heterogeneity of the mainstream modeling languages and complexity of the current model composition techniques. In addition, it is even more difficult to use model composition to support collaborative software development due to some conflicting semantic issues that inevitably emerge. Therefore, it is particularly challenging for developers to systematically identify and understand the impact of these factors on the model's quality issues and effort variables, given the problem at hand. Model managers faced with such unpredictability become reluctant to authorize the model composition. Hence, this is viewed as the main impairment to model composition being more widely accepted in industrial projects where resources and time are tight.

3. APPROACH AND UNIQUENESS

Our research aims at empirically evaluating the effort on merging design models in practice. For this, five industrial case studies were performed so that practical knowledge could be generated. These studies will enable us to address two research questions:

- **RQ1:** What is the *effort* to merge design models?
- **RQ2:** What are the *factors* that impact on the composition effort?

4. CONTRIBUITIONS AND RESULTS

The contributions of this research are briefly described as follows:

- an evaluation framework for model composition effort;
- practical knowledge about the values that the composition effort variables assume in realistic composition scenarios—i.e., the quantification of the required effort on applying the (semi-)automated composition techniques, detecting conflicts and resolving the emerging conflicts;
- the identification of specific scenarios of evolution and factors that lead the composition technique to success (lower effort) or fail (higher effort).

We also summarize some interesting findings:

- the higher the number of overlapping parts of the input models is, the higher the composition effort is;
- conflict propagation between the model elements has a negative impact on the developers' effort;
- sometimes developers prefer "living with conflicts" instead of spending effort to resolve them; and
- the resolution of unexpected conflicting changes is influenced by the developers' reputation in the software development team.

5. REFERENCES

[1] France, R et. al., MDD of Complex Software: A Research Roadmap. ICSE'07, pp. 37–54, 2007.

[2] S. Nejati et al. Matching and Merging of Statecharts Specifications. In: *ICSE'07*, pp. 54–64, 2007.

[3] K. Farias et al. Assessing the Impact of Aspects on Model Composition Effort. In: *AOSD'10*, pp. 73-84, Saint Malo, France, 2010.

[4] K. Farias et al. Empirical Evaluation of Effort on Composing Design Models. In: *ICSE's Doctoral Symposium*, pp. 405-408, 2010.

[5] J. Whittle et. al., Synthesizing Hierarchical State Machines from Expressive Scenario Descriptions, ACM TOSEM, vol. 19, no. 3, 2010.