Semantics Preserving Model Composition

Jon Oldevik

University of Oslo, Department of Informatics,
SINTEF Information and Communication Technology
Oslo, Norway
jonold at ifi.uio.no

1 Introduction

Separation of concerns (SoC) and modularisation are well established strategies for managing complex specifications [1, 2]. However, although software is designed with SoC in mind, the language mechanisms at hand often lead to tangling and scattering of concerns. This has motivated a range of language extensions to support concern specification, such as aspects and subjects in programming and modelling. The current trend is modularisation of cross-cutting concerns into units, e.g. aspects, that can later be composed by some transformation process (composition/merging/weaving). An important issue in this process is how the semantics of the models/programs is preserved.

The focus of this PhD work is on composition and configuration of software specifications from a modelling perspective. Standard mechanisms in modelling (e.g. in UML) provide composition and configuration with well understood characteristics. Examples from UML are class redefinitions, composite structures, composite states, structured activities, interaction decomposition, and package merge.

This work goes beyond those by exploring modelling and composition of concerns at a collaboration level, focusing on their architecture and interaction dimensions. The semantics governing such compositions and their results is of particular interest in this regard. I will address how generative techniques can be used for implementing the compositions and guide semantics preservation. I will also address what semantics preservation means in different modelling and composition contexts. UML is used as the language for experimentation, as this is the de-facto standard for software modelling.

2 Problem Description

The problems of managing SoC are still far from solved. UML provides many sublanguages that allow describing a system from different perspectives, hence providing a certain SoC. However, it is hard not to scatter and tangle concerns, since mechanisms for composing them are lacking. In responsibility-driven design and role modelling, the main focus was interacting roles providing SoC [3]. The OOram method [4] even proposed a mechanism for synthesising (composing) role models. Similar mechanisms exist in UML in the Collaboration and Collaboration Use concepts. These should be well suited for representing separation of concerns in UML. What is missing are mechanisms for composing concerns and reasoning about the semantics of the concerns and the composed result. If two concerns are defined as collaborations (or classes) containing structure and behaviour, it is not obvious nor well-defined how to compose them. This work addresses composition of collaborations/structured classes that contain behaviour defined by interactions and how semantics of the models are preserved in the composition.

In this work, *composition* denotes the combination of two or more model elements (classes, behaviours, etc.) into new model elements. The composition should *preserve semantics* of whole or parts of its constituents. An open question in this regard is how to arrive at a definition of semantics preservation, both formally and pragmatically.

Interactions in the form of sequence diagrams are frequently used to describe high-level requirements, architecture, and detailed design. The built-in mechanisms of UML allow composition of interactions through references (interaction uses). When binding roles (parts) from two classifiers together (as in UML Collaboration Uses), there is no way of specifying how the owned interactions should be composed. Composition of structure is better understood, at least when it comes to binding together parts. However, introducing new parts (or ports) to a structure or refining communication links (connectors) to more complex structures are more challenging. The latter issues are particularly interesting from an architectural modelling perspective. This work will address both syntactic and semantic issues of such compositions.

In product line engineering (PLE) [5] there are specific challenges in representing features and producing product variants. The relationship with mechanisms for concern representation (e.g. hyperspaces, aspects)[6, 7] have been addressed by the research community. Generative approaches are established as key elements of product variant production, but there is no single standard for the PLE domain. This work will investigate the relationship between concern and feature composition by addressing suitable mechanisms for composition. Specifically, it will look at the role *higher-order model transformations* may play in supporting this.

3 Research Conjectures.

The research undertaken focuses on composition of concerns in model-driven engineering (MDE) and how the semantics of models can be preserved in the process. This requires that means for reasoning about semantics of modelling abstractions and composition scenarios are established.

The following summarises the research conjectures:

1. Composition of structured classes with interactions, which preserves semantics, will strengthen preciseness, usability, and reuse in model-driven engineering.
   (a) Strengthening preciseness can be interpreted as increased confidence on the part of the developer that intended meanings are conveyed properly.
   (b) Strengthening usability means that it should be simpler to achieve reuse benefits and simpler to understand how to achieve them.
   (c) Two key questions in this regard are (i) *how can semantics preservation be defined, formally and pragmatically?* and (ii) *how can it be specified and analysed?*

2. Higher-order transformations (HOT) can describe and implement concerns/features and their composition more flexibly than corresponding first-order transformations.
   (a) Is there a specific class of concern-related problems where HOTs are particularly useful?
   (b) How can HOTs help preserve semantics?

4 Proposed Solution.

In order to facilitate composition of structure and interactions, I will define techniques based on existing UML constructs. These techniques must facilitate specification of concerns and semantics for their composition. I will also define the semantics preserving characteristics of these compositions.

A starting point are collaborations of interacting roles and their binding to other roles/classifiers. UML Collaboration provides a mechanism called Collaboration Use for binding roles. This mechanism, however, is limited to binding connectable elements to other connectable elements (such as parts). It does not allow for introduction of new parts, nor does it specify how behaviour is bound.

The initial focus is to explore extension options that allow for more complex compositions with increased preciseness. I will investigate approaches to support the composition of both structure and interactions, including model/graph transformation techniques. The applicability of higher-order transformations for supporting composition of concerns and features will also be explored.

Semantics of structural composition must address compatibility of types of ports and parts and connectivity of parts. Related work on semantics for composition of structural models (classes) based on algebraic operators are described in Herrmann et al [8]. Many other works also address this formally and informally.

Interaction composition must address the semantics of interaction modifications compared with an original intention. A model of interaction semantics must be selected. One option is to explore trace semantics [9] and how sub traces are preserved in compositions. Other options are preservation of contracts, or mapping to state machines and utilise state machine semantics [10].

5 Progress and Expected Contributions.

This work is still in its early stages. Results on composition of architectural models was presented at an AOSD workshop [11] and has been further reworked for MODELS 2007. The focus here was supporting architectural variance by refinement of connectors using architectural aspects. Work on composition of interactions is currently in progress, focusing on specification of concerns with collaborations and interactions and implementation of the compositions process.

Composition based on higher-order transformations is being explored and has been applied in the product line domain. A prototype implementing higher-order transformation as aspects extensions to an existing transformation language has been developed. An article describing the results will be published at SPLC 2007.

The expected contributions are summarised below:

- Techniques for describing structural and behavioural composition of UML specifications, which preserves semantics.
- Implementation of the composition processes.
- Pragmatic and formal definitions of semantics preservation.
- Analysis mechanisms for semantics preservation.
- Techniques and tools for composition using higher-order transformations.

6 Related Work

Klein et al [10] describe a semantic-based weaving algorithm for scenarios based on High-level Message Sequence Charts (HMSC). They define sequential composition of basic messages sequence charts and apply it for weaving an MSC aspect with a base model. Aspects are defined in terms of pointcut and advice MSCs. The approach allows matching sequences not syntactically detectable, such as looping. Matches found are either augmented, replaced, or removed based on the advice definition. They address composition based on a semantic model analysis, but do not address the semantics of the resulting compositions.

Reddy et al [12] describe an approach for support composition with aspects in UML interactions. Aspects are described as a combination of patterns specifications (IPS) introduced in [13] and special stereotypes for interaction fragments. They use tags (stereotypes) to instantiate those aspects in messages or fragments in primary models. An aspect adds messages before and after tagged elements and refines the messages that are part of a tag. [12] specifies that semantics should be maintained in refinement, but not how this is ensured, checked, or what it means.

Clarke and Walker [14] define composition patterns based on a subject-oriented design model. Here, interactions can be composed by using messages as templates. It does not address composition of complete behaviours, only instantiation of subject templates. Usage of templates, however, which is also addressed by IPS's[13], may be a useful mechanism to utilise for interactions, as this is a standard capability of standard UML interactions.

Stein et al [15] describe join point designation diagrams (JPDD), which defines a graphical notation for representing join point selections in UML, also for interactions. It extends standard notation with wildcards, parameterised names and call graph existence operators.

7 Research Methods and Evaluation.

Several example systems will be the basis for studying the mechanisms for composition and semantics preservation and a catalysts for discovering composition use cases. An essential requirement for compositions is semantics preservation; the definition of semantics preservation will therefore give substantial characteristics for analysis and comparison of results. Techniques and developed tools will contribute to validate the approach. Validity will be addressed further by applying the results on multiple example systems and comparing with other approaches. The results will also be trialled on case studies involving real end users, which will provide an external validation.

8 Conclusion

This paper has outlined the work on *Semantics Preserving Model Composition*. It has described the problem and focus area, which is how to preserve semantics in composition of model-based specifications. It addressed specifically how to compose structured classes with interaction behaviour and how semantics can be preserved. It further addresses the role of higher-order transformations in model composition.

Acknowledgement. This work is being pursued in the context of the SWAT project (Semantics-preserving Weaving - Advancing the Technology), funded by the Norwegian Research Council (project number 167172/V30).

References

1. Dijkstra, E.: On the Role of Scientific Thought. (1974)
2. Parnas, D.L.: On the Criteria To Be Used in Decomposing Systems into Modules. Communications of the ACM (1972)
3. Wirfs-Brock, R., Wilkerson, B., Wiener, L.: Designing Object-Oriented Software. Prentice Hall (1990)
4. Reenskaug, T., Wold, P., Lehne, O.A.: Working with Objects: The OORAM Software Engineering Method. Englewood Cliffs: Prentice Hall (1995)
5. Pohl, K., Bockle, G., van der Linden, F.: Software Product Line Engineering - Foundations, Principles, and Techniques. Springer (2005)
6. Ossher, H., Tarr, P.: Multi-Dimensional Separation of Concerns and The Hyperspace Approach. In: Proceedings of the Symposium on Software Architectures and Component Technology: The State of the Art in Software Development, Kluwer (2000)
7. Mezini, M., Ostermann, K.: Variability Management with Feature-oriented Programming and Aspects. In: SIGSOFT '04/FSE-12: Proceedings of the 12th ACM SIGSOFT Twelfth International Symposium on Foundations of Software Engineering, New York, NY, USA, ACM Press (2004) 127–136
8. Herrmann, C., Krahn, H., Rumpe, B., Schindler, M., S., V.: An Algebraic View on the Semantics of Model Composition. In: Third European Conference on Model Driven Architecture - Foundations and Applications. (2007)
9. Haugen, O., Husa, K.E., Runde, R.K., Stolen, K.: STAIRS Towards Formal Design with Sequence Diagrams. Software and Systems Modeling **4** (2005) 355 – 357
10. Klein, J., Helouet, L., Jezequel, J.: Semantic-based Weaving of Scenarios. In: AOSD '06: The 5th international conference on Aspect-oriented software development, New York, NY, USA, ACM Press (2006) 27–38
11. Oldevik, J., Haugen, Ø.: Architectural Aspects in UML. First Workshop on Aspects in Architectural Description, AOSD 2007 (2007)
12. Reddy, R., Solberg, A., France, R., Ghosh, S.: Composing Sequence Models using Tags. In: Aspect-Oriented Modelling Workshop at AOSD. (2006)
13. France, R.B., Kim, D.K., Ghosh, S., Song, E.: A UML-Based Pattern Specification Technique. IEEE Trans. Softw. Eng. **30**(3) (2004) 193–206
14. Clarke, S., Walker, R.J.: Composition Patterns: An Approach to Designing Reusable Aspects. International Conference of Software Engineering (ICSE) (2001)
15. Stein, D., Hanenberg, S., Unland, R.: Expressing Different Conceptual Models of Join Point Selections in Aspect-oriented Design. In: AOSD '06: Proceedings of the 5th international conference on Aspect-oriented software development, New York, NY, USA, ACM Press (2006) 15–26