

Supporting Change in Product Lines within the Context of Use Case-Driven Development and Testing

Ines Hajri
(Supervisors: Arda Goknil and Lionel C. Briand)
SnT Centre for Security, Reliability and Trust
University of Luxembourg, Luxembourg
ines.hajri@uni.lu

ABSTRACT

Product Line Engineering (PLE) is becoming a common practice in industry to enhance product quality, to reduce development costs, and to improve time-to-market. At the same time, many development contexts are use case-driven and this strongly influences their requirements engineering and system testing practices. In this PhD project, we aim to achieve automated and effective change management in a product family within the context of use case-driven development and system testing. To this end, we first provide a modeling method for capturing variability information explicitly in Product Line (PL) use case and domain models. Then, we propose an automated configuration approach to automatically generate Product Specific (PS) use case and domain models from PL models. In addition, we plan to provide a change impact analysis approach for PL use case and domain models and automated regression test selection for system test cases derived from PL use case models.

CCS Concepts

•Software and its engineering → Software product lines;

Keywords

Product Line Engineering, Use Case-Driven Development, Change Impact Analysis, Regression Test Selection

1. INTRODUCTION AND RESEARCH PROBLEM

Product Line Engineering (PLE) is a crucial practice in many software development environments where systems are complex and developed for multiple customers with varying needs. At the same time, many business contexts are use case-driven where use cases are the main artifacts driving requirements elicitation and many other development activities [29]. This is also the case for our industrial partner

IEE S.A. [1], a leading supplier of embedded systems in the automotive domain. This PhD project is motivated by the discussions with IEE which aims to adopt PLE in its software development practice. The current development practice at IEE is use case-driven and based on *clone-and-own reuse* [10]. IEE starts a new product with an initial customer. IEE analysts elicit requirements as use case and domain models. Then, they derive system test cases from the use case models for the initial customer. For each new customer of the product, the IEE analysts need to clone the current models, and negotiate variabilities with the customer to produce new use case and domain models, and to derive and select new system test cases from the updated use cases (i.e., change management for use cases and regression test selection for system test cases). With such practice, the IEE analysts loose track of commonalities and variabilities across products. They, together with the customer, need to evaluate the entire use cases, domain model and test cases.

The *clone-and-own reuse* practice is fully manual, error-prone and time-consuming in industrial settings, which leads to ad-hoc change management for use cases, domain models and test cases in the context of product lines since the variability information is not explicitly represented.

The need for supporting PLE in the context of use case-driven development and regression test selection has been already acknowledged in the literature [4, 13, 12, 14, 31]. Most of the existing approaches introduce feature modeling into practice, including establishing and maintaining traces between feature models and use case models [16, 15, 19, 20, 11, 3]. Due to limited resources, many software development companies find such additional traceability and maintainability effort to be impractical. In addition, most approaches do not provide automated support for use case evolution in a product family in terms of change impact analysis and system regression testing.

We aim, in this PhD project, to provide automated support for the change management of use case models and regression test selection in a product family.

2. RELATED WORK

Many proposed use case-driven configuration approaches [5, 6, 16, 18, 17, 2] require that feature models be traced to use case diagrams and specifications. The analysts need to establish traces between feature models and use cases. The evolution of feature models and use cases also requires these traces to be maintained manually [7]. There are approaches [32, 27, 9, 25] that study the evolution of feature models but they do not address the impact of changes of

variability information on use case models. In addition, in many software development environments, such as traceability and modeling overhead may not be acceptable, as it is the case at IEE. Most of the regression testing approaches in the literature focus on changes in design [8, 28], and source code [24, 26]. In contrast, our PhD project is focused on system test cases where requirements changes are regression tested within the context of product lines.

3. PROPOSED SOLUTION

Our goal is to provide approaches and their accompanying tools to achieve automated, effective change management and regression test selection for a family of products within the context of use case-driven development and testing.

To accomplish this goal, our proposal is divided into four pillars: (1) a modeling method in which variability information is explicitly represented in Product Line (PL) use case diagrams, specifications and domain models without any feature model, (2) an automated configuration approach to generate Product Specific (PS) use case and domain models from PL models and configuration decisions, (3) a change impact analysis approach for evolving configuration decisions and for evolving PL models, and (4) an automated regression test selection approach for a family of products.

4. EXPECTED CONTRIBUTIONS

A product line use case modeling method. Our motivation is to rely, to the largest extent possible, on common practices to achieve widespread applicability and to minimize the modeling overhead. The method integrates and builds on existing work and captures variability in PL use case and domain models at a level of granularity enabling both precise communication and guidance for the product configuration. It needs to be supported by a tool which automatically confirms the consistency of PL use case and domain models.

A configuration approach for use case-driven development. In order to facilitate use case-driven configuration in industrial practice, the configuration approach should provide a high degree of automation while the analysts are interactively guided for their decisions. With the interactive guidance and the proper tool support, the analysts can make decisions and resolve decision contradictions, which leads to the automatic generation of PS use case and domain models.

A change impact analysis approach for PL use case and domain models. Change can occur both in configuration decisions and variability aspects of PL models. For decision changes in a product, the impact on other decisions needs to be assessed and the reconfiguration should be considered in the PS models. To this end, we plan to develop a change impact analysis approach to identify impacted configuration decisions and parts of PS models that need to be reconfigured when the PL models and the configuration decisions evolve.

A regression test selection approach for product lines. Regression test selection is a particular application of change impact analysis, that consists in choosing, from an existing test set, test cases that can and need to be rerun to ensure existing, unmodified functionalities are still working correctly [8]. When requirements evolve in a product family, the change impact on the execution of system test cases derived from these requirements need to be assessed. We plan to provide an automated regression test selection ap-

proach for system test cases derived from use case models. For system test cases and their traces to use case models, we plan to rely on the Use Case Modeling for System Tests Generation approach (UMTG) [29, 30], that automatically generates executable system test cases from PS use case and domain models.

5. CURRENT RESULTS

At this stage, the first two pillars have been achieved.

Product line use case modeling method. We proposed, applied, and assessed the Product line Use case modeling Method (PUM) to support variability modeling in PL use case diagrams, specifications, and domain models [21], without making use of feature models, thus avoiding unnecessary modeling overhead. PUM adopts existing PL extensions for use case diagrams [23] and domain models [34]. For modeling variability in use case specifications, we introduced new product line extensions for the Restricted Use Case Modeling method (RUCM) [33]. PUM is supported by a tool relying on Natural Language Processing (NLP) to check the consistency of PL use case and domain models.

Use case-driven configuration. We proposed, applied, and assessed a use case-driven configuration approach based on our modeling method (PUM) [22]. The approach supports three activities. First, the analyst is guided to make configuration decisions in an appropriate order. Second, the consistency of configuration decisions is ensured by automatically identifying contradicting decisions. Third, PS use case and domain models are automatically generated from PL models and configuration decisions. Our approach is supported by a tool, PUMConf, relying on NLP and integrated with IBM DOORS.

6. PLAN FOR EVALUATION AND VALIDATION

We plan to assess the proposed approaches and tools in terms of feasibility, adoption effort, expressiveness, and comparison with current practice. To this end, we rely on reporting industrial case studies and questionnaire studies in close collaboration with our industrial partner IEE. Moreover, we have delivered tutorials and plan to continue to do so in order to help IEE adopt the proposed approaches and the accompanying tools in their development context.

The evaluation started with the first two pillars of our research program. For instance, we evaluated the feasibility of our configuration approach in an industrial context via reporting (i) an industrial case study, i.e., Smart Trunk Opener (STO) developed by IEE, to demonstrate its feasibility and (ii) the results of a questionnaire based survey at IEE to investigate how the configuration tool is perceived to improve industrial practice. Results from the questionnaires suggest that our approach is practical and beneficial to capture variability and to configure PS use case and domain models in industrial settings.

We further plan to conduct an empirical study for our regression test selection approach and a user study to assess the practicality and usability of our tools.

Acknowledgments

I gratefully acknowledge my PhD supervisors, Lionel C. Briand and Arda Goknil, for their guidance and support. I acknowledge financial support from IEE and FNR under grants FNR/P10/03 and FNR10045046.

7. REFERENCES

- [1] IEE (International Electronics & Engineering) S.A., <http://www.iee.lu/>.
- [2] pure::variants, http://www.pure-systems.com/pure_variants.49.0.html.
- [3] M. Alf erez, J. Santos, A. Moreira, A. Garcia, U. Kulesza, J. Ara ujo, and V. Amaral. Multi-view composition language for software product line requirements. In *SLE'09*, pages 103–122, 2009.
- [4] V. Alves, N. Niu, C. Alves, and G. Valen a. Requirements engineering for software product lines: A systematic review. *Information and Software Technology*, 52:806–820, 2010.
- [5] R. Bonif acio and P. Borba. Modeling scenario variability as croscutting mechanisms. In *AOSD'09*, pages 125–136, 2009.
- [6] R. Bonif acio, P. Borba, and S. Soares. On the benefits of scenario variability as croscutting. In *EA-AOSD'08*, pages 1–6, 2008.
- [7] G. Botterweck and A. Pleuss. Evolution of software product lines. *Evolving Software Systems*, 2014.
- [8] L. C. Briand, Y. Labiche, and S. He. Automating regression test selection based on UML designs. *Information and Software Technology*, 51(1):16–30, 2009.
- [9] J. Burdek, T. Kehrler, M. Lochau, D. Reuling, U. Kelter, and A. Schurr. Reasoning about product-line evolution using complex feature model differences. *Automated Software Engineering*, 2015.
- [10] P. Clements and L. Northrop. *Software Product Lines: Practices and Patterns*. Addison-Wesley, 2001.
- [11] K. Czarnecki and M. Antkiewicz. Mapping features to models: A template approach based on superimposed variants. In *GPCE'05*, pages 422–437, 2005.
- [12] E. Engstrom. Regression test selection and product line system testing. In *ICST'10*, pages 512–515, 2010.
- [13] E. Engstrom and P. Runeson. Software product line testing - a systematic mapping study. *Information and Software Technology*, 53:2–13, 2011.
- [14] E. Engstrom, P. Runeson, and M. Skoglund. A systematic review on regression test selection techniques. *Information and Software Technology*, 52(1):14–30, 2010.
- [15] M. Eriksson, J. Borstler, and A. Asa. Marrying features and use cases for product line requirements modeling of embedded systems. In *SERPS'04*, pages 73–82, 2004.
- [16] M. Eriksson, J. Borstler, and K. Borg. The pluss approach - domain modeling with features, use cases and use case realizations. In *SPLC'05*, pages 33–44, 2005.
- [17] M. Eriksson, J. Borstler, and K. Borg. Managing requirements specifications for product lines - an approach and industry case study. *Journal of Systems and Software*, 82:435–447, 2009.
- [18] M. Eriksson, H. Morast, J. Borstler, and K. Borg. The pluss toolkit - extending telelogic doors and ibm-rational rose to support product line use case modeling. In *ASE'05*, pages 300–304, 2005.
- [19] A. Fantechi, S. Gnesi, I. John, G. Lami, and J. Dorr. Elicitation of use cases for product lines. In *PFE'03*, pages 152–167, 2004.
- [20] A. Fantechi, S. Gnesi, G. Lami, and E. Nesti. A methodology for the derivation and verification of use cases for product lines. In *SPLC'04*, pages 255–265, 2004.
- [21] I. Hajri, A. Goknil, L. C. Briand, and T. Stephany. Applying product line use case modeling in an industrial automotive embedded system: Lessons learned and a refined approach. In *MODELS'15*, pages 338–347, 2015.
- [22] I. Hajri, A. Goknil, L. C. Briand, and T. Stephany. Configuring use case models in product families. *Software & Systems Modeling*, pages 1–33, 2016.
- [23] G. Halmans and K. Pohl. Communicating the variability of a software-product family to customers. *Software & Systems Modeling*, 2:15–36, 2003.
- [24] M. J. Harrold, J. A. Jones, T. Li, D. Liang, A. Orso, M. Pennings, S. Sinha, S. A. Spoon, and A. Gujarathi. Regression test selection for java software. In *OOPSLA'01*, pages 312–326, 2001.
- [25] A. Pleuss, G. Botterweck, D. Dhungana, A. Polzer, and S. Kowalewski. Model-driven support for product line evolution on feature level. *Journal of Systems and Software*, 85:2261–2274, 2012.
- [26] G. Rothermel, M. J. Harrold, and J. Dedhia. Regression test selection for c++ software. *Software Testing, Verification and Reliability*, 10(2):77–109, 2000.
- [27] T. Thum, D. Batory, and C. Kastner. Reasoning about edits to feature models. In *ICSE'09*, pages 254–264, 2009.
- [28] Y. L. Traon, T. Jeron, J.-M. Jezequel, and P. Morel. Efficient object-oriented integration and regression testing. *IEEE Transactions on Reliability*, 49(1):12–25, 2000.
- [29] C. Wang, F. Pastore, A. Goknil, L. C. Briand, and M. Z. Z. Iqbal. Automatic generation of system test cases from use case specifications. In *ISSTA'15*, pages 385–396, 2015.
- [30] C. Wang, F. Pastore, A. Goknil, L. C. Briand, and M. Z. Z. Iqbal. UMTG: a toolset to automatically generate system test cases from use case specifications. In *ESEC/SIGSOFT FSE'15*, pages 942–945, 2015.
- [31] S. Wang, S. Ali, A. Gotlieb, and M. Liaaen. Automated product line test case selection: Industrial case study and controlled experiment. *Software & Systems Modeling*, 2015.
- [32] J. White, J. A. Galindo, T. Saxena, B. Dougherty, D. Benavides, and D. C. Schmidt. Evolving feature model configurations in software product lines. *Journal of Systems and Software*, pages 119–136, 2014.
- [33] T. Yue, L. C. Briand, and Y. Labiche. Facilitating the transition from use case models to analysis models: Approach and experiments. *ACM Transactions on Software Engineering and Methodology*, 22(1), 2013.
- [34] T. Ziadi and J.-M. Jezequel. Product line engineering with the uml: Deriving products. In *Software Product Lines*. Springer, 2006.