Modelling Adaptability and Variability in Requirements

Phil Greenwood,
Ruzanna Chitchyan, Awais Rashid
SCC, Lancaster University
Lancaster, UK
chitchyan, greenwop, marash
@comp.lancs.ac.uk

Joost Noppen
School of Computing Sciences
University of East Anglia
Norwich, UK
j.noppen@uea.ac.uk

Franck Fleurey, Arnor Solberg
SINTEF
Oslo, Norway
franck.fleurey, arnor.solberg@sintef.no

***Abstract*—The requirements and design level identification and representation of dynamic variability for adaptive systems is a challenging task. This requires time and effort to identify and model the relevant elements as well as the need to consider the large number of potentially possible system configurations. Typically, each individual variability dimension needs to identified and modelled by enumerating each possible alternative. The full set of requirements needs to be reviewed to extract all potential variability dimensions. Moreover, each possible configuration of an adaptive system needs to be validated before use. In this demonstration, we present a tool suite that is able to manage dynamic variability in adaptive systems and tame such system complexity. This tool suite is able to automatically identify dynamic variability attributes such as variability dimensions, context, adaptation rules, and soft/hard goals from requirements documents. It also supports modelling of these artefacts as well as their run-time verification and validation.**

***Index Terms*—adaptive systems, feature models, natural language processing, modelling;**

I. Overview

This demonstration presents a new tool supported methodology, developed in the DiVA project[1], for constructing and managing requirements and design time models for adaptive systems. Not only do the tools support the construction of these models, but they also simulate and validate configurations that may occur due to run-time adaptation. The main challenges addressed by these tools are:

1) Increasing productivity and quality in adaptive system development;
2) Managing the combinatorial explosion of artefacts of such systems;
3) Supporting requirements and design time validation of adaptation policies and configurations.

The following subsections discuss the nature of these challenges and then describe the DiVA approach to tackling them.

II. Challenges

A. Challenge 1: Efficient construction of adaptive systems

It is essential for developers to efficiently cope with the complexity of adaptive systems. In terms of requirements engineering, adaptive systems and their adaptation potential are normally represented by enumerating the variability of their problem domain and potential solutions (e.g, AND/OR-decompositions of goals, feature trees [5], [8], user profile vectors [9], and frame elements [1], [7], [6]). The specific adaptation requirements are normally defined as constraints on the above models. However, even such tasks require significant effort due to, for instance, the need to identify and understand relevant parts of the text from large requirements documents.

B. Challenge 2: Handling Combinatorial Explosion

One of the key challenges when developing a system with dynamic variability is the combinatorial explosion of artefacts. The combinatorial explosion comes from having variability present throughout the development at variation points. This leads to a possible explosion of several adaptive system artefacts, such as, configurations, variant dependencies, and adaptation rules [3]. Moreover, this number can grow even when the system has already been deployed, because requirements can change for corrective or preventive maintenance. The exponential growth of artefacts in systems with dynamic variability presents a major problem. During requirements engineering, the combinatorial explosion makes adequate analysis of all possible requirement combinations in practical due to excessive time/effort demanded. At design-time it makes it impossible to validate and test all possible configurations.

C. Challenge 3: Target Configuration Validation

The best possible way to validate an adaptive system is to validate all the possible configurations and transitions of the system [2], [10], [4]. However, this implies specifying, by hand, all the possible configurations and all the transitions of the adaptive system to validate. Yet, this is impossible for systems with significant variability due to the combinatorial explosion of the number of artefacts (as per the above subsection). Previously, the choice of particular configurations has not been considered an RE problem. Instead, target configuration validation has been discussed in terms of static evaluation mechanisms (e.g., goal satisfaction via goal decomposition trees [8]). Yet, to validate requirements for run-time adaptation, it is necessary to maintain a casual connection between

[1]The DiVA Project website: http://www.ict-diva.eu

requirements and the run-time system representation. To date, this has not been addressed in RE research.

III. DiVA Approach

The DiVA approach provides a new tool supported methodology for construction of adaptive systems. In this demonstration we illustrate our tools' ability to:

- *Automatically extract elements of variability.* Our tool suite is able to efficiently identify the dimensions of variability that are present in the adaptive system, thus addressing Challenge 1. To achieve this, we apply natural language processing to the textual requirements documents provided. This processing not only groups related requirements together but also identifies elements of variability specified within them to produce an initial variability model in the form of a FMP feature model[2].
- *Enrich variability models.* Goal models justify the existence of all functional requirements (hard goals) and quality measures (soft goals) of a software system in terms of stakeholder intentions. The natural language approaches implemented by our tools are also able to identify candidate goals from the requirements text, further contributing to Challenge 1. For further elaboration of the generated feature/goal model, we apply a lexicon-based variability/commonality, adaptation, and context identification to enrich the feature model produced.
- *Model adaption.* Our methodology provides a domain specific language and editors to support efficient specification of adaptive systems, coping with the specific adaptive system complexities, and so addresses Challenge 2. These adaptation models define the variability in the system, the variability in the context of the system and rules to link changes in the context of the system with the configuration to be used.
- *Configuration verification and validation.* Design time validations are performed upon the adaptation models using smart selection algorithms and identifying the most appropriate configuration from the complete configuration space, as well as by providing design time simulations to analyse and validate adaptation behaviour, thus addressing Challenge 3. Each of these configurations is fully connected to their requirements level models. This allows the best configuration to be further analysed at requirements level, validating the selected configuration against stakeholder preferences and expectations, analysing conflicts and dependencies.
- *Synchronise models.* Further tool support is offered to allow the created feature and adaptation models to remain consistent with each other when changes are made. This involves ensuring that consistent views are maintained and traceability supported between the models and to ensure Challenge 3 continues to be addressed even when the models evolve.

[2]Any relevant notation can be targeted (e.g OMG's CVL)

IV. Case-Studies

During the course of the DiVA project, we applied these tools to two industrial case-studies: a Customer Relations Management system and an Airport Crisis Management system. Through these case-studies we observed that the benefits of our tools are more pronounced in longer documents. The larger the document the more manual effort is needed to read and fully understand the requirements to construct accurate models. However, our tool suite does not encounter such limitations. In fact, due to the nature of the natural language processing techniques applied, longer documents produce more accurate results. More generally, the industrial experts evaluating these tools found them to produce models that are a useful starting point for system development and an aid to understanding the relationship between requirements.

In this demo, we will apply the tools to the Airport Crisis Management System. We will begin by processing a textual requirements document to illustrate the DiVA methodology and tools to construct an adaptation model. We will then show how design time validation of the model is performed, how a target configuration is chosen, and then validated at the requirements level.

Acknowledgements

This work funded by the DiVA project (EU FP7 STREP, contract 215412) and the DISCS project (IEF-221280).

References

[1] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics - Volume 1*, COLING '98, pages 86–90, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

[2] Nelly Bencomo and Gordon Blair. Software engineering for self-adaptive systems. chapter Using Architecture Models to Support the Generation and Operation of Component-Based Adaptive Systems, pages 183–200. Springer-Verlag, Berlin, Heidelberg, 2009.

[3] Arnor Solberg et al. Diva whitepaper - a model-based approach for construction and run-time management of adaptive systems: Diva practices and lessons learned. *DiVA Project*, (D7.4), 2011.

[4] Ji Zhang et al. Modular verification of dynamically adaptive systems. In *Proceedings of the 8th ACM international conference on Aspect-oriented software development*, AOSD '09, pages 161–172, New York, NY, USA, 2009. ACM.

[5] Kun Chen et al. An approach to constructing feature models based on requirements clustering. In *Proceedings of the 13th IEEE International Conference on Requirements Engineering*, pages 31–40, Washington, DC, USA, 2005. IEEE Computer Society.

[6] Sotirios Liaskos et al. On goal-based variability acquisition and analysis. In *Proc. International Requirements Engineering Conference (RE'06*, pages 11–15. IEEE Computer Society, 2006.

[7] Charles Fillmore. The case for case. In Emmon Bach and R. Harms, editors, *Universals in Linguistic Theory*. Holt, Rinehart, and Winston, New York, 1968.

[8] Heather J. Goldsby, Pete Sawyer, Nelly Bencomo, Betty H. C. Cheng, and Danny Hughes. Goal-based modeling of dynamically adaptive system requirements. In *15th Annual IEEE Int. Conf. On the Engineering of Computer Based Systems*, 2008.

[9] Bowen Hui, Sotirios Liaskos, and John Mylopoulos. Requirements analysis for customizable software: A goals-skills-preferences framework. In *th IEEE Int. Requirements Engineering Conference (RE'03*, pages 117–126, 2003.

[10] Ji Zhang and Betty H. C. Cheng. Model-based development of dynamically adaptive software. In *Proceedings of the 28th international conference on Software engineering*, ICSE '06, pages 371–380, New York, NY, USA, 2006. ACM.