Deriving Usage Model Variants for Model-based Testing: An Industrial Case Study

Hamza Samih, Mathieu Acher, Ralf Bogusch, Hélène Le Guen, Benoit Baudry

Deriving Usage Model Variants for Model-based Testing: An Industrial Case Study

Hamza Samih
and Hélène Le Guen
ALL4TEC, France
{hamza.samih, helene.leguen}@all4tec.net

Ralf Bogusch
Airbus Defence & Space, Germany
ralf.bogusch@cassidian.com

Mathieu Acher
and Benoit Baudry
IRISA/Inria Rennes, France
mathieu.acher@irisa.fr, benoit.baudry@inria.fr

***Abstract*—The strong cost pressure of the market and safety issues faced by aerospace industry affect the development. Suppliers are forced to continuously optimize their life-cycle processes to facilitate the development of variants for different customers and shorten time to market. Additionally, industrial safety standards like RTCA/DO-178C require high efforts for testing single products. A suitably organized test process for Product Lines (PL) can meet standards. In this paper, we propose an approach that adopts Model-based Testing (MBT) for PL. Usage models, a widely used MBT formalism that provides automatic test case generation capabilities, are equipped with variability information such that usage model variants can be derived for a given set of features. The approach is integrated in the professional MBT tool MaTeLo. We report on our experience gained from an industrial case study in the aerospace domain.**

***Keywords*—Product Line, Model-based Testing, Usage Model, Orthogonal Variability Model, Requirements**

I. Introduction

Mass production is now more than a standard in the industry. Numerous organizations engineer a set of related systems that share generic features (characteristics) while having more specific ones. *Product Line Engineering* (PLE) aims at developing such families of products, leveraging component reuse, allowing shorter time-to-market, reducing development cost while taking into account the *variability* and diversity of users and customers [1]–[4]. One significant concern in PLE is related to the combinatorial explosion of configuring or testing large product lines [3], [5]–[7].

Model-based Testing (MBT) aims at inferring test cases from a test model that is based on the functional requirements [8]. A test model can be represented using several formats such as UML state-machines, usage model, or even Markov chains. From this test model, one can define different testing strategies and derive a set of relevant test cases [6], [9]. Although behavioural MBT is well established for single-system testing, a survey shows insufficient support of PL-based MBT [10]. Specifically, *usage models*, a widely used formalism in MBT, are employed to test only one individual system. The lack of variability support for usage models precludes their exploitations in industrial settings.

The main idea of our proposal is to establish formal correspondences between features, requirements and a usage model. We equip usage models with *variability* information in order to formally document what can vary in a usage model, and in addition to systematize the derivation of usage models variants – each variant-specific usage model being exploited afterwards for generating test cases of the variant of a PL. We develop an extension of MaTeLo (a professional MBT tool) to support engineers in deriving usage model variants.

We evaluate our approach in an industrial case study in the aerospace domain. Practitioners report a reduction of the cost for test case development and highlight the minimal invasiveness of the solution so that established requirements and usage models can be reused. We also learned that the variability methodology impacts the way requirements are specified and requires a basic training before adoption.

The remainder of the paper is organized as follows. Section II motivates our work and introduces background information. Section III reports on an application of the approach in an industrial setting. Section IV concludes the paper.

II. Context & background

Our approach consists of adapting MBT for product lines. Model-Based Testing helps to generate automatically test cases from requirements. In this work we use MaTeLo[1] usage models (roughly a Markov chain test model) [11] to represent the possible usage scenarios of a system under test (SUT) [12]–[14]. MaTeLo is a Model-based Testing solution, which allows automated generation of test cases for complex systems. The complete formalization of a usage model can be found in [14]. In this paper, we add variability means to usage models in order to allow deriving usage model variants for a set of related products from a single PL usage model. The variability of a PL is represented by an Orthogonal Variability Model (OVM) [1].

However, the variability model is separated from the test artefacts. During the realization of the usage model, functional requirements are associated with transitions of the model. In order to support the derivation of usage model variants, we additionally relate functional requirements with the features of the variability model [15]. In this way correspondences between features and transitions of the usage model can be established.

For PL testing, we configure a set of variants according to the OVM model. We propose an algorithm to derive usage model variants automatically. The derivation of a variant consists of selecting requirements related to each feature that compose the variant. Subsequently, we select all transitions to be kept, utilizing the requirements attached to the transitions.

[1] Markov Test Logic, a MBT tool developed by ALL4TEC

Finally, we prune the usage model according to the identified requirements and transitions yielding a valid variant-specific usage model which is suitable for automatic generation of test cases [11].

III. Case Study

A. Realization

We have implemented the presented approach in an Eclipse-based prototype tool called MaTeLo Product Line Manager (MPLM). MPLM extends the MaTeLo tool suite with generation of test models for a product line. An experimental case study was performed with the industrial partner Airbus Defence & Space in the frame of the ARTEMIS Joint Undertaking research project MBAT[2] in order to validate the approach from an industrial point of view.

B. Industrial needs

The goal of this research is to provide a solution that allows generating test cases for a product line and can easily be adopted in an industrial environment. The main industrial needs related with this goal are:

- Reduce test cost
- Improve product quality
- Minimize cost of deployment

The following questions are derived from the above needs and constitute the basis for the assessment performed in the case study: (1) Can we reduce cost for test case development? (2) Can we reuse test artefact across product variants? (3) Can we reduce test cost resulting from changes in requirements? (4) Can we reduce the number of defects present in product variants? (5) What is the additional effort required to setup the tool environment and to provide training to the test team?

C. Case study in a nutshell

Airbus Defence & Space develops avionic systems that support helicopter pilots in degraded visual environments which can be caused by rain, fog, sand, dust and snow. In this case study we employ the landing symbology function which is part of the situational awareness suite Sferion™. The landing symbology function enables the pilot to mark the intended landing position on ground using a head-tracked HMS/D (Helmet Mounted Sight and Display) and HOCAS (Hands On Collective And Stick). During the final landing approach the spatial awareness of flying crews is enhanced by displaying 3D conformal visual cues on the HMS/D. Additionally, obstacles residing in the landing zone can be detected and classified using a real-time OWS (Obstacle Warning System). The situational awareness suite Sferion™ constitutes a product line. Different features can be selected for the landing symbology function depending on the customer and the helicopter platform to which the solution shall be deployed.

First, we identify relevant product features and potential configurations based on business and market information in order to define the *PL scope*. The results are captured in a *variability model* using the OVM approach and the Vedit[3] tool which is integrated in MPLM (see Fig. 1). Features such as "Mark landing position" are introduced with dependencies to mandatory (e.g. "Check for no ground") and optional (e.g. "Check for obstacles") features.

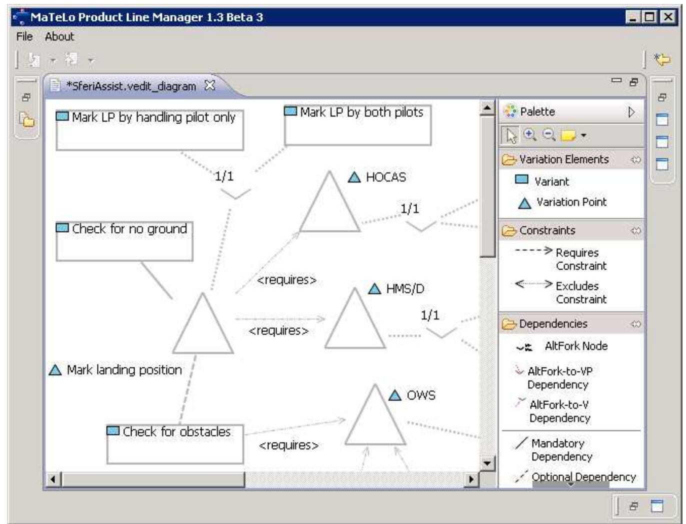

Fig. 1: OVM model of Sferion™

A set of *PL requirements* covering all features of the PL is created with the requirements management tool IBM Rational DOORS. The requirements are imported into the MaTeLo usage model editor and a *PL usage model* is created (see Fig. 3). The requirements are assigned to the respective transitions of the usage model. In addition, test stimuli and expected behaviours are defined in the usage model completing the usage model specification.

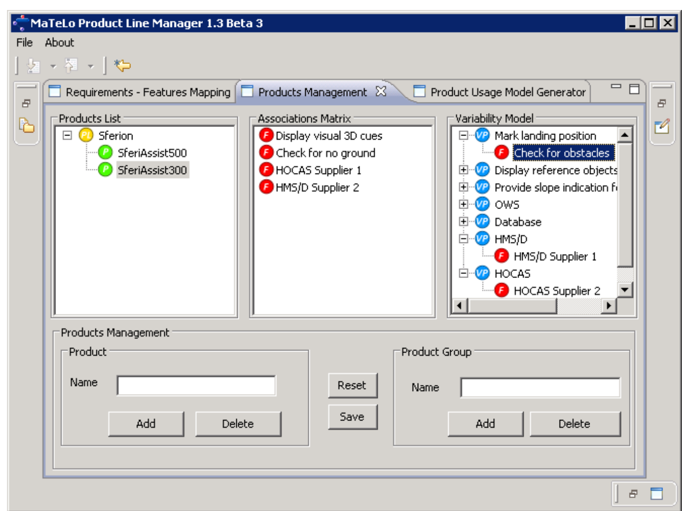

Fig. 2: Configure products in MPLM

With the help of MPLM the *requirements - feature association* is defined based on both the PL usage model and

[2] Combined Model-based Analysis and Testing of Embedded Systems, http://www.mbat-artemis.eu

[3] http://www.sse.uni-essen.de/varmod

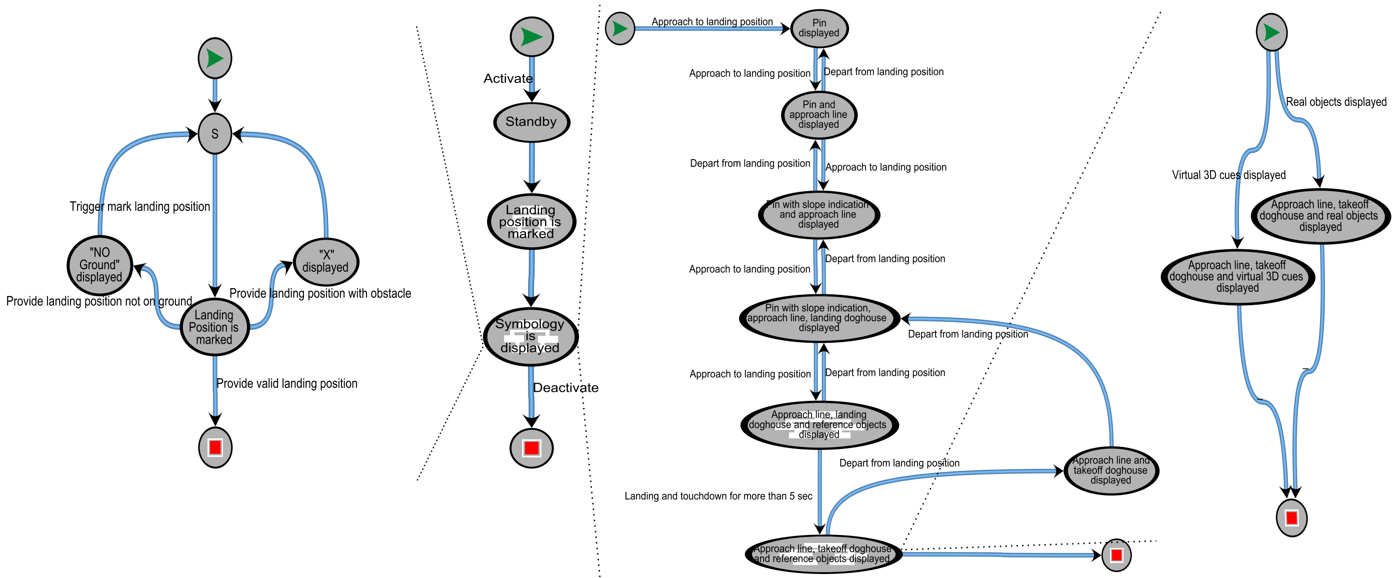

Fig. 3: Usage model of Sferion™

the OVM model. Then, *configurations* are created by selecting the features which should be present in the desired product. During the configuration dependencies and constraints between features are respected. According to Figure 2 two different products are defined: SferiAssist500 (high-end product) and SferiAssist300 (low-end product). The optional feature "Check for obstacles" is only present in the high-end product.

Finally, the MPLM transforms the PL usage model into *usage model variants* for the desired products (see Fig. 4). The MaTeLo Testor tool is used for the generation of *test suites* from derived usage model variants.

The experimental study comprises 12 system requirements and 16 features. The PL usage model provides 25 states, 45 transitions, and 5 chain instances. MPLM generates valid usage model variants with an average generation time of less than one second on a standard PC equipped with a 2.8 GHz processor and 8 GB RAM.

D. Assessment

The questions derived from the industrial needs have been assessed based on the experience gained from the realization of the case study.

1) Reduction of cost for test case development: Model-based Testing (MBT) is one key enabler for test cost reduction. After creation of the usage model, efforts for generating test cases are minimized due to the automation. During the case study we performed a measurement program in order to measure the efficiency gain for test case development. We selected a reference project from the avionic domain that did not apply MBT. Interviews with test managers were conducted to get the efforts spent during test case development. The number of requirements covered by the test cases was counted and normalized using weighting factors which allows for requirements of different complexity. During the case study we measured the efforts for setting up the usage model and the test automation. The normalized number of requirements was determined. The efficiency of test case development, i.e. the average effort needed to cover a normalized requirement by test, was calculated for both the reference project and the case study. The comparison of the efficiency yielded a cost reduction of 18%.

2) Reuse of test artefact: In the past, separate projects were setup for different customers and consequently test case design was done independently for all product variants without systematic reuse. Now, the usage model is the primary artefact that is reused across the PL. Moreover, we manage reuse on the usage model level, rather than on the level of test cases. The advantage is that much less artefacts need to be managed for the PL. However, in order to avoid unnecessary iterations it is mandatory to have a sound understanding of the variabilities a priori. This is to well structure the requirements and the usage model according to the variabilities and hence facilitate the transformation of the PL usage model into usage model variants. For example, requirements have to be written in a way that allows to clearly associate them with features. Usage models should be developed such that variants can be easily derived (e.g. using separate paths representing variable behaviour).

3) Cost of deployment: The approach is mainly based on already established artefacts like requirements and usage models. Only the OVM approach is added. The complexity of linking variability models and usage models is minimized since requirements - features associations are used. All tool extensions are provided by a single add-on for the MaTeLo tool environment. Thus, the additional effort for tool deployment is negligible. Regarding training we recommend to provide basic training on PL engineering in order to make the team aware of potentially new topics like PL scoping and variability modelling.

4) Avenues for improvements: We learned that the approach opens avenues for possible extensions and more research.

Reduction of number of defects. Our approach is to perform tests for all products. Different test case generation strategies may be employed in product variants yielding different test cases. Hence the coverage of test criteria is increased and consequently the opportunity to detect faults. The approach could be extended to support analysis of test results. For example, if a test fails in one product, a hint may be given to the PL and other derived products in order to avoid unnecessary tests until the fault is removed.

Support for changes. In a realistic environment requirements are subject to changes. Based on the traceability between requirements and usage model elements, the impact of requirements changes can be analysed. Currently, there is no separation in the usage model variant between the part that is derived from the PL usage model and the product-specific part. This implies that the change impact needs to be analysed in each usage model variant; incremental testing strategies can thus be considered (see, e.g., [16]).

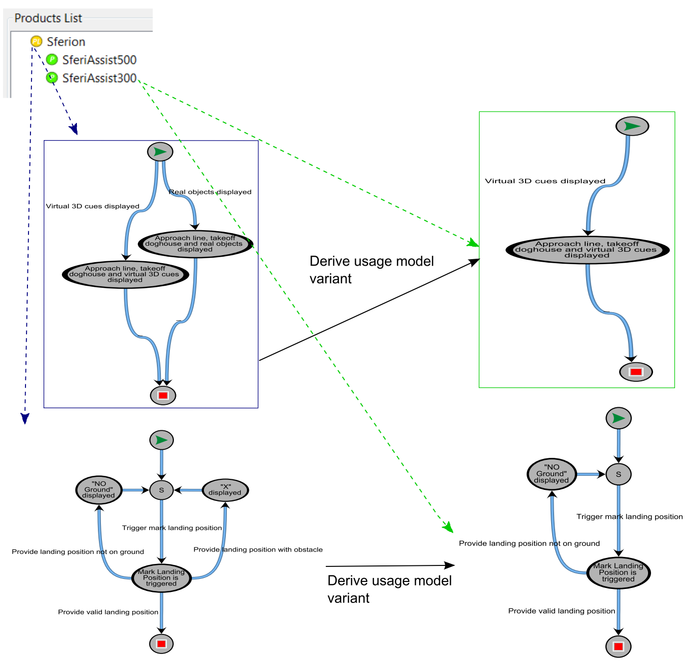

Fig. 4: Derive usage model variants

IV. Conclusion

We presented a tooled supported approach that enables model-based testing in the frame of product line engineering. Test cases can automatically be generated not only for one product, but for many variants according to the specified features. The proposed solution adds variability information to usage models. The variability is described in a separate OVM model. The features of the variability model are associated with the functional requirements of the system under test. Then, variants can be configured by selecting relevant features according to the OVM model. The corresponding usage model variants are automatically derived utilizing the associations between features and functional requirements.

The solution is implemented in the MPLM tool that extends the professional model-based testing tool MaTeLo. We reported on our industrial case study in the aerospace domain that was performed with the industrial partner Airbus Defence & Space in the frame of the ARTEMIS Joint Undertaking research project MBAT. The evaluation of the approach showed that the test cost for product line testing is significantly reduced due to the managed reuse of test artefacts. The deployment is facilitated based on the fact that the complexity of the solution is minimized by keeping the OVM model separate from already established test artefacts.

We are currently working to integrate MPLM in the new MaTeLo tool platform based on Eclipse RCP and will continue applying the approach to other industrial case studies in other domains.

Acknowledgments. The research leading to these results has received funding from the ARTEMIS Joint Undertaking under grant agreement no. 269335 (ARTEMIS project MBAT), French DGCIS and German BMBF.

References

[1] K. Pohl, G. Böckle, and F. J. v. d. Linden, *Software Product Line Engineering: Foundations, Principles and Techniques*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[2] J. Bartholdt, R. Oberhauser, and A. Rytina, "An approach to addressing entity model variability within software product lines," in *Software Engineering Advances, 2008. ICSEA '08*.

[3] A. Classen, P. Heymans, P.-Y. Schobbens, A. Legay, and J.-F. Raskin, "Model checking lots of systems: efficient verification of temporal properties in software product lines," in *ICSE'10*.

[4] M. Acher, P. Collet, P. Lahire, and R. B. France, "Familiar: A domain-specific language for large scale management of feature models," *Science of Computer Programming (SCP)*, vol. 78, no. 6, pp. 657–681, 2013.

[5] I. Cabral, M. B. Cohen, and G. Rothermel, "Improving the testing and testability of software product lines," in *SPLC'10*.

[6] X. Devroey, M. Cordy, G. Perrouin, E.-Y. Kang, P.-Y. Schobbens, P. Heymans, A. Legay, and B. Baudry, "A vision for behavioural model-driven validation of software product lines," in *ISoLA'12*.

[7] T. Thüm, S. Apel, C. Kästner, I. Schaefer, and G. Saake, "A classification and survey of analysis strategies for software product lines," *ACM Computing Surveys*, 2014.

[8] M. Utting and B. Legeard, *Practical Model-based Testing*. Morgan-Kaufmann, 2007.

[9] H. Le Guen and T. Thelin, "Practical experiences with statistical usage testing," in *Software Technology and Engineering Practice, 2003*.

[10] E. Engström and P. Runeson, "Software product line testing - a systematic mapping study," *in Inf. Softw. Technol'11*.

[11] A. Feliachi and H. Le Guen, "Generating transition probabilities for automatic model-based test generation," in *ICST*, 2010, pp. 99–102.

[12] J. Zander-Nowicka, *Model-Based Testing of Real-Time Embedded Systems in the Automotive Domain*. Fraunhofer-IRB-Verlag, 2009.

[13] J. G. Kemeny and J. L. Snell, *Finite Markov chains*. Van Nostrand Reinhold, 1960.

[14] H. Le Guen, R. Marie, and T. Thelin, "Reliability estimation for statistical usage testing using markov chains," in *ISSRE'04*.

[15] H. Samih, "Relating variability modeling and model-based testing for software product lines testing," in *ICTSS'12*, Aalborg, Denmark.

[16] M. Lochau, I. Schaefer, J. Kamischke, and S. Lity, "Incremental model-based testing of delta-oriented software product lines," in *TAP'12*.