On the Impact of Obliviousness and Quantification on Model Composition Effort

Everton Guimaraes
PUC-Rio
Rio de Janeiro, Brazil
eguimaraes@inf.puc-rio.br

Alessandro Garcia
PUC-Rio
Rio de Janeiro Brazil
afgarcia@inf.puc-rio.br

Kleinner Farias
PUC-Rio
Rio de Janeiro, Brazil
kfarias@inf.puc-rio.br

ABSTRACT
Researchers and practitioners advocate that design properties, such as obliviousness and quantification, can improve the modularity of software systems, thereby reducing the effort of composing design models. However, there is no empirical knowledge about how these design properties impact model composition effort. This paper, therefore, performs an empirical study to understand this impact. The main contributions are: (i) quantitative indicators to evaluate to what extent such design properties impact model composition effort; (ii) an objective evaluation of the impact of such modularity properties in 26 versions of two software projects by using statistical tests; and (iii) lessons learned on whether (and how) modularity anomalies related to misuse of quantification and obliviousness in the input models can significantly increase model composition effort.

Categories and Subject Descriptors
K.6.3 [**Software Management**]: Software Development.

General Terms
Measurement, Design, Experimentation,

Keywords
Aspect-Oriented Modeling, Model Composition, Modularity, Measurement

1. INTRODUCTION
Industry and academia recognize the importance of *model composition* in evolving Aspect-Oriented (AO) and Object-Oriented (OO) design models. Model composition can be briefly defined as a set of tasks that should be performed over two input models, M_A and M_B, to produce an output intended model, M_{AB}. For this, software developers use the composition algorithms (e.g. merge and override algorithms) to match the model elements in M_A and M_B by automatically "guessing" their semantics and then combine the corresponding elements to create M_{AB}. Nevertheless, these algorithms usually produce an output composed model, M_{CM}, that does not match with the intended model, M_{AB} The reason is that M_A and M_B often conflict with each other and these conflicts are commonly converted into inconsistencies in M_{CM}. Hence, developers should invest some considerable effort to resolve them, i.e. transforming M_{CM} into M_{AB}. As a result, compose design models is still considered a tedious, error-prone, and time-consuming task [6].

Our expectation is that the way of structuring the model elements in M_A and M_B may significantly affect the effort to accommodate the changes from M_B to M_A. Researchers and developers use aspect-oriented modeling (AOM) [1] to produce well-modularized software. The superior modularization of crosscutting concerns is essentially reached due to two crucial AO properties: *quantification* and *obliviousness*. The first property is defined as the idea that one can write unitary and separate statements that have effect in many non-local places in design modules [8]. When the quantification property holds, it follows that aspects may crosscut an arbitrary number of component interfaces simultaneously. The second property states that the design places where these quantifications were applied did not have to be specifically prepared to receive these enhancements [8]. Some studies show that systems with a superior modularity are less prone to exhibit design problems [13], and therefore, less prone to manifest conflicts during the composition process. A strong relationship can be observed between obliviousness and quantification with traditional modularity properties, such as cohesion and coupling. The problem is that little has been done to evaluate the impact of obliviousness and quantification on model composition effort. Still, a systematic description on what factors affect the developers' effort and how they ideally and practically should be evaluated is insufficiently covered in the literature.

The hypothesis of this paper is that AO models may minimize the composition effort to some extent. Our intuition is that a more effective modularization, supported by *obliviousness* and *quantification,* may better accommodate the changes, thereby reducing the model composition effort. However, it is by no means obvious that this hypothesis holds. It may be, for instance, that a high quantification in AO models has a detrimental effect on the composition effort, e.g. sometimes developers should examine all points crosscut by an *aspect* in M_A and M_B so that the conflicting changes are not converted into inconsistencies in M_{CM}. Without the understanding of the effects of obliviousness and quantification, developers may improperly use mechanisms found in AOM or even spend a high, unnecessary effort. By analyzing these properties, we can grasp to what extent they may impact on the way conflicts arise and propagate in the composed model [13].

This paper, therefore, investigates the impact of obliviousness and quantification on the model composition effort. In particular, we evaluate (1) the *conflict rate* produced during the compositions and (2) the effort required to resolve the conflicts [5][1]. We compare the conflict rate and recovery effort produced during the evolution of AO and OO models supported by model composition

SAC'14, March 24-28, 2014, Gyeongju, Korea.

algorithms. Thus, this comparative analysis allows understanding the impact of *obliviousness* and *quantification.* Our investigation is conducted in the phase of architectural design. The main contributions of our study are: (i) a set of quantitative indicators to evaluate to what extent quantification and obliviousness influence the composition effort (Section 3); (ii) empirical evidence, supported by statistical tests, about the impact of the modularity properties on model composition effort (Section 4); and (iii) lessons learned on whether (and how) modularity anomalies related to misuse of quantification and obliviousness in the input models significantly increase model composition effort.

2. BACKGROUND

Model Composition is defined as a set of activities that should be accomplished to combine two (or more) design models, M_A and M_B, to produce a composed model, M_{AB}. The M_A model represents the actual versions of a software system, while the M_B model (also refereed to as *delta model*) represents changes request that need to be performed to reach the intended model. We have also used the terms *intended model* (M_{AB}) and *composed model* (M_{CM}) to differentiate, respectively, between the composition intended by developers and the composition produced by a composition algorithm. A model *composition algorithm* defines the semantics of the model composition relationship and specifies how the input models should be manipulated to compose them. Our work focuses on two well-established composition algorithms [2]: (i) *override* – it defines that for all pairs of corresponding elements in M_A and M_B, M_A's elements should prevail over M_B's corresponding elements when the composed model is generated; and (ii) *merge* – it defines that for all corresponding elements in M_A and M_B, the elements should be unified. Currently, there is very limited knowledge regarding the amount of effort required to apply model composition algorithms. Our work aim at quantifying model composition effort more precisely. We focuse on evaluating two main activities that are most influential into the composition effort. Firstly, we evaluate the composition effort when an algorithm is used to compose design models. In this case, we considered two different paradigms, object-oriented (OO) and aspect-oriented (AO). The number of conflicts or *conflict rate* was assessed in a composition scenario that each paradigm is likely to produce. Secondly, we measured the recovery effort, which is defined as the number of operations (i.e. insert, removal and update) that have to be performed to address emerging conflicts in the composed model, thereby making it reach the model initially intended by the software designer.

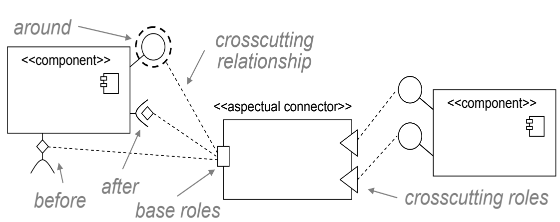

Figure 1 - AOM Language for Architectural Models [10]

Furthermore, we briefly identified two broad categories of conflicts: (i) *syntactic conflicts,* which arise when the composition algorithms result 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. Our analysis relied on syntactic and semantic conflicts recurrently investigated in recent studies [4]. We focused on those conflicts that often required more effort to be solved and that have been recurrently investigated in recent studies [4][12]. More details about the complete list of composition conflicts can be found at [4]. As aforementioned, the composition effort was evaluated using both OO and AO paradigms. For the latter, we used AOM to specify the system design. The use of AOM helps at improving the system modularity and, therefore, might reduce the burden of model composition tasks. The main goal of AOM is to provide software developers with means to express aspects and crosscutting relationships in their design models. The improved modularity provided by the use of AOM is related to the separation of concerns by supporting representation of concerns that cut across the system module. Crosscutting concerns are represented by a new model element, called *aspect*. Several AOM languages [10] have been proposed for modeling aspects at many levels of abstraction, ranging from use cases and architectural design to detailed software design.

Figure 1 illustrates the AOM language used in our study. The two main reasons for choosing this language are: (i) we selected architecture models as our focus due to the availability of existing models and their compositions in the history of the systems; and (ii) the language has been adopted for architectural modeling in other contexts [4][7] and provides a number of modeling features related to the modularity properties under analysis. The AOM approach is an extension of the UML's component diagram. The language allows expressing different forms of aspect-component collaborations. Aspectual connectors – represented by rectangles - define component interfaces, components and operations that are affected by the aspect. They are associated with the crosscutting relationships represented by dashed arrows. The language also provides means to represent the 3 types of advices available in aspect-oriented languages. So, *before* and *after* advices are represented by diamonds on the component interfaces, while the *around* advice is represented by a dashed circle. Finally, the language supports the visual modeling of pointcut designators and sequencing operators

3. STUDY SETTINGS

3.1 Research Question and Study Hypotheses

In this study, we compare the use of both OO and AO paradigms in composing design models for evolving software systems. The goal is to check to what extent the support for quantification and obliviousness in AOM makes a difference against OO modeling in terms of composition effort. Therefore, we derive the research question (RQ): *Can the obliviousness and quantification observed in the input aspect-oriented models help to reduce conflicts in output models?* Table 1 summarizes the study hypotheses. For each hypothesis, we have defined the null hypothesis and the alternative hypothesis, but only latter is represented in the table. Our analysis enconpasses 52 different composition scenarios provided by the two target systems, which contain different types of aspects with respect to their quantification and obliviousness.

Table 1 - Summary of Study Hypotheses

Hypothesis	Definition
H_1	Aspects with higher obliviousness can help to avoid the emergence of conflicts in AO models: CR (AO-model A) < CR (OO-model B)
H_2	Aspects with higher quantification in AO models lead to higher conflict rate: CR (AO-model A) > CR (OO-model B)

The hypothesis H_1 states that obliviousness and quantification may impact on composition effort. For example, aspects with higher quantification may lead to a higher *conflict rate* in the composed model. The more the number of aspects affecting the base components, higher is the number of relationships with the base components (more information has to be exposed). On the other hand, the hypothesis H_2 states that higher the obliviousness degree, the lower is the *conflict rate* in the composed models. A higher obliviousness means that the base elements are more unaware about the presence of aspects in the system design, which implies in a lower number of modifications required for implementing those aspects. As consequence, the composed models become less prone to present conflicts.

3.2 Target Applications

Systems with different characteristics were selected to evaluate the composition effort. Mobile Media (MM) is a software product line which purpose is to provide support for manipulation of photos, music and videos on mobile devices. In turn, Health Watcher (HW) is a framework that supports the registration and management of complaints to the public health system. The common reasons why MM and HW were selected as target applications are: (i) the architectural models are the artifacts used to reason about change requests and derive new products; and (ii) the original developers produced the architectural models without any of the model composition algorithms under assessment in mind, thereby avoiding any bias and entailing a more natural software development scenario. The evolution scenarios of MM range from changes in heterogeneous mobile platforms and additions of many alternatives and optional features [7]. Moreover, MM was selected as target applications for two specific reasons: (i) we had available a total of seven fully documented evolution scenarios, which could be expressed with model compositions; and (ii) different types of change were performed in each release, including refinements of the architecture style employed. On the other hand, during the evolution of HW many maintenance tasks have been performed. Most of those tasks are from adaptive and perfective nature [11]. The use of both OO and AO paradigms to express the architectural design allow us to investigate how they may impact on the system modularity, and hence, in the perfective and adaptive changes performed during the system evolution. Moreover, HW was selected as target application for two specific reasons: (i) all the 10 evolutions scenarios are available; and (ii) many changes were performed during the system evolution (e.g. insertion of design patterns to improve the system modularity).

3.3 Quantifying Composition Conflicts

Model composition effort is composed by two measures: *conflict rate* and *recovery effort*. Firstly we analyze the *conflict rate* that represents the number of conflicts identified in a composition scenario. Basically, the *conflict rate* relies on computing metrics that identify both syntactic and semantic conflicts. The metrics used to identify different types of syntactic and semantic conflicts in the composed model can be found at [3]. Though the collected metrics, it is possible to quantify the *conflict rate* from a specific composition scenario in the system evolution. As a result, we can calculate the density of conflicts that is represented by the conflict rate divided by the total number of elements in the composed model. Our second analysis is based on measuring the *recovery effort*, which represents the number of operations (creations, updates and removals) performed to make the composed model reach the intended model. If the composed model presented conflicts, a set of basics operations must performed by the developers in order to solve those conflicts in the composed model [9]. After measuring the *recovery effort*, the composed model is checked in order to verify if there is any occurrence of conflict propagation. This enables us to check whether the presence of aspects in the input models has any impact on the way that composition conflicts are propagated.

3.4 Obliviousness and Quantification Metrics

We have used specific metrics (see Table 2) to assess modularity properties in AO models. Firstly, we evaluated the quantification in the composed models by using the metric *Set of Join Points* (SJP). The SJP metric considers the set of join points in the base elements that are captured by the aspects in the input model. For each aspect, we considered the number of join points presented in each pointcut specified in the design model. Besides considering the explicit join points declared in the pointcuts, we are also considering all the join point references. In other words, when aspects make use of the *wildcard* mechanism, we are also counting all the references to join points in the base elements that are affected by the aspectual element.

Table 2 - Measuring Quantification and Obliviousness

Metric	**Description**
Quantification	Set of *join points* of a Pointcut (SJP)
Obliviousness	Number of modifications performed in the model elements (e.g. classes, parameters list) to receive change request in aspectual components.

Secondly, we evaluate the degree of obliviousness of the base elements regarding the presence of aspectual components. We considered the number of operations required to accommodate the aspectual components being added in the composed model. The idea is that the obliviousness should be quantified as the amount of preparatory actions performed on the base classes (or other aspects) to enable their interaction with aspects. The set of preparatory modifications performed in the base components indicates how they are (un) aware of the presence of aspects, as well as the changes required to implement them. The higher the number of preparatory modifications being implemented, the lower is the degree of obliviousness. We considered changes in terms of class inheritance, interfaces, methods and modifications in the methods parameters list. The collected measures allow us to compare whether models with a higher (or lower) obliviousness tended to present lower conflict rate.

3.5 Evaluation Procedures

In order to perform our investigation, we need to undergo a number of evaluation procedures that are described bellow:

Deriving AO and OO Model Releases. We used both OO and AO versions of our target systems to identify to what extent the presence of aspects may impact on the quality of the composed model. The composition algorithms were applied for each version of our target applications. The goal of using different composition algorithms was to identify if the outcomes, in terms of *conflict rate* and *recovery effort* were the same or not. All the releases of OO and AO versions realized the same changes.

Model Releases and Composition Specification. Different model compositions were considered for both target applications. For the MM, we considered 5 evolution scenarios using both paradigms, totaling 20 compositions. Similarly, for HW we considered 8 evolution scenarios, totaling 32 compositions. For each evolution scenario the *merge* and *override* algorithms were applied for both target systems. Moreover, changes performed during the system evolution are visible in the architectural design. After AO and OO model versions have been derived and the composition specified, the composition algorithms were applied and assessed the composition conflicts.

3.6 Execution and Assessment Phase

The execution and assessment phase is basically structured in three main activities, which are better described below:

Model Refactoring. It is an essential activity to define the input models, and hence, to express the model evolution as an explicit model composition relationship. The architectural models of both target applications were refactored as means to specify the delta model itself, as well as to represent the change scenarios as composition relationships. Before creating the delta model, it is necessary to identify the differences between the composition scenarios. For doing so, we considered the evolution description provided by the original designers in a previous study [7].

Model Composition and Measurement. Firstly, we focused on investigating the composition heuristics instead of explicit composition techniques. All the composition descriptions were documented, including the data collected from our metrics suite. Since a well-validated metrics suite for model composition is not available yet, the suite of metrics defined in our previous study [3] was used. Although those metrics have not been extensively validated, their feasibility and efficacy were evaluated [4].

Effort Assessment. To support a detailed data analysis, the assessment phase was further decomposed in three main stages aiming to: (i) identify the *conflict rate* produced during the model composition; (ii) assess the effort to resolve a set of conflicts previously identified. This stage also embraces the analysis of whether the use of aspects has a higher impact on the way composition conflicts are propagated; and (iii) compute the data regarding the quantification and obliviousness in the AO models to evaluate their impact on the *conflict rate*.

4. STUDY RESULTS

In this section we evaluate our study hypotheses based on the data collected after the composition algorithms have been applied. We tested if all the data follow a normal distribution by applying the Shapiro's test [1]. The main trends were also calculated. Finally, the Wilcoxon signed rank test **Error! Reference source not found.** was used to validate our hypotheses, as well as Pearson's correlation test **Error! Reference source not found.** was applied to analyze to what extent the modularity properties are related to the emergence of composition conflicts.

4.1 H1: Obliviousness and Conflict Rate

Our first hypothesis ($H_{1.0}$) evaluates whether the degree of obliviousness impact on *conflict rate* measures. As previously discussed, our hypothesis assumes that the higher the number of modifications required to resolve conflicts in the composed model, the lower is the obliviousness of the base elements. The base elements considered are those providing join points to aspects defined in the system design. Our investigation relies on applying both composition algorithms for each release of the target applications. In this sense, our analysis evaluated whether there is a positive correlation between obliviousness degree and *conflict rate* in the composed model is positive. Before testing the hypothesis, we evaluated if the degree of obliviousness in AO and OO models are different. For doing so, we applied the Wilcoxon signed-rank test since our goal was compare release by release the obliviousness of the base elements in the system design. As a result, the test showed a p-value = 0.001 which is lower than the significance level adopted in our study. The results indicated a difference between the degrees of obliviousness when using different paradigms for representing the system architecture.

Table 3 – Correlation Analysis between Quantification and Conflict Rate

Variable	Median	Mean	S.D.	Corr.
Obliviousness	3	7.333	8.795	0.497
Conflicts	0.105	0.498	0.735	

As a second step, a Pearson's correlation test was performed to measure the strength of the linear relationship between degree of obliviousness and conflict rate. Table 3 summarizes the results of applying the Pearson's correlation test. Assuming a sample size (SS) = 26 and p-value = 0.05, the correlation test presented a calculated p-value = 0.009654 and a correlation coefficient = 0.497829. The calculated correlation coefficient indicates that there is a moderate relationship between the obliviousness degree and *conflict rate*. In other words, our correlation analysis suggests that the higher the number of modifications in the base elements to accommodate changes related to aspectual components, the higher the *conflict rate* in the composed models. A higher number of modifications implies in a lower degree of obliviousness. In this sense, the alternative hypothesis $H_{1.1}$ is confirmed and we can say that, in general, AO models with lower obliviousness degree tend significantly to present higher *conflict rate*.

4.2 H2: Quantification and Conflict Rate

This section discusses to what extent quantification has impact on the emergence of conflicts in the composed models. Similarly to the first hypothesis, the results remain consistent independently of the composition algorithm used in all the composition scenarios in both target systems. Considering all the evolution scenarios, we found that the number of join points in the HW is higher than in the MM. A higher number of join joints, which implies in a higher quantification, might yield to a higher number of conflicts. The quantification can also be affected by the level of details of the information exposed in the input models. For example, when analyzing the MM models a low number of join points can be observed if compared with the HW models. It is caused due to the high abstraction of MM models, which implies in less information being exposed to the system developers. The smaller the amount of information that can be observed regarding the join points that are affected by the aspects, the lower are the quantification measures that can be collected; and (ii) the conflict rate in the HW is smaller than in MM, since in the latter most part of the pointcuts in the aspects affect only 1 or 2 join points. In turn, most part of the aspects in the HW has pointcuts affecting more than 3 joint points.

Table 4 – Correlation Analysis between Quantification and Conflict Rate

Variable	Median	Mean	S.D.	Corr.
Quantification	9.0	15.17	15.65	0.470
Conflicts	0.105	0.498	0.735	

Moreover, we also observed that although the HW design models have a higher level of details, the larger number of elements does not necessarily generate more conflicts in the composed model. Our correlation analysis is aimed at examining whether the *conflict rate* are directly related to the quantification degree. In order to examine the strength of relationship between *conflict rate* and the quantification degree, we have applied the Spearman' correlation. Table 4 shows descriptive statistics related to the correlation between quantification and *conflict rate* in AO. Assuming that the sample size (SS) = 26 and p-value = 0.05, the Pearson's correlation test presented A correlation coefficient = 0.470866 and calculated p-value = 0.0202. The correlation coefficient presented a positive value that indicated a moderate correlation between the quantification degree and *conflict rate* observed in the composed models. That is, the results suggest that when the quantification in the AO models increases, the *conflict rate* increase as well. As the calculated p-value is lower than 0.05, the correlation results are statistically significant. In this way, the hypothesis $H_{2.1}$, that aspect with higher quantification tend significantly to present higher *conflict rate*, is confirmed.

5. DISCUSSION

We investigated which specific design practices in AOM might increase the model composition effort. The results revealed that many conflicts were often associated with the misuse of the AO mechanisms, which tend to either harmfully reduce obliviousness or increase quantification.

Correlation between Modularity Anomalies vs. Harmful Obliviousness and Quantification. We investigated how the presence of specific modularity anomalies could impact on the composition effort. The results showed that the presence of modularity anomalies [13] is often caused by the misuse of AO mechanisms, such as pointcuts and advice. Thus, our focus was on the modularity anomalies related to misuse of quantification mechanism and the harmful obliviousness reduction. The input models of each composition scenario were analyzed in both target systems to check whether the presence of modularity anomalies would impact on the composition effort. Modularity anomalies were independently detected for each composition scenario in a previous study [13] and confirmed by the actual developers. A modularity anomaly is related to conflicts if the latter is manifested in a design element in the input models that was anomalous (i.e. elements containing that contain at least an instance of a design anomaly).

Types of Modularity Anomalies Studied. Nine types of modularity anomalies were identified: (i) Composition Bloat; (ii) Forced Join Point; (iii) Duplicated Pointcut; (iv) Anonymous Pointcut; (v) God Aspect; (vi) Lazy Aspect; (vii) God Pointcut; (viii) Idle Pointcut and (ix) Redundant Pointcut. Those modularity anomalies have been documented in previous work [13], and most part of them is directly related to the misuse of pointcuts. For example, aspects exhibiting the anomaly Anonymous Pointcut might have high quantification and can possibly generate more conflicts in the composed models. The Anonymous Pointcut has no signature and all the information is exposed in a pointcut expression. For example, when a pointcut affecting 3 join points is represented by an expression, each of these join points are represented through a different relationship between aspectual component and the affected base element(s). Thus, the higher the number of information exposed in a pointcut expression, higher is possibility of conflicts been arising in the composed model.

Correlation between Modularity Anomalies and Composition Conflicts. We collected instances of each modularity anomaly and computed the conflicts for all the composition scenarios. Tables 5 and 6 show the input data used for the correlation test between the modularity anomalies and the composition conflicts. The *conflict rate,* observed in each release when the *merge* and *override* algorithms are applied, is represented by CR-M and CR-O, respectively. When analyzing the MM, only in the composition scenario R3 (R_N – represents the number of the release in the system under analysis) the presence of modularity anomalies (M.A) not impact in the emergence of conflicts. Although 6 instances of modularity anomalies were identified, they have not produced any composition conflicts when one of the composition algorithms was applied. In other composition scenarios of the MM, we observed that the presence of modularity anomalies influenced the emergence of composition conflicts.

Table 5 – Correlation Analysis between Conflicts and Modularity Anomalies for Mobile Media

MM	R1	R2	R3	R4	R5
M.A.	1	3	6	7	7
CR-M	38	21	0	66	0
CR-O	38	36	0	73	6

Table 6 – Correlation Analysis between Conflicts and Modularity Anomalies for Health Watcher

HW	R1	R2	R3	R4	R5	R6	R7	R8
M.A.	22	27	49	51	51	50	50	50
CR-M	50	10	1	0	2	3	0	11
CR-O	41	4	1	0	0	0	0	34

Another interesting effect of the presence of anomalies could be observed in the HW system where the number of anomalies in the input models is fairly high. As example we can mention the composition scenarios from R3 to R7 where most part of the modularity anomalies are not related to the emergence of conflicts - none of the elements involved in the composition were infected with any modularity anomaly. Moreover, from release R6 onwards the anomalies remain constant during the system evolution. The low number of conflicts from release R3 to R7 in the HW is explained by the fact that in those scenarios the system evolution consists basically on applying design patterns to improve the system modularity. A more critical scenario occurs when instances of modularity anomalies are propagated through the system releases. Considering the evolution scenario from release R1 to R2, for example, there are some cases where Anonymous Pointcut and Duplicated Pointcut anomalies are observed in the input models, but not solved. For these composition scenarios the number of instances of those anomalies are the same of the previous release. Therefore, a Pearson's correlation test was applied to verify whether there is true correlation between those modularity anomalies and the conflicts when considering all the composition scenarios. Table 7 shows that the correlation coefficient presented a positive value equals to 0.194, which indicates a positive correlation but with a low statistical significance. We also observed that most part of the modularity anomalies identified in HW is related to pointcut problems, which somehow expected since the aspectual elements of the HW models have a high quantification.

Table 7 – Correlation: Obliviousness and Conflict Rate

Variable	Median	Mean	S.D.	Corr.
Anomalies	49	36.54	20.39	0.194
Conflicts	0.105	0.498	0.735	

Most Frequent Anomalies related to Model Composition Conflicts. In the MM system, the 3 most recurrent modularity anomalies were Duplicate Pointcut, God Aspect and Lazy Aspect, considering the total of modularity anomalies of each release. They are responsible for around 95% of the modularity anomalies presented in MM releases. In turn, for the HW system the 3 most recurrent modularity anomalies were Redundant Pointcut, Anonymous Pointcut and Idle Pointcut. Those modularity anomalies are responsible for more than 85 % of the total of modularity anomalies in the HW When analyzing the total number of conflicts related to the presence of modularity anomalies, we found that: (i) the 3 most recurrent anomalies of MM were responsible for around 64% of composition conflicts related to the presence of modularity anomalies. From the total number of conflicts caused by those anomalies, the percentage of conflicts related to the anomalies Duplicated Pointcut, God Aspect and Lazy Aspect are 72%, 16% and 12%, respectively; and (ii) the number of conflicts related to those three anomalies is even higher HW system, reaching 92%. From this high percentage of conflicts, we found that around: 64% of conflicts are related to Redundant Pointcut, 20 % are related to Anonymous Pointcut, and around 16% are related to Idle Pointcut. The results showed that pointcut-related anomalies are the ones consistently causing more composition conflicts in both systems.

6. TREATS TO VALIDITY

Internal Validity. Our study met the internal validity due to two reasons; (i) the temporal precedence criterion was met, i.e. the instability of design models preceded the inconsistencies and composition effort; and (ii) the covariation of the independent and dependent variables was observed, i.e. the magnitude of the effects on the *conflict rate* and *recovery effort* depended on variations in the modeling approaches (AO and non-AOM).

Construct Validity. We evaluated whether the quantification method is correct and if it was accurately done, as well as whether the manual composition threatens the validity. To mitigate these threats, we established study guidelines and engaged the authors and the developers of the target applications in discussions about the problems observed with the data collection. We also checked if the quantification procedures were carefully planned and followed well-known quantification guidelines [1].

External Threats. External threats are related to limitations to generalize our results in a broader context (only two target applications and a single AOM language were used). Other empirical studies with compositions of larger UML models are required. Moreover, the number of properties and details (i.e. granularity) of the model elements in each composition scenario may directly affect the results. Further empirical evaluations are indeed fundamental to confirm or refute our findings in other real-world design settings involving UML model compositions.

7. FINAL REMARKS

This paper presents a quantitative study to assess the potential advantage of aspect-orientation in reducing conflict resolution effort. The first finding is that, even in scenarios where *conflict rate* of AO models was so close to (or higher than) the *conflict rate* of OO model. The conflict was similar in AO and OO models. As expected, we also found that the presence of aspects in input models improved modularization and, therefore, tended to better localize conflicts. We also observed that: (i) a higher degree of obliviousness in the AO models led to a significant decrease of conflicts when compared with the OO model counterparts; and (ii) aspects with higher quantification were the cause of higher *conflict rate* in AO models. As a second finding, we identified *modularity anomalies* in the input models that can be related with the *conflict rate* and *recovery effort*. Our goal was to identify what type of *modularity anomaly* is more often observed, so that we can suggest which refactoring in AO models can be performed to reduce the composition effort. Developers must be careful when using AO languages to build the system design and avoid cases where: (i) the aspects have a high quantification and obliviousness is low; (ii) some modularity anomalies in the input models can be related to the high quantification of the aspect's points or the misuse of AO mechanisms; and (iii) the overuse of aspects with high quantification, particularly those pointcuts that are associated with the modularity anomalies investigated in this study.

8. REFERENCES

[1] Chitchyan, R. *et al*., 2005. Survey of Aspect-Oriented Analysis and Design Approaches, Technical Report AOSD-Europe-ULANC-9, AOSD-Europe, May.

[2] Clarke, S., 2001. Composition of Object-Oriented Software Design Models. PhD Thesis, Dublin University, January.

[3] Farias, K., Garcia, A., Whittle, J., 2008. On the Quantitative Assessment of Class Model Compositions: An Exploratory Study, In Proc. of MODELS'08, France.

[4] Farias, K., Garcia, A., Whittle, J. 2010. Assessing the Impact of Aspect on Model Composition Effort, In Proc. of AOSD'10, pp. 73-84, Oslo, Norway.

[5] Farias, K., Garcia, A., Lucena, C., 2012a. Evaluating the Impact of Aspects on Inconsistency Detection Effort: A Controlled Experiment, *In Proc. of the 15th MODELS.*

[6] Farias, K., Garcia, A., Lucena, C., 2012c. Effects of Stability on Model Composition Effort: an Exploratory Study. *Journal of Software and Systems Modeling*.

[7] Figueiredo, E. *et. al*., 2008. Evolving Software Product Lines with Aspects: An empirical Study on Design Stability, *In Proc. of ICSE'08*, pages 261-270, Germany.

[8] Filman, R., Friedman, D., 2001. Aspect-Oriented Programming is Quantification and Obliviousness. *RIACS Technical Report in Proc. of OOPSLA'01*.

[9] Fleury, F. et al., 2007. A Generic Approach for Automatic Model Composition, *In Proc. of the Workshop on Aspect-Oriented Modeling at MoDELS*, pp., 7-15.

[10] Garcia. A. et. al., 2009 Representing Architectural Aspects with a Symmetric Approach, In Proc. of the 15th Int'l Workshop on Early Aspects, AOSD'09, USA.

[11] Greenwood, P. *et al*., 2007. On the Impact of Aspectual Decompositions on Design Stability: An Empirical Study, *In Proc. ECOOP'07*, Germany.

[12] Guimarães, E.; Garcia, A.; Farias, K., 2010. Analyzing the Effects of Aspect Properties on Model Composition Effort: A Replicated Study. *In Proc. of the 6th AOM Workshop on Aspect-Oriented Modeling @ MODELS*, Norway.

[13] Macia, I. et al., 2011. An Exploratory Study of Code Smells in Evolving Aspect-Oriented Systems, *In Proc. of AOSD'11*, pp. 203-214, Brazil.

[14] Wholin, C., Runeson P., Host, M., Ohlsson, M., B, Regnell, B., Wésslen, A., 2000. Experimentation in Software Engineering: an Introduction. Kluwer Academic Publishers, Norwell, US.