Reverse Engineering of Domain Knowledge for Improving Configuration Management

Vasil Tenev Martin Becker Angjela Davitkova Damjan Gjurovski
Fraunhofer Institute for Experimental Software Engineering IESE, Kaiserslautern, Germany
{vasil.tenev, martin.becker, angjela.davitkova, damjan.gjurovski}@iese.fraunhofer.de

Abstract: As a product family evolves with the increasing number of customer specific members, the product configuration becomes extremely intricate. Configuration key-value settings are often incompletely documented, so their influence on the product structure and behaviour remains hidden. Since side effects and interdependencies of configuration settings are only partially known, the products can only be configured manually. In order to make the product variant management more efficient, we present an approach to reverse engineer the configuration knowledge from product configurations using data analysis techniques. We use correlation analysis to extract dependencies between configuration items. Our approach is conducted on an industrial product family with thousands of individually configured product instances. Each product configuration contains between 20 000 and 30 000 configuration parameters. Our goals in this case are (i) to accelerate the configuration process, (ii) to increase the cost-effectiveness for quality assurance, and (iii) to extract and document the domain knowledge.

Keywords: reverse engineering; correlation mining; configuration management

1 Introduction

In contrast to the reference model for product line engineering [1], companies do not follow variation management approaches in the practice. Typically, they are driven by short delivery timelines and higher expectations for customisation. Applying incremental and solution oriented approaches, like Clone-and-Own, has its challenges. Due to missing documentation of a variability model and traces to the variation points in the code, it is hard to understand how different configurations affect the product. Moreover, validation and verification of consistency is almost impossible, because of the number of individually specified product instances. In this work, we investigate an industry case of a medium-sized enterprise, which supplies stationary cyber-physical systems to approximately 10 different big-size clients. While there are commonalities between the clients on a functional level of abstraction, they require high customisation of the system instances. Each product instance in the field needs to be individually configured to local infrastructure, leaseholder, business model, and much more. This situation increases the variability complexity in addition to the typical challenges with respect to the evolution of the product family. Currently, a product has between 20k and 30k configuration parameters on average. This configuration data (key-value pairs) is distributed over hundreds or even thousands of INI files per product instance. A staged semi-automatic configuration process is used in the case that requires experienced experts and causes a bottleneck for the overall delivery process.

The following chapters are structured as follows. Section 2 presents related work on the topic of correlation mining. In Section 3, we discuss our approach to extract and document the domain knowledge for improving the configuration support and management. Section 4 presents our conclusions and provides a short summary.

2 Related Work

In the context of this paper, we see the key-value pairs in an INI file as configuration parameters and their respective values. Thus, we treat these configuration parameters as technical low-level product features and settings.

Since feature correlations are often documented in feature models, studies on the extraction of feature models are related to this work. Czarnecki et al. [2] introduced the concept of probabilistic feature model and extracted soft and hard feature constraints from product configurations. Lora-Michiels et al. [3] also proposed a reverse engineering approach to extracting a feature model including structural and transversal feature correlations from product configurations. However, the correlations they both identified are only between binary features. They used association-mining techniques to identify feature correlations. However, neither of them considered complex correlations between sets of multiple feature assignments.

This industry case study is based on our previous work on reverse engineering complex feature correlations from product configurations [4] and describes further related work in more details.

3 Configuration Knowledge Extraction

While we previously [4] concentrated on the computational analysis of extracting complex feature correlations, in this study a need for more systematic approach emerged, like the GQM measurement model [5]. We identified the main business goals for this industry case. From there we derived questions and respective measurements on the configuration data in order to extract the relevant information needed for taking strategical decisions.

3.1 Goals

Our investigations showed that there are three top priorities in this context: fast configuration, economical quality assurance, and efficient access to domain knowledge.

(G1) Fast Configuration. The biggest pain point for a company with such degree of product customisation is the maintenance cost due to the (re-)configuration need for every product instance in the field. A reconfiguration must be done after every kind of software update, be that a new software release or a distribution of a bug fix/patch. Additionally, a slow configuration process increases the hurdle for new clients and it cannot be justified in practice.

(G2) Economical Quality Assurance. Quality assurance is one of the main cost factors in the development of variant-rich systems. Domain knowledge about configuration parameters interrelations and possible values in machine-readable form are essential for fully automated completeness and consistency analysis. Thus, possible misconfigurations can be detect more effectively.

(G3) Efficient Access to Domain Knowledge. Not only machines, but also humans require efficient and practical access to configuration knowledge. New employees are faster in finding and learning information. Therefore, they are able to start (re-)configuring products by themselves earlier. The workload can thus not only be distributed between more domain experts, but reduced in total.

In order to assess the business aspects of these goals, more information is required. On one side, dependencies between product features and settings are important. On other side, configuration settings have effects on the product functional and non-functional properties that could not be neglected.

3.2 Questions

The goals elaborated in Subsection 3.1 raise the need to answer the following questions:

(Q1) How similar are the configurations? On first place, answers to this question provide means to accelerate the configuration process (G1). Automated extraction of such information enables computer-aided support: for optimising the update/release approach; and for the planning of reconfiguration steps. Second, quality assurance (G2) can take advantage to limit the amount of unit tests by identifying overlapping between configurations.

(Q2) What is the distribution of the values? The distribution of the values per configuration parameter mainly affects the economy of quality assurance (G2) and more specific testing. Planning and generating of tests, as well as verifying coverage, can only be achieved and automated if there is knowledge on the usage of values. Moreover, the availability of this information supports the comprehending of configuration knowledge (G3).

(Q3) What are the relations between parameters and values? Relations between parameters and values can be dependencies, implications, correlations, etc. They contribute to all three goals (G1, G2, G3), by limiting the configuration space in formal and machine-accessible way.

(Q4) How did the configuration space evolve over time? Usage of configuration parameters changes over time respectively to the evolution of the product family. Herewith, the configuration space can also be reduced by recognising legacy parameters and values. Excluding them and redefining the scope supports all goals (G1, G2, G3).

3.3 Measurements

In this Subsection, we discuss the metrics and the tools we discovered to be most helpful for improving the configuration management in this industry case study with respect to the defined questions from previous section.

(M1) Number of equivalent key-value pairs. We applied the Fraunhofer Variant Analysis tool [6] to assess the similarity between configurations (Q1). In this case, we found that the client is the most significant factor for the diversity. Therefore, additional comparisons on configurations within the same client were deducted for data noise reduction.

(M2) Number of values per key. Measuring range and distribution of values (Q2) was used additionally to compute a statistical default value per configuration parameter.

(M3) Association analysis. A data mining technique is used for discovering relationships among the different elements, i.e. keys alone or key-value pairs (Q3). For performance reasons, a new set-based association rule-mining algorithm was developed. Unlike the traditional association rule mining, such as approaches based on the Apriori Principle [7], this algorithm utilizes that once elements appear together at least once, they are a relevant pattern.

(M4) Number of parameter changes. We count how many parameters changed or do not change at all with respect to their values and usage over time (Q4).

In the end, all raw data, measurement results and extracted information are persisted into a configuration knowledge database. This database is used as a storage and source of data for further analyses. Moreover, it is applied for the automation of configuration processes and integration to the workflow of the domain experts.

4 Conclusion

In this paper, we presented our GQM-based approach for improving the configuration management. It was developed during an industry case study to reverse engineer configuration knowledge on usage of parameters and settings, as well as, retrieving relationship information.

References

[1] ISO/IEC 26550 Software and systems engineering -- Reference model for product line engineering and management, 2015.

[2] Czarnecki, K. et al. 2008. Sample Spaces and Feature Models: There and Back Again. *SPLC 2008, Limerick, Ireland, September 8-12*, (2008), 22–31.

[3] Lora-Michiels, A. et al. 2010. A Method Based on Association Rules to Construct Product Line Models. *VAMOS 2010. Proceedings* (2010), 147–150.

[4] Zhang, B. and Becker, M. 2014. Reverse Engineering Complex Feature Correlations for Product Line Configuration Improvement. *EUROMICRO-SEAA 2014, Verona, Italy, August 27-29, 2014*, 320–327.

[5] Basili, V. et al. 2014. GQM+Strategies: A Comprehensive Methodology for Aligning Business Strategies with Software Measurement. (Feb. 2014).

[6] Tenev, V.L. et al. 2017. Variant Analysis: Set-Based Similarity Visualization for Cloned Software Systems. *SPLC 2017, Volume B, Sevilla, Spain, September 25-29*, (2017), 22–27.

[7] Tan, P.-N. et al. 2018. *Introduction to Data Mining (2nd Edition)*. Pearson.