Empirical Evaluation of Effort on Composing Design Models

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

ABSTRACT
The importance of model composition in model-centric software development is recognized by researchers and practitioners. However, the lack of empirical evidence about the impact of model composition techniques on developers' effort is a key impairment for their adoption in real-world design settings. Software engineers are left without any guidance on how to properly use certain model techniques in a way that effectively reduces their development effort. This work aims to address this problem by: (1) providing empirical evidence on model composition effort through a family of experimental studies; (2) defining quantitative indicators to objectively assess key attributes of model composition effort; (3) deriving a method to support the systematic application of composition techniques; and (4) conceiving a new model composition technique to overcome the problems identified throughout the experimental evaluations.

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

General Terms
Experimentation, Design, Documentation, Measurement

Keywords
Model Composition, UML, Empirical Studies

1. INTRODUCTION
Model-driven development (MDD) aims at shifting the software development focus from code to models. For the MDD vision to become industrial reality, it is required to correctly support an essential task: *model composition* [1]. The term model composition refers to a set of activities that should be performed to combine two (or more) *input models*, M_A and M_B, in order 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} (see Figure 1). We use M_{CM} and M_{AB} to differentiate between the output models produced by a composition technique and the model desired by developers respectively. In practice, these models do not match ($M_{CM} \neq M_{AB}$) because the input models conflict in some way.

Model composition is crucial to many model-driven software development tasks—e.g., to manage a set of evolving enterprise models and reconcile models developed in parallel by different development teams. In collaborative MDD, for example, separate teams may concurrently work on a partial model of the overall software system model to allow developers to concentrate more effectively on specific parts of the model relevant to them. At some point, it is necessary to bring these parts together to generate a big picture of the overall model using some model composition technique, which can be a (semi-)automated or traditional, manual one. However, there is an effort to produce a composed model. Manual composition of input models is an error-prone and time-consuming task. On the other hand, it is not clear to what extent the use of automated model composition techniques [6] [7] reduces the effort of software developers. In practice, there is always some considerable effort to either specify the input model relations or manually resolve the conflicts in the composed model. If the composition effort is high, then this means that the potential benefits of the model composition e.g., gains in productivity in evolving models, can be compromised.

Before adopting any composition technique in realistic settings of MDD, it is necessary to have actual evidence about the effort related to the use of the technique. This need has been the key point of improvement required in enterprise modeling tools that support model composition, such as IBM Rational Software Architecture (RSA). However, there is little reported empirical evaluation on the actual composition effort required to produce the intended models. This is viewed as the main impairment to apply composition in real-world MDD projects where resources and time are tight.

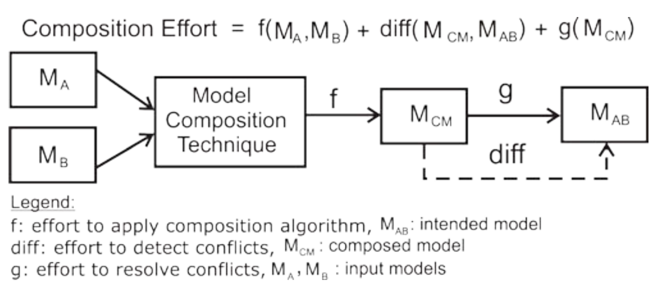

Figure 1. Overview of model composition effort: an equation.

Typically, the composition effort becomes higher than expected because different, non-trivial activities should be performed before the intended model can be delivered, including (Figure 1): (1) to select and apply the model composition technique; (2) to detect a wide range of composition anomalies such as syntactic, semantic and design conflicts; and (3) to resolve the identified conflicts. Unfortunately, both commercial (e.g. IBM RSA and Borland Together) and academic model composition techniques (e.g. Kompose, MATA and Theme/UML [5]) only ameliorate these problems to some extent. Each technique has a set of predefined *composition algorithms* that specify how the input models are manipulated and how composition conflicts are tamed. Moreover, such techniques can be classified in two categories based on the way model compositions are specified: full-fledged composition descriptions or heuristic-based compositions. In the first case, the developers explicitly specify all the relations and compositions between model elements. In the second case, the input models are merged based on heuristics, which match input model elements by "guessing" the semantics of the input elements.

This PhD research is aimed at empirically investigating the impact of different composition techniques on model composition effort. For this, a family of empirical studies is performed so that practical knowledge can be generated. The main goals of the research are: (1) to gather empirical evidence about the model composition effort based on the use of heuristics or full-fledged composition descriptions; (2) to define quantitative indicators to assess relevant attributes of model composition effort; (3) to derive an understandable workflow to assist the disciplined use of model composition techniques; and (4) to propose a new model composition technique to overcome the problems identified throughout the research.

2. PROBLEM DESCRIPTION AND LIMITATIONS OF RELATED WORK

According to [1], the state of the practice in assessing model composition provides evidence that composition is still in the craftsmanship era. In fact, there is little empirical evidence that evaluates effort estimation on model composition in order to justify (or not) their use in industrial projects. As a result, there exists no scientific foundation about the obtained enhancement of software productivity neither about the nature of side-effects. Moreover, there are limited insights on the effort required to apply composition algorithms, detect and resolve conflicts. This is particularly important as there is not much evidence if using the (semi-)automated composition technique outperforms traditional, manual model compositions in terms of effort. In the following, we describe the specific problems (PD) to be addressed in our research.

PD1. Although model composition claims many potential benefits, it has not been largely adopted in industrial projects. The main reason is the lack of practical knowledge about the trade-off between the composition effort and the feasible benefits. In practice, high effort is likely due to some problems. First, current techniques are not applied easily because they are not lightweight and intuitive for developers. Secondly, the detection of semantic and syntactical conflicts is a non-trivial activity, because it requires an interpretation of the models and a deep knowledge about the syntax of the modeling language, respectively. Equally, design conflict requires a high effort to detect when design rules (e.g. OCL-based design constraints) are challenged. Third, resolution of conflicts requires that many aspects (e.g. static and dynamic) of a software system are understood so that conflicts can be properly tamed. To analyze the trade-off implies a decision to be made with full comprehension of both upsides and downsides of a particular composition technique. However, the lack of understanding about how to assess such problems represents the main impairment to verify the composition claims.

PD2. Models are used for describing solutions, code generation and as means for communication between developers. However, given the growing heterogeneity of model composition techniques and complexity of the current modeling language e.g., UML and its profiles, it is particularly challenging for developers to objectively assess the impact of composition techniques on the quality of the models without a set of quantitative indicators. The main difficult is to address unexpected conflicts as well as quantify the effect of their propagation. Conflicts are flaws that impair model quality attributes such as correctness, consistency, non-redundancy, stability and completeness. Consequently, such conflicts are likely to affect the use of the models. Even worse, in practice, decisions whether (or not) $M_{CM} = M_{AB}$ are based mainly on feedback from experts (which determine "goodness" of the composed models) rather than empirical data; and these multiple sources of feedback often diverge.

3. RESEARCH QUESTION AND HYPOTHESES

The central question in our research is to understand how composition effort is affected by the application of particular composition algorithm. This point can be, in turn, decomposed into three specific research questions regarding the three assessment points (*f, g, diff*) defined in our composition effort equation (Figure 1).

- **RQ1:** What is the relative effort to apply the current (semi-)automated composition techniques with respect to the traditional, manual composition of models?
- **RQ2:** What is the effect of composition conflicts in models in particular with respect to misinterpretation and effort to detection?
- **RQ3:** Is the effort in resolving the conflicts greater than the effort saved by automation?

Our work is based on empirical studies, which aim to provide evidence in order to answer these questions and support or refute three hypotheses discussed in the following subsections.

3.1 H1: Composition Application Effort

The null hypothesis assumes that the effort in using the automated and traditional, manual composition is essentially the same or lower (RQ1). Based on the composition effort equation (Figure 1), the alternative hypothesis states that the effort in applying, detecting, and resolving the composition conflicts requires a higher composition effort.

Null Hypothesis 1, H_{1-0}: There is no difference in effort to produce the composed model applying the (semi-)automated model composition techniques and the traditional, manual one, or automated techniques lead to lower effort.

H_{1-0}: $f(M_A,M_B)_{Automated} \leq f(M_A,M_B)_{Manual}$

Alternative Hypothesis 1, H_{1-1}: The effort in applying the (semi-)automated model composition techniques is greater than the traditional, manual ones.

H_{1-1}: $f(M_A,M_B)_{Automated} > f(M_A,M_B)_{Manual}$

3.2 H2: Conflict Detection Effort

The effort to detect conflicts is directly related to the manner in which model elements are organized. For instance, it is expected that the more modularized the input models are, the lesser is number of conflicts observed. Therefore, we are concerned with investigating how certain model decompositions, such as aspect-oriented modeling (AOM) [11], lead to lower or higher effort to detect conflicts in output models. We are particularly interested in AOM because it aims to facilitate the composition of models and their crosscutting effects. It is important to point out that model composition conflicts have a tendency to propagate in a composed model. That is, the introduction of one conflict can often lead to multiple other conflicts as a result of a "knock-on" effect. An example would be the conflict whereby a composed model is missing an important operation. This semantic conflict leads to a "knock-on" syntactic conflict if another model requires the operation. In the worst case, there may be long chains of conflicts all derived from a single conflict.

Studying the propagation effects is crucial because propagation directly affects the effort measure in detecting conflicts – e.g., a propagation chain of length *n* may (or not) be fixed by resolving a single conflict rather than the expected *n* conflicts. Thus, we are interested, for instance, in understanding the possible conflict propagation patterns in aspect-oriented (AO) and non-AO models and its impact on the effort on conflict detection. This leads to the second null hypothesis and alternative hypothesis as follows. The second null hypothesis assumes that the effort in detecting conflicts in AO and non-AO models is essentially the same or lower (RQ2). The alternative hypothesis states that the effort to detect conflicts in AO models is higher than in non-AO models.

Null Hypothesis 2, $H_{2\text{-}0}$: There is no difference between the effort to detect conflicts in AO and non-AO models.
$H_{2\text{-}0}$: diff($M_{CM\text{-}AO}$,$M_{AB\text{-}AO}$) ≤ diff($M_{CM\text{-}non\text{-}AO}$,$M_{AB\text{-}non\text{-}AO}$).

Alternative Hypothesis 2, $H_{2\text{-}1}$: Aspect-oriented modeling leads to a higher effort to detect composition conflicts in AO models than non-AO models.
$H_{2\text{-}1}$: diff($M_{CM\text{-}AO}$,$M_{AB\text{-}AO}$) > diff($M_{CM\text{-}non\text{-}AO}$,$M_{AB\text{-}non\text{-}AO}$).

3.3 H3: Conflict Resolution Effort

If $f(M_A,M_B)_{Automated}$ produces M_{CM} with conflicts, then it is required to check if the effort to resolve such conflicts is not higher than the effort to produce the intended model manually ($f(M_A,M_B)_{Manual}$). In this context, the third null hypothesis assumes that the effort in resolving conflicts is lower or equal to effort saved by the automation. The alternative hypothesis assumes that the effort to resolve the conflicts exceeds the effort saved.

Null Hypothesis 3, $H_{3\text{-}0}$: There is no difference in the effort in resolving the conflict ($g(M_{CM})$) and the effort saved by automation ($f(M_A,M_B)_{Automated}$).
$H_{3\text{-}0}$: $g(M_{CM}) \le f(M_A,M_B)_{Manual} - f(M_A,M_B)_{Automated}$

Alternative Hypothesis 4, $H_{4\text{-}1}$: The effort to resolve the conflicts in the composed model is greater than the effort saved by the automation.
$H_{3\text{-}1}$: $g(M_{CM}) > f(M_A,M_B)_{Manual} - f(M_A,M_B)_{Automated}$

4. RESEARCH METHOD AND PROGRESS

According to Wohlin [3], empirical studies provides a controlled way of evaluating activities that have people (e.g. designers and researches) and techniques (e.g. model composition techniques) heavily related. The focus of our research is to provide empirical evidence about the model composition effort based on real-world software development settings. Therefore, we will exploit a combination of empirical research methods – surveys, case studies and experiments – in the context of industrial software projects.

4.1 Definition of Quantitative Indicators

As a first step, an exploratory study was already performed in order to capture an initial set of quantitative indicators for assessing and comparing composition algorithms in two case studies [2]. Thus, a suite of metrics was defined to quantify syntactic, semantic and design conflicts arising at composed models. These models were produced with well-known heuristic techniques for model composition, such as override, merge and union [5]. The initial evaluation has demonstrated the feasibility and efficacy of the quantitative indicators to pinpoint whether a model composition algorithm was (or not) properly chosen. The metrics were used to quantify design conflicts, such as modularity anomalies, and different types of conflict rates during the composition process. For example, in some cases, the composed models exhibited several forms of non-obvious undesirable conflicts and anti-modularity factors. However, obviously more investigations on its applicability to large and more realistic models and different model decompositions were required.

4.2 Impact of Aspect Models on Effort

The second step aimed at investigating whether (or not) aspect-orientation reduces conflict resolution effort when compared with non-aspect oriented models as improved modularization may better localize conflicts (H2) [4]. Thus, a quasi-experiment was conducted to analyze the impact of aspects on conflict resolution, consequently, on composition effort as a whole. In particular, a total of 60 compositions were performed to express the evolution of six releases of a real-world software product line, namely Mobile Media [9], developed in two versions: aspect-oriented and object-oriented models. The releases of the Mobile Media's models were explicitly represented by a sequence of composition relationships. In this case, the three composition algorithms, such as *merge*, *override* and *union,* were applied to precisely accommodate the changes into the based model. The composed models produced were compared against each other (both AO and non-AO models) in terms of their conflict rate and effort to resolve the identified conflicts. The findings identified specific scenarios where aspect-orientation properties, such as obliviousness and quantification, results in a lower composition effort. For example, aspect models with higher quantification contribute to higher conflict rates in AO models.

4.3 Relation between Stability and Effort

Understanding the impact of composition algorithms i.e., *override, merge* and *union,* on the model stability is important for measuring the effort and for the good quality models. A model is stable if, when observed over the input models and the output model, the differences in the measures of the model characteristics (e.g. coupling and cohesion) are considered small [10]. Thus, the main concern was to systematically identify and analyze which factors affect positively and negatively the production of stable output models, when the algorithms are applied. However, conflicts have been identified as detrimental to stability of the models; therefore, provide a negative impact on the composition effort i.e. they increase the effort to produce the intended model (H2).

The model characteristics were collected through the suite of metrics defined in the first study. The algorithms were used to evolve AO and non-AO models of the along of six releases of the product line under study in similar way to the previous study. The research method used was the quasi-experiment; in which the independent variable considered was the composition algorithms, while the dependent variable was the stability of the model in question. The collected empirical data disclosed a heavy reliance on stability and composition effort: that is, instable models required a greater effort to reach the intended model because they have, for example, a higher amount of composition conflicts. On the other hand, stable models required a lower effort to reach the intended model. Hence, this study has shown how the stability measure can be used for identifying the better composition scenarios to use the algorithms and for quality notions for specific purpose of composition. Additionally, the stability measure has shown to be potentially useful as a predictor for the effort estimation to overcome composition conflicts.

4.4 Initial Proposed Approach

There were several interesting observations from these previous case studies and quasi-experiment. The most significant was the strong dependence of model composition effort on the following factors: (1) the type of evolution, model naming policy, semantic (meaning) matching between model elements; (2) the level of details of the models considered during the matching process; and (3) modeling approach. These factors exerted higher influenced on the composition effort than the specific composition algorithms themselves. Thus, an innovative, flexible model comparison approach was proposed based on matching algorithms [8]. The proposed approach was fully implemented as an Eclipse plug-in by a match operator that combines syntactical matching rule, synonym dictionary and typographic similarity technique to a semantic, ontology-based technique. Ontologies are semantically richer, have greater power of expression than UML models and can be formally verified for consistency, thus providing more reliability and accuracy to model comparison, taming the semantic matching problem. Moreover, a workflow was created to provide users with guidance that facilitates the inclusion of new matching algorithms.

4.5 Effort to Compose Industrial Models

The next step will concentrate on designing and carrying out two experiments to assess the composition effort of industrial UML-based models in industrial software projects. On the one hand, these experiments will help to refute or confirm the hypotheses defined in Section 3. In this sense, professional experiments are supposed to validate portions of the hypotheses, in particular those that should be validated in industry projects. On the other hand, after the analysis and interpretation of the collected data, the initial model composition technique will be improved. The plan is to have an initial model composition tool built in the end of 2010. The tool will be built on top of Eclipse Platform using the GMF, EMF, UML2, and GEF plug-ins. Finally, a controlled experiment will put in practice to both assess both the tool and the proposed technique.

5. CONTRIBUITIONS AND VALIDATION

The expected contributions of this research are: (1) an evaluation framework for model composition effort; (2) empirical evidence whether: the effort in applying the (semi-)automated composition techniques is greater than the traditional, manual composition techniques; AO modeling leads to a higher effort to detect composition conflicts in AO models than non-AO models; the effort in resolving the conflicts in the composed model is greater than the effort saved by the automation. Moreover, practical knowledge about the effect of composition conflict on the misinterpretation of the models. And from the lesson learned will be defined a new model composition technique.

The intent is that this technique can be useful to evolve industrial models with lower effort in the daily work of developers. For this, a controlled experiment will be performed to generate data to more precisely test the hypotheses defined previously. We plan to have a group of developers that compose industrial models manually and another group using the proposed tool to test our hypotheses. Moreover, observing the usage of the tool, gathering feedback through interviews and analyzing the collected data from the evaluation framework will produce more accurate information about the composition effort. In the interviews, we also plan to gather information about how the conflicts are perceived by developers. Finally, we should point out that model composition is in initial stage and its empirical-driven improvement is necessary to the evolution of MDD field.

6. REFERENCES

[1] France, R. and Rumpe, B. Model-Driven Development of Complex Software: A Research Roadmap. In Future of Software Engineering at ICSE'07, pp. 37–54, 2007.

[2] Oliveira, K., Garcia, A. and Whittle, J. On the Quantitative Assessment of Class Model Compositions: An Exploratory Study. In: 1th ESMDE at MODELS, 2008.

[3] Wohlin, C., et. al. A. Experimentation in Software Engineering: an Introduction. Kluver Academic Pub., 2000.

[4] Farias, K., Garcia, A. and Whittle, J. Assessing the Impact of Aspects on Model Composition Effort. In: 9th Int. Conf. AOSD'10, Saint Mello, France 2009. (to appear)

[5] Clarke, S. Composition of object-oriented software design models, Ph.D. Thesis, Dublin City Univ., January, 2001.

[6] Reddy, R., et. al. Directives for Composing Aspect-Oriented Design Class Models, TAOSD, vol. 1, 2008.

[7] Cottenier, T., Berg, A. and Elrad, T. Joinpoint Inference from Behavioral Specification to Implementation. In: 21st ECOOP, pages 476–500, Berlin, Germany, July, 2007.

[8] Oliveira, K. Breitman, K. and Oliveira, T. A Flexible Strategy-Based Model Comparison Approach: Bridging the Syntactic and Semantic Gap. JUCS, vol. 15, no. 11, 2009.

[9] Figueiredo, E. et. al, Evolving Software Product Lines with Aspects: An empirical Study on Design Stability, In: 30th ICSE'08, pages 261–270, Leipzig, Germany, 2008.

[10] Kelly, D. A Study of Design Characteristics in Evolving Software Using Stability as a Criterion. In IEEE TSE, vol.32, no. 5, pages 315–329, May 2006.

[11] Clarke, S. et. al. Generic Aspect-Oriented Design with Theme/UML, AOSD, pages 425–458, Addison-Wesley, Boston, 2005.