On the Comparative Evaluation of Aspect-Oriented Model Composition Techniques

Kleinner Farias, Alessandro Garcia, Carlos Lucena
Opus Research Group - LES, Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Rio de Janeiro – RJ – Brazil
{kfarias,afgarcia,lucena}@inf.puc-rio.br

***Abstract*—Aspect-oriented model (AOM) composition techniques are responsible for expressing crosscutting relationships between aspect and base models. However, given their growing heterogeneity, it is particularly challenging for designers to both objectively assess them and take decision on which is the best according to their supported composition facilities. This work provides a systematic comparison of representative AOM composition techniques. For this purpose, we propose a set of proposed evaluation criteria: the model composition process followed; the manner of matching the input models; the type of composition strategies used; the manner of assessing the composition results; and how the techniques address conflict resolution. Such criteria enable us to pinpoint the commonalities and differences as well as strengths and shortcomings of the three studied techniques.**

Keywords: model composition, aspect-oriented modeling, aspect-oriented model composition techniques

I. INTRODUCTION

Model composition can be viewed as an operation where a set of activities should be performed over two input models, M_a (the *receiving model* or *base model*) and M_b (the *aspect model*), in order to produce an output model M_{ab} (composed model). It is put in practice by establishing a composition relationship that is denoted by the equation: $M_a + M_b \rightarrow M_{ab}$. One example of a typical composition relationship would be the crosscutting relationship present in Aspect-Oriented Modeling (AOM) languages.

AOM languages aim at improving separation of concerns by supporting the modular representation of requirements that cut across multiple software modules in a new model element, namely *aspect*. At the core of such languages lives a well-defined model composition technique, which is applied to define the crosscutting relationship between aspect and base models. A particular challenge is to cope with the potential conflicts that may occur in aspect interactions. For example, when multiple aspects crosscut the same base model an undesired superimposition of the aspectual behaviors at the same join point can arise. Moreover, different composition orders may lead to various inconsistency problems, either due to the side-effects caused by the superimposition or due to the requirements enforced by the system [1]. For example, the sale aspect may have to be only applied in the presence of the security aspect since it is only permitted to sell something in secure environment settings.

Some techniques have already been developed and reached a considerable level of maturity. However, given their growing number and heterogeneity, it is particularly challenging for designers to both objectively assess them and take decision on which is the best approach. This can be explained by: (i) most of the AOM composition techniques pursue distinguished model composition strategies; (ii) they use different kinds of model comparison algorithms, sometimes exploiting syntactic and semantic issues during the matching; (iii) not all techniques are able to identify and resolve model composition conflicts; and (iv) most of them do not have a well-defined model composition process.

There is a need for both AOSD and MDD communities to provide a clear identification of commonalities and differences between the current techniques. To the best of my knowledge, there is no work in this sense. Similar, but focusing on different points of view [2], [3], [4], existing surveys in AOM are more comprehensible with respect to the target development phases, mapping of concepts between AOP (programming level) and AOM (modeling level) rather than to present a comparative view of the model composition techniques.

This work intends to systematically investigate a representative set of AOM composition techniques. As a first step, we present a set of criteria that are used to assess the studied techniques and are capitalized from previous experience [5] in the design and implementation of model composition techniques (Section II). Then, we compare and pinpoint the commonalities and differences as well as strengths and shortcomings of the studied techniques. For this, we make use of the proposed criteria (Section III). Finally, we present final remarks and future work (Section IV).

II. EVALUATION SETTINGS

A significant factor behind the difficulty of comparing model composition techniques is the lack of a common understanding of the basic requirements and criteria to be taken into consideration. In part, this fact can be explained, on the one hand, due to different concepts that have been created and diffused by AOM-based approaches, including: (*i*) Theme/UML [6], [7] - the most popular of them, (*ii*) France's approach [8], [9], [10], [11], and (*iii*) Cottenier's

Motorola Weaver [1], [12], [13]. Such techniques are the target of this work for two main reasons. First, they rely on very heterogeneous model composition characteristics and particularities. Second, they have been published in flagship conferences or journals.

The rationale behind the selection of the evaluation criteria (discussed below) is to assort specific points of evaluation that are natural of the model composition realm. For instance, composition conflicts that arise from using specific composition strategies and manipulating model properties (e.g., set *false* to the boolean property, *isAbstract*, instead of *true*) previously defined on their metamodels. The proposed criteria are discussed in what follows.

Model Composition Process. It describes a well-defined, step-wise guidance, made up of a set of activities, for enabling designers to: (i) accomplish reliable model composition, and (ii) avoid errors during the composition of the input models through explicit application of good practices. Thus, the presence of guiding activities and an accompanying workflow can yield superior understandability and usability of the proposed model composition technique.

Model Comparison. A basic step to achieve consistent model composition lays in the ability to compare input model elements. This criterion aims at assessing the activity of finding commonalities and differences between the input models. For instance, before composing Ma and Mb, it is required to verify the existence of semantic and syntactic overlaps between them. Overlaps between input models are sometimes undesired as they can lead to syntactical and semantic conflicts, misinterpretation of the output models when the composition is accomplished.

Model Fusion. Although the model composition techniques can be suitable to merge models in specific scenarios (e.g., model enhancement), their merge algorithms may not be suitable for other scenarios (e.g., model deletion). Thus, techniques tend to make use of different types of algorithms, namely composition strategies, to fit for any new need. This criterion is focused on assessing the types of strategies used, particularly the strategies *merge*, *override*, and *union*. Note that these strategies define the semantics of the composition relationship.

Composition Evaluation. This criterion evaluates if the techniques provide means for assessing the effects of the integration of the input models. Given the growing number and heterogeneity of model composition strategies, it is difficult to identify and systematically quantify undesirable phenomena that arise from the output composed models, e.g. inconsistent models at composition-time. Thus, the lack of such evaluation support can cause serious impairments on the adoption of the technique in mainstream software development. First, there is a lack of canonical set of indicators to quantify harmful properties associated with the output models, such as composition conflicts. Second, designers ultimately rely on feedback from experts to determine goodness of specific composition strategies and respective composed models.

Model Transformation. This criterion is dedicated to assess if and how the techniques deal with conflicts (e.g., semantic conflict) and problems (e.g., modularity anomalies) identified at the composition evaluation stage. The goal is to transform an unsound model into a sound model. With this need in mind, this criterion pretends checking if a technique provides or not, model conflict identification and resolution. Conflict resolution may be based on a mechanism to detect conflicts and then resolve them automatically, or manually with the help of designer.

III. COMPARATIVE ANALYSIS OF AO MODEL COMPOSITION TECHNIQUES

A comparative analysis of the AO model composition techniques are presented and discussed throughout this section. In particular, we systematically assessing them in terms of the criteria proposed in Section II. An ideal real-world model composition technique should combine the strengths of the existing approaches, which are individually described in the following. Each technique was analyzed based on the respective references mentioned in Section II.

A. *The Motorola Weavr Approach of Cottenier* et al.

The authors tackle an important and well-know problem in AOSD: the aspect-to-aspect interference, which occurs when multiple aspects are deployed jointly such that different composition orders may give rise to various inconsistency problems. The interference is due to order of composition not to be correctly specified. Thus, the authors describe how aspect precedence can be specified explicitly at the modeling level in order to derive a correct composition order and therefore reduce the aspect interference problem in AOM.

Model Composition Process. The approach encompasses two phases in the composition process: (i) *advice instantiation*, in which advices are instantiated based on the pointcuts they are bound to, and (ii) *advice instance binding*, in which aspects are woven into the base models in one of the following two ways: wrapping or inlining. In the wrapping mode, the original joinpoint actions are replaced by an operation call to the corresponding advice instance. For the inlining mode, all advice instances are actually inlined in the base model [12]. However, it is not provided a well-defined composition workflow with all the flows of activities put together in a synchronized manner. Moreover, the paper does not describe how to analyze or detect aspect interferences, or supporting reasoning about the system correctness after simultaneously composing multiple aspects. In short, a design process for the approach has not yet been described.

Model Comparison. The model comparison mechanism is based on the bind relationship and on regular expression matching. It ignores important issues of model comparison that may lead to problems, such as: (i) lack of flexibility to determine correspondences among model elements; (ii) poor user interaction; (iii) lack of focus on certain model properties; (iv) require a large amount of human effort i.e., it is necessary to analyze all model properties; (v) most do not scale up to dealing with complex models; (vi) do not take into account the model semantics during the matching.

Model Fusion. The model composition is not based on composition strategies and specifies a clear distinction between aspectual model (the crosscutting concern) and base model when a composition relationship is established, thus being an asymmetric approach. This distinction is supported by use of a special kind of construct stereotyped by the name <<advice>>. The latter encapsulates the behaviour of the crosscutting concern, <<aspect>>, in which pointcuts and advices are encapsulated. However, the approach does not explicitly tackle semantics or syntactic issues, which are central elements as model composition to be put in practice. As model composition is strictly related to ability to handle the model properties defined in the UML metamodel, if an approach does not take into account such properties, it can have potentially serious problems. For example, it can give rise for semantics and syntactical conflicts.

Composition Evaluation. The approach does not provide means for assessing the side-effects of the model integration. In such circumstances, it can give rise for serious impairments on the adoption of it by other designers and developers in mainstream software development. Note that if end user is not expert on model composition, he/she is going to have serious problems to validate the output model, basically to define whether it is valid or not, a simple activity but that can impact on a whole model-driven development process. One of the shortcomings we can point out is the absence of a set of indicators (metrics or heurists) to quantify harmful properties associated with the output models, such as composition conflicts and modularity anomalies. Consequently, it can compromise the process of decision whether the output model is valid or not. To resolve possible conflicts, the approach allows only defining precedence relationships between aspects using a stereotype <<follows>>, derived from the UML meta-class Dependency.

Model Transformation. The approach is not concerned on solving the natural problems that can arise from the composition. Consequently, model refactoring and model transformation is not taken into consideration.

B. *Model Composition Approach of France* et. al.

France et. al. try to show how a signature-based composition procedure can be used to compose class models and describe how composition directives can be used to ensure that the composition procedure produces desired output models. Aspects are modeled using template diagrams, i.e., package diagram templates, class diagram templates and communication diagram templates. It envisages an aspect model as a primary model, and one or more aspect models that each describes a feature that crosscuts the primary model.

Model Composition Process. Regarding the model composition process, it is defined as a set of aligned and linked activities to orchestrate the model composition and solution of problems that can arise from natural composition process. For this, a set of composition directives plays a central role.

Model Comparison. A name-based composition procedure was used to merge UML models . Model elements with the same name are merged to form a single element in the composed model. The composition procedure assumes that elements with the same name represent consistent views of the same concept. However, this may not always be the case. For example, consider an aspect-oriented design consisting of a primary model that describes a class representing a server that provides unrestricted access to services via operations in the class, and an instantiated aspect model that describes the same server class with access control features. In this case, simple name-based merging of the two classes and the operations in them could lead to operations that are associated with inconsistent specifications (a primary model operation and its corresponding aspect model operation would have the same name, but different argument lists and specifications). Often, a more sophisticated form of composition is needed to produce composed models with required properties. To meet this need, the authors proposed the use of composition directives to ensure that the name-based composition procedure produces desired results.

Model Fusion. The approach divides explicitly the models into the primary and aspect models, assigning roles for each one throughout the model composition process, thus assuming an asymmetric approach. However, the approach does not tackle semantics issues, which can compromise, in fact, the composition as whole. With respect to using of UML templates, the approach is similar to Theme/UML. For readability purposes, however, the authors prefer to provide a notation different to standard UML templates and in contrast denote template model elements using '|'. This notation is based on the Role-Based Metamodeling Language [8]**,** which is a UML-based pattern language designed as an extension to the UML. The use of packages for capturing concerns caters for scalability, although this has not yet been demonstrated within an example encompassing several concerns.

Composition Evaluation. The approach provides means for assessing the side-effects of the model integration. Although it is not a robust mechanism for assessing model composition, an initial efforts is done in this direction. Likewise the Cottenier's proposal, this fact can also give rise for serious impairments on the adoption of it by designers and developers in mainstream software development. Moreover, a set of directives are proposed to handle the possible conflicts and problems during the composition, namely model directive that specifies the order in which aspects are composed with a primary model. Two stereotypes are presented to determine the order of the composition as well as the Cottenier's proposal, such as <<follow>> and <<proceeds>>.

Model Transformation. Following the model composition process proposed by France *et. al.* in [11], after composition, the composed model can be formally analyzed against desired properties to uncover design errors. For example, one can analyze the models against well-formedness rules to identify badly formed models or one can analyze the models against desired semantic properties. The authors also propose a technique for uncovering semantic problems during composition in [11]**.** In the approach, the semantic property to be verified is used in the composition process to generate proof obligations. Establishing that a composed model has the stated semantics properties requires discharging the proof obligations. In some cases, the uncovered problems can be resolved using composition directives. For this end, an appropriate set of directives are identified and used to compose the context-specific aspect and primary models. In other cases, more substantial changes may be required. For example, it may be determined that another variant of the aspect model is needed or that the primary model has to be significantly refactored. Thus, it represents a clear open research question. Unlike the Cottenier's proposal, this approach has a clear concern with respect to conflict identification and resolution.

C. *The Theme/UML Approach of Clarke* et al.

The Theme/UML approach is based on UML and presents a new approach of decomposing requirements and aligning them within design models according to a *theme,* and composing them back using *composition relationship*. The approach is strongly based on its model composition technique and currently is integrated to a model-driven development process. Theme/UML poses no restrictions on what UML diagrams might be used for modeling. Nevertheless, package and class diagrams are particularly used for modeling structure and sequence diagrams are used for behavioural modeling.

Model Composition Process. Three phases are defined: modelling, composition, and transformation phase. The modelling phase comprises the production of aspect-oriented models compatible with the Theme/UML language. The composition phase relies on the Eclipse Theme/UML plugin to perform composition of the AO and OO base model, resulting in a Platform Independent Model (PIM). In the transformation, the PIM is transformed into a Platform Specific Model (PSM) via the plug-in. This model is then transformed into source code using openArchitectureWare [15].

Model Comparison. It introduces composition semantics for UML class diagrams. The approach defines a new design construct, called composition relationship that supports the specification of how design models should be composed. With this composition relationship, it is possible to: (i) identify and specify overlapping and non-overlapping concepts, and (ii) specify how models should be integrated, and how conflicts in equivalent elements are reconciled. The identification of the overlapping parts is based on the name of the input models alone and, as a consequence, it is a weakness of the approach. Moreover, the approach does not take into account the semantics of the model during the matching process. This can be very restrictive and compromise the model composition process as a whole.

Model Fusion. Concerns are encapsulated in UML packages denoted with a stereotype <<theme>>. Composition in this approach is based on composition strategies, which operate over an aspectual model (the representation of the crosscutting concern) and the base model. However, there is no explicit notational distinction between such input models; both are treated as themes, which can characterize the approach as being symmetric. Three types of model composition have been defined, which can be employed throughout the process, i.e. the merge, override, and bind composition strategies. Each strategy owns a specific semantics.

Model Transformation. The approach presents an initial effort to tackle conflicts that can emerge during the composition. However, for the best of our knowledge, it was the first work to pay attention for identifying and solving conflicts of model composition. It is presented no clear conflict resolution strategy through which identified conflicts are solved. Such conflicts can emerge due to models have been specified separately and there might be differences in their specifications. Hence, these differences must be reconciled for the output model. In short, reconciliation strategies are employed to overcome conflicts and problems in the output model.

IV. CONCLUSIONS AND FUTURE WORK

Model composition is one of the pillars of AOM. In this work we tried to systematically analyze the distinguishing characteristics of a representative set of AOM composition techniques. We presented a set of criteria for enabling a comparative evaluation of such techniques. Although we can conclude that studied techniques have reached a considerable maturity, they still present some limitations that should be further investigated. For example, when precise, well-defined models are defined, it is common to associate them with semantics constraints. These constraints should be considered when performing model composition, so that the specified semantics is not challenged. The studied techniques are not able yet to deal with the issue of merging such constraints.

Even though the studied techniques have been fully implemented [1], [7], [11], [12], [13] in-depth empirical studies are still required to evaluate the scalability of the AOM composition approaches in real-world settings. We also need to better understand their applicability and complementarities in different application domains. Finally, we note that improvement in model composition techniques is paramount to the adoption and evolution of aspect-oriented modelling and model-driven engineering in general. We hope that the issues outlined throughout the paper encourage researchers to cope with AO model composition problems, thus fostering a new generation of techniques and tools to support AOM in a more effective manner.

REFERENCES

[1] T. Cottenier, A. Berg, and T. Elrad, "The Motorola WEAVR: Model Weaving In A Large Industrial Context", In: 6th AOSD, Vancouver, Canada, March, 2007.

[2] A. Schauerhuber, W. Schwinger, E. Kapsammer, W. Retschitzegger, M. Wimmer, and G. Kappel, "A Survey on Aspect-Oriented Modeling Approaches", Technical Report, Vienna University of Technology, October 2007, avaible at www.bioinf.jku.at/publications/ifs/2006/1506.pdf

[3] R. Chitchyan, A. Rashid, P. Sawyer, A. Garcia, M. Pinto, J. Bakker, B. Tekinerdogan, S. Clarke, and A. Jackson, "Survey of Aspect-Oriented Analysis and Design Approaches", Technical Report D11 AOSD-Europe-ULANC-9, AOSD-Europe, May, 2005.

[4] G. Blair, L. Blair, A. Rashid, A. Moreira, J. Araujo, and R. Chitchyan, "Engineering Aspect-Oriented Systems", In: Aspect-Oriented Software Development, pp. 379–406, Addison-Wesley, Boston, 2005.

[5] K. Oliveira, K. Breitman, and T. Oliveira, "A Flexible Strategy-Based Model Comparison Approach: Bridging the Syntactic and Semantic Gap", Journal of Universal Computer Science (to appear).

[6] S. Clarke, "Extending Standard UML with Model Composition Semantics", Science of Computer Programming, vol. 44, no. 1, pp. 71–100, July, 2002.

[7] S. Clarke, and E. Banaissad, "Aspect-Oriented Analysis and Design the Theme Approach", Addison-Wesley, Upper Saddle River, March 2005.

[8] R. France, D. Kim, S. Ghosh, and E. Song, "A UML-Based Pattern Specification Technique", IEEE Trans. Software Eng., vol. 30, no. 3, pp. 193–206, 2004.

[9] R. France, I. Ray, G. Georg, and S. Ghosh, "Aspect-oriented Approach to Early Design Modeling", IEE Proceedings Software, vol. 151, no. 4, pp. 173–185, August 2004.

[10] R. France, S. Ghosh, T. Trong, "Model Driven Development Using UML 2.0: Promises and Pitfalls", IEEE Computer Society, vol. 39, no. 2, pp. 59–66, February, 2006.

[11] R. Reddy, S. Ghosh, R. France, G. Straw, J. Bieman, E. Song, and G. Georg, "Directives for Composing Aspect-Oriented Design Class Models", Transactions on AOSD I, LNCS 3880, pp. 75–105, Springer-Verlag, 2006.

[12] J. Zhang, T. Cottenier, A. Berg, and J. Gray, "Aspect Composition in the Motorola Aspect-Oriented Modeling Weaver", vol. 6, no. 7, Journal of Object Technology, 2007

[13] T. Cottenier, A. Berg, and T. Elrad, "Joinpoint Inference from Behavioral Specification to Implementation", In: 21st European. Conference on Object-Oriented Programming, Berlin, Germany, July, 2007.

[14] Unified Modeling Language: Infrastructure version 2.1, Object Management Group, February, 2007.

[15] openArchitectureWare, The openArchitectureWare Project Team, on line URL: http://www.openarchitectureware.org/, 2009.