A Basis for Feature-oriented Modelling in Event-B

Jennifer Sorge, Michael Poppleton, Michael Butler

Electronics and Computer Science, University of Southampton
{jhs06r,mrp,mjb}@ecs.soton.ac.uk

Abstract. Feature-oriented modelling is a well-known approach for Software Product Line (SPL) development. It is a widely used method when developing groups of related software. With an SPL approach, the development of a software product is quicker, less expensive and of higher quality than a one-off development since much effort is re-used. However, this approach is not common in formal methods development, which is generally high cost and time consuming, yet crucial in the development of critical systems. We present a method to integrate feature-oriented development with the formal specification language Event-B. Our approach allows the user to map a feature from the feature model to an Event-B component, which contains a formal specification of that feature. We also present some patterns, which assist the user in the modelling of Event-B components. We describe a composition process which consists of the user selecting an instance in the feature model and then constructing this instance in Event-B. While composing, the user may also discharge new composition proof obligations in order to ensure the model is consistent. The model is then constructed using a number of composition rules.

1 Introduction

Current critical systems have become more complex and more common, which requires them to be developed more efficiently and preferably with the application of formal methods to ensure a safer system. The development with formal methods is very time-consuming and costly, so in many cases formal methods are not used. In our work we use the formal method Event-B [1], which is based on first-order logic and set theory. It is structured into a dynamic part (describing system behaviour) and a static part (describing contant data and types). The dynamic part is referred to as a machine, and the static part is called a context. Event-B is supported by the open tool Rodin [1]. In non-critical systems, it is common to use SPL development techniques to save time and develop better software faster. One such approach is feature modelling [2], which is used to structure a set of related software products into common and variable requirements, referred to as features. The feature diagram can be used to generate

[1] The continued development of the Rodin toolset is funded by the EU research project ICT 214158 DEPLOY (www.deploy-project.eu).

instances of the software family; this is done by the selection of different features within the diagram.

Currently, the development of Event-B models is a time-consuming task, which requires a lot of proof. Our motivation is to reduce the development time and to considerably reduce reproof by reusing Event-B components for which proof obligations have been discharged. During the process of composition, composition proof obligations can be discharged. By experimentation we have shown that a lot of the original proof obligations do not have to be reproved during composition, thus saving a lot of prover resources.

An extended version of this paper is available from [3].

2 Process

In Figure 1 we present the composition process. The feature model is formed by features which may be associated with Event-B components. The composition process entails a subset of features to be selected from the feature model to form a feature model instance, thereby selecting several of these Event-B components. These components are composed pair-wise, and composition proof obligations can be discharged to prove properties and to ensure consistency of the composition. The final Event-B machine represents the formal specification which is associated with the feature model instance and is obtained by composing these components.

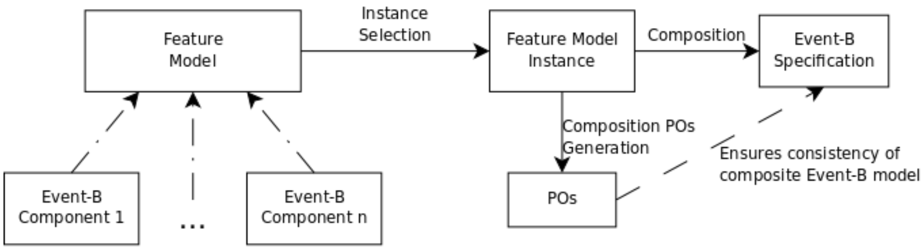

Fig. 1. Overview of Composition Process

2.1 Integration of Feature Models and Event-B

In order to link feature models with Event-B, we develop Event-B components which represent features in a feature model and that can be linked to them by name. Only leaf node features can be mapped to an Event-B component. We have developed a number of Event-B modelling patterns which provide guidelines that help the construction of single Event-B components. An Event-B component may consist of zero or more contexts, but must be consistent and independent of any other components. Refinement is also currently not supported. All proof obligations for a component must be discharged.

2.2 Proof and Composition

Composition of components is n-wise, if more than two features that are mapped to components are selected in the feature model. The composition process, however is pair-wise. This means that always two components are composed. All proof obligations for a component have been discharged during the creation of a component, however, when composing two components, new proof obligations may come up. We will refer to these as composition proof obligations. Proof obligations that are only concerned with one component are referred to as component proof obligations.

Composition proof obligations are proof obligations that can be discharged during composition of two components. They are based on component proof obligations and their task is to reduce reproof of component proof obligations.

Once composition proof obligations have been discharged, the two components can be composed. We have developed a number of composition rules that can be applied to the composition of Event-B contexts and machines. The outcome of composition is one single Event-B model.

3 Conclusion and Future Work

Our work demonstrates the integration of feature models and Event-B, thus enabling SPL development for formal methods and providing a way to prove certain properties about a composition. Currently our work is fundamentally theoretical, however we have been collaborating in the development of a Rodin plugin to integrate this theoretical approach with the Rodin platform. This plugin contains a composition tool, and in future will support feature model editing and an instance generator[4, 5].

References

1. Abrial, J.R.: Modeling in Event-B: Systems and Software Engineering. To be published by Cambridge University Press (2009)
2. Pohl, K., Böckle, G., van der Linden, F.: Software Product Line Engineering: Foundations, Principles, and Techniques. Springer (2005)
3. Sorge, J., Poppleton, M., Butler, M.: A Basis for Feature-oriented Modelling in Event-B. (2009) http://eprints.soton.ac.uk/69645/.
4. Poppleton, M., Fischer, B., Franklin, C., Gondal, A., Snook, C., Sorge, J.: Towards Reuse with"Feature-Oriented Event-B". (2008)
5. Gondal, A., Poppleton, M., Snook, C.: Feature composition-towards product lines of event-B models. (2009)