Analyzing the Effects of Aspect Properties on Model Composition Effort: A Replicated Study

Everton Guimarães, Alessandro Garcia, Kleinner Farias
Opus Research Group – LES, Department of Informatics
Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rio de Janeiro, Brazil
{eguimaraes, afgarcia, kfarias}@inf.puc-rio.br

ABSTRACT

Nowadays, model composition plays a central role on software engineering activities. For example, reconciling design models developed in parallel by different software development teams. It can be defined as a set of activities that should be performed over two input models, M_A and M_B, to produce an output composed model, M_{CM}. Usually, the problem is that the M_{CM} does not correspond to the output intended model. It can be explained by the fact that M_A and M_B conflict in some way, and the emerging conflicts must be detected and solved. However, these two activities tend to be critical as model composition is usually a cumbersome manual task. This paper presents the replication of an exploratory evaluation about the impact of aspect-orientation properties, such as quantification and obliviousness, on conflict resolution effort. The intuition is that improved modularity of aspect-oriented models may help to better localize composition conflicts. We investigate this issue through an application different from the one used in the previous study, in which three well-established composition algorithms (i.e., *override*, *merge* and *union*) were used to evolve the design models. The goal was to analyze how the impact of varying quantification and obliviousness degrees are correlated with the rate of semantic and design conflicts. We identified some cases where quantification and obliviousness reduce conflict rate and the composition effort.

General Terms

Measurement, Design, Experimentation, Aspect-Orientation

Keywords

Assessment, Aspect-Oriented Modeling, Model Composition, Software Architecture, Empirical Studies.

1. INTRODUCTION

Nowadays there are an increasing number of researchers who have focusing on defining and improving model composition techniques. This can be related to the fact that model composition is playing a main role on model-driven software engineering. According to [3], model composition can be viewed as an operation where a set of activities should be performed over two input models Ma (base model) and Mb (delta model) in order to produce a composed model (Mab). Model composition techniques [14,] [12] [13] are often used, for instance, to change or evolve existing models. In the context of aspect-oriented modeling [1], a common composition scenario could be the merge of the base model and the aspectual model.

It is well known that model composition is a highly intensive manual task [2]. Once model composition is carried out, some conflicts in the output models can arise. Thus, these conflicts must be detected and resolved in order produce a correctly composed model. However, resolving conflicts can be a very difficult task since its resolution depends on the meaning of the models and the semantic information associated with the model, normally, is not available. Unfortunately, there is limited knowledge on which model properties contribute to reduce conflicts and their resolution effort.

In the context of AOM, it is even worse: modelers are left without any evidence about the impact of basic aspect properties, such as obliviousness and quantification [15], on model composition. Based on the results of our previous study [16], there is a working hypothesis that aspect-orientation may alleviate the effort on conflict resolution. The previous study identified that aspect orientation can: (i) better localize conflicts in the presence of full obliviousness, (ii) increase the rate of conflicts when aspects have a high quantification degree, and (iii) even in the second case though, conflict resolution effort tended to be lower in aspect-oriented modeling. However, the original study was limited to analyze model compositions in a single application, which make it difficult to gather more general conclusions.

Therefore, the aim of this paper is to report the replication of the first study [16], which aimed to produce initial evidence about the interplay of AOM properties and model composition conflicts. Our study replication relied on a Web-based system, differing from the application domain of the first study: a product line for mobile devices. Our intention was to confirm or refute on the finding that higher conflict rates are observed in the presence of aspects with higher degree of quantification. Moreover, we tried to check in different settings whether higher degree of obliviousness tend, in fact, to yield compositions of AO composed models that are closer to the intended compositions.

The remainder of this paper is organized as follows. In section 2, we introduce key concepts related to model composition and AOM that are relevant for this paper. In section 3, we present our key evaluation procedures, describing the working hypotheses. In section 4, we present the evaluation based on the new application case. In section 5, we compare our study with related work and present some threats to validity. Finally, in section 6, we present some final remarks and future works.

2. BACKGROUND

Model composition has been studied for many years in software engineering [2] and in other related disciplines. This section presents the underlying concepts to understand our study. Section 2.1 defines model composition more precisely. Our study will compare model composition in AO and non-AO modeling settings. Section 2.2 presents the AOM language used in our study.

2.1 Model Composition

The term model composition is used to define a set of activities that should be accomplished to combine two (or more) input

models, Ma and Mb, in order to produce an output composite model, called Mab. In this paper, we use the terms *intended model* (Mab) and resulting model (Mcm) to differentiate between the composition the developer desires, and the composition produced by an algorithm. Usually, Mab is different from Mcm, because the input models conflict in some way. Besides this, we assume two hypothetical input models, named Ma and Mb. So, two elements from Ma and Mb respectively are *corresponding* whether they have been identified as equivalent in the matching process.

A model composition algorithm defines the semantics of the model composition relationship and specifies how the input models should be manipulated in order to compose them. In our work, we have used three composition algorithms: (i) override, where for all pairs of corresponding elements in both models, Ma's elements should override Mb's corresponding elements; (ii) merge, where the corresponding elements of both models are combined; and (iii) union, where for each pair of corresponding elements respectively in Ma and Mb, they should be both present in the output model in order to preserve their distinguished identification. When using the algorithms, we take into consideration that composition is applied in the direction from Ma to Mb. In addition, the elements in the input models that are not involved at the correspondence match remain unchanged and are inserted into the output model.

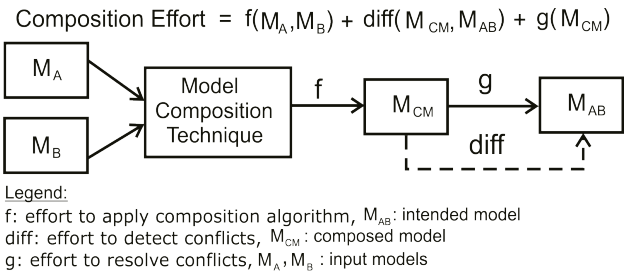

Figure 1 - Model Composition Effort

Once we apply a composition algorithm in two input models Ma and Mb, some conflicts can arise. In this sense, we need to assess the effort to solve these conflicts. In our study, the composition effort can be calculated according to the equation showed in Figure 1. The equation shows that the model composition effort is composed by: (i) the effort to apply a model composition algorithm f(Ma, Mb); (ii) the effort to detect undesirable conflicts in the output model diff(Mcm, Mab); and (iii) the effort to solve conflicts g(Mcm). Once a composed model (Mcm) has been produced, the next step is to measure the effort to transform Mcm into the intended model (Mab). If Mcm is equal to Mab, then the diff (Mcm, Mab) = 0 and g(Mcm) = 0. Otherwise, the diff (Mcm, Mab) > 0 and g(Mcm) > 0.

We know that in an input model when the composed model does not match the intended model, it means that some conflicts can arise. In practice, we can briefly identify two broad categories of conflicts: (i) syntactic conflicts, which arise when the composition algorithm results in a model not conforming to the modeling language metamodel; and (ii) semantic conflicts, where the meaning of the composed model does not match that of the intended model.

2.2 Aspect-Oriented Modeling

In this paper we will analyze and compare the model composition effort using AO and non-AO models. The use of AO models is due to the fact that AOM languages aim at improving modularity and, therefore, might reduce the burden of model composition tasks. The goal of AOM is to provide software developers with means to express aspects and crosscutting relationships in their models. Currently there are AOM languages for modeling aspects at many levels of abstraction, such as architectural design and detailed design. For instance, the study realized in [3] shown that a variety of approaches, as *Composition Patterns* [4], AODM [5], Theme/UML [6] [9] have been proposed in order to represent and specify aspect-oriented design and software architecture. The AOM language used in our study is depicted in Figure 2. This was chosen because it has been widely used in other works [7] [8] and, more importantly, in our original empirical study [16], which we have replicated. The notation proposed by this AOM language supports the visual symmetric representation of aspect-oriented software architecture. This language is an extension of UML for component models. In this sense, some properties of model elements are considered: interface, operation, attribute, component, relationship, crosscutting relationship and join points.

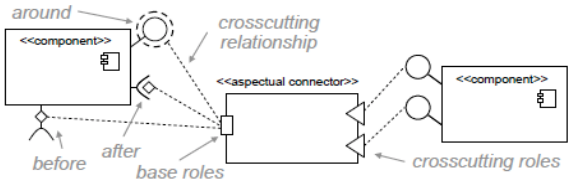

Figure 2 - Architectural Notation Description

This AOM language also provides notation that allow express different forms of aspect-component collaborations. The aspectual connectors represent collaborations. An *aspectual connector* is represented by rectangles (see Figure 2) and it defines what component interfaces, components and operations that are affected by the aspect. Moreover, the *aspectual connectors* are associated with the crosscutting relationships represented by dashed arrows. It also provides means to represent advices of aspects introducing a diamond on the interfaces (before and after advices), or even a dashed circle, in the case of *around* advice.
According to [18], *obliviousness* and *quantification* have been proposed as necessary properties for aspect-oriented programming (AOP). In other words, these properties were proposed as the core characteristics of AOP, and they are often used to define whether a language is aspect-oriented or not [16]. They can also be considered when characterizing AOM languages.

O*bliviousness* [18][15] can be defined as the act or effect of base components being oblivious with respect to aspectual components. Obliviousness implies that component interfaces and services do not have to be specifically prepared to receive the enhancements provided by aspects. *Quantification* [15][16] is defined as the ability to write unitary and separated statements that enhance many non-local scattered places in a system. When the quantification property holds, it follows that aspects may crosscut an arbitrary number of component interfaces simultaneously.

3. EXPERIMENTAL PROCEDURES

3.1 Hypothesis

Even though aspect-oriented modeling has gained more attention over this decade, there is currently very limited knowledge as to how aspects, when incorporated in input models, affect the model composition effort. More specifically, there is no knowledge

about: (i) what extent the composition of aspect-oriented models affects the emergence of conflicts in the output composed models and (ii) whether and how aspect-orientation properties, such as obliviousness and quantification (Section 2.2), may influence model composition conflicts. In this context, the hypothesis of this study can be derived from three key research questions:

- RQ1: Does the composition of AO models produce higher rate of conflicts than non-AO models?
- RQ2: What is the impact of AOM on the way conflicts propagate in the output model?
- RQ3: Does the degree of obliviousness in the input models can help to avoid conflicts?

For each research question above, we considered two hypotheses: (i) the null hypothesis, and (ii) the alternative hypothesis. Each one will be described in the following. The null hypothesis of RQ1 assumes that the composition effort for combining either AO or non-AO models is the same. In other others, in both cases they entail similar conflict rates. By the fact that aspects may crosscut many elements in a model (and based on results of our original study), the alternative hypothesis states that AO models may lead to more conflicts than non-AO models. Moreover, conflicts can be propagated in the composed models. In other words, when a conflict is introduced on model composition, other conflicts can arise over the spread of conflicts. The investigation of conflict propagation is important because it can directly affect the effort in resolving conflicts. As the previous hypothesis, the second null hypothesis assumes that conflicts equally spread through output AO and non-AO models. The third hypothesis can be viewed as a refinement of the second hypothesis, but with a deeper analysis around the aspect orientation properties. We will investigate whether AOM properties, i.e. the higher degree of obliviousness, can be related with arising conflicts or ameliorating them on AO models.

Table 1 - Summary of the proposed hypothesis

Hypothesis		Description
H1	Null	H.1-0: The CR of AO models and non-AO models are the same or similar. CR(AO) = CR(OO)
	Alternative	H.1-1: AOM leads to a lower CR than non-AO modeling. CR(AO) < CR(OO)
H2	Null	H.2-0: There is no difference in the way composition conflicts are propagated in non-AO and AO models: Prop (AO) = Prop (non-AO)
	Alternative	H.2-1: The use of aspects leads to a higher propagation of composition conflicts: Prop (AO) > Prop (non-AO)
H3	Null	H.3-0: aspect with higher the degree of obliviousness does not avoid conflict.
	Alternative	H.3-1: aspect with higher degree of obliviousness avoids conflicts on model composition.

Table 1 summarizes our hypotheses. In order to test these three sets of hypotheses, we are replicating our first empirical study [16] that was developed using the Mobile Media product line. The goal is to confirm or refute our initial findings. In this replicated study, we are using now the Health Watcher system that contains different types of aspects, such as design patterns, exception handling, concurrency control, distribution and persistence. Some of these aspects are introduced throughout new releases. Once we obtained the data of the two systems, we will also perform a comparison between the results found in both studies. In our previous study [21], we found that quantification influenced the apperance of conflicts. The presence of aspects with lower quantification (in the input models) led to fewer syntactic and semantic conflicts in the output models. We also observed that when a conflict arises in aspects with higher quantification, higher rates of syntactic and semantic conflicts occurred in the output models. Therefore, aspects with higher quantification degree can lead to higher conflicts. We also found that higher degree of obliviousness tended to produce as result a composed model that is nearest to the intended model.

4. Evaluation Procedures

This section presents the way that we quantify conflict resolution (Section 4.1) and certain evaluation procedures (Section 4.2). As previously mentioned, the target case used in our study is the Health Watcher system [10]. Figure 3 presents an illustrative example of an architectural aspect and its crosscutting effect over two interfaces of a base component. The diagram follows the architectural notation that we have previously described. The figure shows how the ServletCommanding aspectual component is bound to the base component HWServlet. The ServletCommanding aspect has a *before* advice which affects the HWServlet provided interfaces *doGet ()* and *doPost* ().

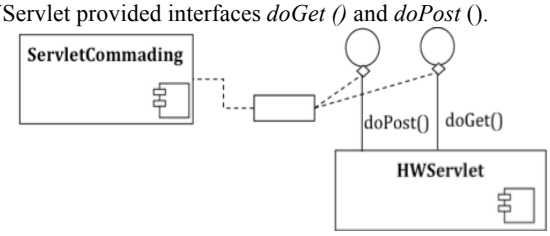

Figure 3 - Example of an aspect in the Health Watcher component model

4.1 Quantifying Conflict Resolution

In our study, we want to quantify the number of conflicts that can arise in a model composition, as well the number of activities that have to be done in the output model in order to make it reach the intended model. Hence, our analysis first relies on quantifying the amount of composition conflicts in the output models. For this measure, we have an associated metric called *conflict rate* (CR). The CR allows us to measure the density of composition conflicts in the output models. Moreover, the CR metric is used to assess the differences between the AO and non-AO models. It is important to point out that CR is defined from multiple conflict metrics, which can be found in [3].

In order to quantify number of operations to transform the composed model in the intended model, we use other metric, called *recovery effort* (RE). The RE calculated the operations of creation, delete, and update needed to produce the intended model. Once the RE data is collected, the output model is checked to verify if there is any occurrence of conflict propagation. The RE metric enables us to check if the presence of aspects in the input models has any impact on the way composition conflicts is propagated. Both metrics were applied to the output (composed) model in order to analyze if there is composition conflicts as well if they propagate in each Health Watcher release. The metrics collected allow us to assess whether the output model has (or not) conflicts after a composition algorithm is applied. The values for

these metrics showed in section 5 are related to 3 releases of the Health Watcher system (R1, R5, and R9).

4.2 Evaluation Procedures

In order to define the target model versions and releases, we have selected both non-AO and AO versions of the Health Watcher system. These two model versions of the same system enabled us to identify if the presence of aspects in the input models had positive or negative effects on the quality of the output model. To select the target model versions and releases, to activities were made: (i) deriving AO and non-AO models releases; and (ii) model releases and composition specifications. These activities are briefly described in the following.

4.2.1 Deriving AO and non-AO Model Releases

For each release of Health Watcher, we have analyzed the code to make it sure that the existing models or correctly abstracting its architecture. Once the architecture model was checked, we applied the composition algorithms that we presented before. The goal of applying different composition algorithms was to identify the conflicts and its propagation in both AO and non-AO versions. The AO models were represented with the AOM notation presented before, while the non-AO models were represented with conventional UML component models. The notation used in this work to express the architectural models has been also used in [5, 20], which was important to contrast our results with the original study [16].

4.2.2 Model Releases and Composition Specification

In our study, we considered 3 releases of the Health Watcher. Thus, after analyzing the code we derived the architecture for each selected release using model compositions. These releases were selected because they were the ones where the changes implied visible modifications in the architectural design. For each new release, the previous release was modified in order to accommodate the features to be modified, inserted or deleted. In order to evolve each release, some activities like delete, insert, and update the entities present in the previous release need to be performed. After AO and non-AO model releases has been derived and the composition been specified, we need to execute the model composition and assess possible arising model composition conflicts. Thus, we need to follow some additional activities, which are described in the next subsections.

4.2.3 Composition and Measurement Phase

The architectural models of Health Watcher were reviewed according to the existing code. There was also a need to make it sure that the AO and non-AO models (for all the releases) were well aligned, i.e. describing the same functionalities. Comparing the differences from on release to another, the differences were identified and gathered into the input model delta (Δ). From one release to another, 6 compositions were produced: 3 compositions following override, merge, and union from the current release to Δ, and 3 compositions in the inverse direction. The composition algorithms were used to generate the evolved models, so that we could assess the impact of aspects on the model composition effort. Then, we have assessed the conflict rate and the effort to resolve the conflicts using the metrics described in Section 4.1.

5. COMPOSITION EFFORT ANALYSIS

This section presents the results of applying the conflict resolution metrics to both AO and non-AO output composed models. Figure 4 illustrates the results for the CR metric obtained following both the override and merge algorithms. Each pair of bars shows CR measures for two specific compositions: (i) R1R5 – represents the composition that derived the release R5 from release R1, and (ii) R5R9 -- represents the composition that derived the release R9 from release R5. The results of the other compositions followed similar trends. The first pair of bars always represents the results for AO models, while the second one presents the results for non-AO models.

Conflict Rate: AO vs. Non-AO Models. The first observation allows us to conclude that the conflict rate measures have favored aspect-orientation in both merge and override cases. The presence of aspects in the input models produced lower conflict rate than aspect-free models when the override algorithm was applied. The inferiority of AO models was around 10% in most of the cases. For instance, the superiority in the first composition (R1R5) was: (i) around 10% for the override case (0.50 against 0,44), and (ii) almost no difference for the merge case (0.40625 against 0,391304). This means that the superiority of the AO decomposition in better containing conflict emergence and propagation is approximately 10% in the first model composition. The same reasoning applies for the merge case in the second composition (R5R9). The only exception was the override case in the second composition: the conflict rate decreases from 0.5098 (non-AO version) to 0.2931 (AO version) on the composition that derived the model of release 9 (represented by R5R9 in the histogram). As we explained above, considering four model compositions, the CR was lower in the presence of aspects in almost all cases. With this observation, we can conclude that the null hypothesis **H1-0** is refuted and the alternative hypothesis **H1-1** is confirmed. From this perspective, we can also state that this result enforces the findings of our first case study related to hypothesis H1-1 where the AO models of the Mobile Media SPL presented lower CR values than the non-AO models.

Varying Quantification Degree and Conflict Rate. However, it is also interesting to observe that we gathered different results in our previous study [16] when we undertook a more detailed analysis. In the original study, the trend in the superiority of AO models with respect to the CR metric was always higher than 10%, but higher than this and reaching 40% in many cases. After a systematic analysis of the aspect definitions in all the model compositions produced in both systems – which were target of the first and this study – we found that a key factor for the difference was the degree of quantification of the aspects found in those systems. While the vast majority of the aspects in the Mobile Media architecture (focus of the first study) affects a few join points (from one to three interfaces), most of the aspects in the Health Watcher architecture tend to affect more than three join points. In Mobile Media, the aspects tend to implement varying domain-specific features of the product line. In Health Watcher, the architectural aspects tend to modularize widely-scoped crosscutting concerns, such as persistence and distribution. This enables us to have more evidence that: higher the degree of aspect quantification, higher the probability of conflicts emerging in the composition.

For instance, a high number of conflicts in the output model was observed when the *ServletCommanding* aspect (Figure 3) was introduced in the release R5. The purpose of this aspect is to localize the implementation of the Command *design pattern.* This aspect has a high quantification and it affects 13 components. However, it is important to note that even though the composition

of non-AO models (R1R5) still has a higher CR in both composition algorithms (*override* and *merge*), the aspect leads to a scenario where aspects with higher quantification imply in a higher CR. This difference becomes clearer when we compare the quantification degree of the *ServletCommanding* aspect with the aspects introduced in the composition R5R9.

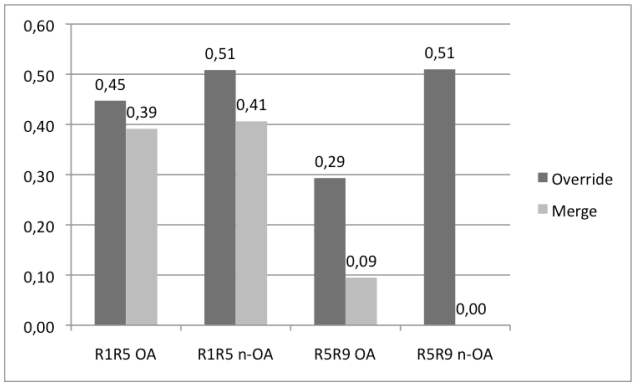

Figure 4 - CR produced y override and merge algorithms

Aspects and Conflict Propagation. The RE can be a good measure to analyze the presence of conflict propagation [16], which is related to our second hypothesis. As observed in our previous study, the higher the effort for recovering the output model, the higher the chance of conflict propagation being observed in the output model. The RE is calculated for both AO and non-AO models. Figure 5 depicts the recovery effort measures to transform the output model produced by the override and merge algorithms into the intended model.

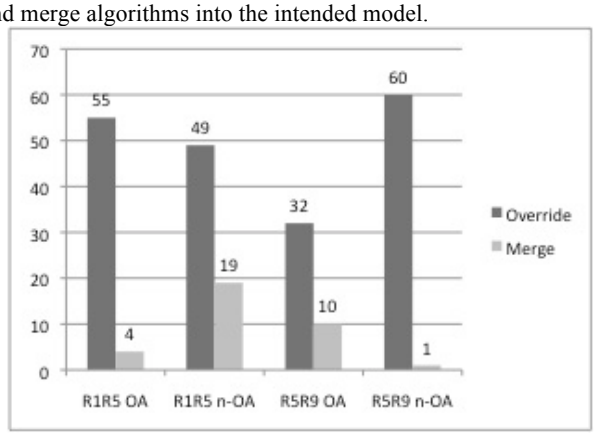

Figure 5 - Conflict resolution effort to recover the output model produced by override and merge algorithms

The analysis of Figure 5 allows us to conclude that aspects affect the way conflicts flow troughout the output models. We could identify in some cases that certain conflict propagation patterns are recurring. This happens in both AO and non-AO models. However, the ocurrence of conflict propagation is lower in AO models. There is a sensible difference on the way composition conflicts are propagated in non-AO and AO models. In general, the 'aspect-specific conflict propagation' is often caused by a conflict arising in a single aspect and spreading through affected elements in the base model. In some cases, cyclic conflict propagation occurs. This happens, for instance, when aspects share the same join points. Thus, the hypothesis **H2-0** is refuted, and the hypothesis **H2-1** is confirmed. As a consequence, the results observed in this replicated study reinforce the findings showed in our first exploratory study [3].

Composition Conflicts and Obliviousness. Some conflicts can be related to the degree of obliviousness observed when producing the input models. Analyzing the composed aspect-oriented models, an interesting finding was that their composition tended to yield a lower CR when the obliviousness degree of base model elements is higher. For instance, the HWSynchronization aspectual component affects a number of base components that have from none to little awareness about the presence of this aspect. For instance, services of the affected components EmployeeRepository and ComplaintRepository did not offer any hooks for the HWSynchronization aspect. Similar to this case of the HWSyncronization aspect, all the other base components with high obliviousness tended to entail any conflict in the output model. This is also the case of components affected by the aspect ComplaintState, HWPersistence and HWErrorHandling. So, similar to the previous study, we can confirm that the aspects with higher degree of obliviousness can help the designer to avoid conflicts in model composition. However, in this replicated study, we could better analyze this phenomenon as HealthWatcher has a higher rate of components that are oblivious to aspects. The reason in the domain-specific nature of most of the Mobile Media aspects. Therefore, we can conclude that the **H3-0** hypothesis is refuted, while the **H3-1** hypothesis is confirmed.

6. RELATED WORKS

To the best of our knowledge, our results are the first to systematically investigate the relation between different types of quantification and obliviousness degrees and model composition effort. Although a wide variety of model composition techniques has been proposed in different domains, including merging of statecharts [12], composition of UML models [6] [18], aspect-oriented modeling [19], and AO composition of models [13]. None has more systematically investigated the impact of aspect properties on composition effort. As a consequence, the lack of empirical evidence hinders the understanding of side effects peculiar to such properties on model composition process as whole. Even worse, we have observed in industrial projects that developers ultimately rely on feedback from experts to determine "how good" the input models and their compositions are. According to [2], the state of the practice in assessing model quality provides evidence that modeling is still in the craftsmanship era and when we assess model composition this problem is accentuated.

In the current literature, some metrics have been purposed for supporting the evaluation of OO and AO model composition specifications. For instance, some metrics [20] are defined to quantify the effort to specific compositions between two or more requirements models, such as scaffolding and mobility. However, their metrics are targeted at evaluating the reusability and stability of explicit model composition specifications. Therefore, these metrics cannot be directly applied to our context. Besides this, some previous works investigate the effect of using UML diagram and its profiles with different purpose. In [29], Briand et. al. looked into the formality of UML models and its relation with model correctness and comprehensibility. However, none has investigated the influence of quantification and obliviousness on model composition effort. Specially, even though this study replicates the study presented in [17], they are strikingly different in many ways. For example, the size of the software product lines models considered during the experimental study is different. The

main contrast is that this study has a more specific character for focusing on aspect properties. On the other hand, Farias and colleagues have a broader character, taking into account different issues of aspect-oriented models and model composition facets.

Finally, we can highlight that the need for assessing models during a model composition process has neither been pointed out nor proposed by current model composition techniques [4] [6] [9] [11] [12] [13] [14]. For example, the UML built-in composition mechanism, namely package merge [14], does not define metrics or criteria to assess the merged UML models. Moreover, it has been found to be incomplete, ambiguous and inconsistent [20]. We therefore see this paper as a first step in a more ambitious agenda to empirically assess the effect of aspect properties on model composition issues.

7. THREATS TO VALIDITY

Our replicated study has obviously a number of threats to validity that range from internal threats to external threats. The main possible threat to the internal validity of this study concerns inconsistency in measuring the composition conflicts and conflict resolution effort. Since the metrics were calculated manually, the collected data may not always be consistent. Hence, this might lead to inconsistent results. However, we have minimized this threat by establishing guidelines, periodic reviews with developers of the architectural models, and by engaging in discussions in cases of problems.

External validity threats concerns limitations to generalize the results of the study to a broader context. The main threats are: (i) the use of single target application, and (ii) the use of specific metrics to compute the model composition effort. Obviously, more investigations involving other case studies with compositions of larger UML models are required. Moreover, the generalization of the results might be limited to component models of SPL architectures similar to the one used in this study. We observed that the number of properties and details (i.e. granularity) of the aspect-oriented model elements taken into consideration throughout the compositions affect directly the composition results. Consequently, it is necessary to observe that to generalize our findings to other types of model (e.g. behavioral models, such as state machines or interaction models) with different levels of abstraction is needed to make further investigation.

8. CONCLUSIONS AND FUTURE WORKS

This paper presents an exploratory study to assess the advantage of aspect-orientation in reducing conflict resolution effort. This study aims to reinforce some findings of our previous study [3]. Moreover, we analyze how obliviousness can be related with model composition conflicts. In this study, model composition was used to express the evolution of architectural models in the Health Watcher System [10]. The improved modularization on the AO models, tended to help us a better localize conflicts. Besides it, we have observed that two main findings of our first study were confirmed with this new case study, they are: (i) a higher degree of obliviousness between base models and aspects led to a significant decrease of conflicts when compared with the non-AO model counterparts, and (ii) aspects with higher quantification were the cause of higher conflict rates in AO models.

Moreover, we can point out other finding that the higher degree of obliviousness can lead to lower conflict rate in aspect-oriented models However, there is a need for more investigations around that in order to ensure this relation in other scenarios and also, using other UML diagrams (with different levels of abstraction); and (ii) the conflict resolution effort was similar in AO and non-AO models. Finally, this paper presented an exploratory study that aims to investigate the effect of aspect properties on conflict resolution effort. However, further empirical studies are still required to evaluate the impact of AO modeling on model composition effort.

9. REFERENCES

[1] Chitchyan, et al. 2005. Survey of Analysis and Design Approaches", Survey of AOSD-Europe Network of Excellence, May, 2005.

[2] R. France and B. Rumpe. Model-Driven Development of Complex Software: A Research Roadmap. In: FOSE'07 co-located with ICSE'07, pages 37–54, 2007.

[3] Assessing the Impact of Aspect on Model Composition Effort, http://www.inf.puc-rio.br/~kfarias/aosd10

[4] S. Clarke et. al., Composition Patterns: an approach to designing reusable aspects. In: 23rd ICSE, Canada, 2001.

[5] D. Stein et. al., Designing aspect-oriented crosscutting in UML, In: AOM'02 at AOSD, 2002.

[6] S. Clarke and E. Banaissad. Aspect-Oriented Analysis and Design: The Theme Approach. Addison-Wesley, 2005.

[7] I. Groher et. al., Aspect-Orientation: from Design to Code, In: Early Aspects - AO Requirements Engineering and Architecture Design, Lancaster, March, 2004.

[8] E. Figueiredo et al., Evolving Software Product Lines with Aspects: An Empirical Study on Design Stability, In: ICSE'08, pages 261–270, 2008.

[9] A. Garcia et al. Representing Architectural Aspects with a Symmetric Approach. In EA'09 held at AOSD'09, 2009.

[10] S. Soares et. al., Implementing Distribution and Persistence Aspects with AspectJ, In: OOPSLA'02, pp. 174-190, 2002

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

[12] Y. Reddy et al. Directives for Composing Aspect-Oriented Design Class Models, Trans. on AOSD, 1(1):75–105, 2006.

[13] T. Cottenier et. al., The Motorola WEAVR: Model Weaving in a Large Industrial Context, In AOSD'07, 2007.

[14] J. Dingel et. al., Understanding and Improving UML Package Merge. Journal of Soft. Syst. Modeling, 7(4):443–467, 2008.

[15] R. Filman and D. Friedman. Aspect-Oriented Programming is Quantification and Obliviousness. In *Int'l Workshop on Advanced Separation of Concerns at OOPSLA'00*, 2000.

[16] K. Farias et. al., Assessing the Impact of Aspects on Model Composition Effort, In: 9th AOSD'10, France, March 2010.

[17] S. Katz, Diagnosis of harmful aspects using regression verification. In: FOAL: Foundations Of Aspect-Oriented Languages, pages 1–6, March. 2004.

[18] R. Chitchyan *et. al*, Semantic vs. Syntactic Compositions in Aspect-Oriented Requirements Engineering: an Empirical Study. In: Proc. AOSD'09, Virginia, USA, March 2009.

[19] L. Briand et. al., An Experimental Investigation of Formality in UML-Based Development, IEEE TSE, 31(10):833–849, 2005.

[20] OMG. Unified Modeling Language: Infrastructure version 2.2. Object Management Group, February 2008.