Fraunhofer
IESE

Supporting Product Configuration in Application Engineering Using EXConfig

Authors:
Bo Zhang
Martin Becker

IESE-Report No. 016.16/E
Version 1.0
May 2016

A publication by Fraunhofer IESE

Fraunhofer IESE is an institute of the Fraunhofer Gesellschaft.
The institute transfers innovative software development techniques, methods and tools into industrial practice, assists companies in building software competencies customized to their needs, and helps them to establish a competitive market position.

Fraunhofer IESE is directed by
Prof. Dr.-Ing. Peter Liggesmeyer
(Executive Director)
Prof. Dr. Dieter Rombach
(Director Business Development)
Fraunhofer-Platz 1
67663 Kaiserslautern
Germany

Supporting Product Configuration in Application Engineering Using EXConfig

Bo Zhang and Martin Becker

Fraunhofer Institute Experimental Software Engineering (IESE)
Kaiserslautern, Germany
{bo.zhang, martin.becker}@iese.fraunhofer.de

ABSTRACT

Nowadays systems are often developed following the product line approach in order to reduce time to market, achieve lower cost, and ensure high quality. To this end, common and variable requirements of product variants are specified as features in a feature model, so that each product variant can be configured and customized along the development lifecycle. While developers in family engineering tend to use variability management tools (e.g., pure::variants) for feature modeling and developing core assets for reuse, such sophisticated tools might be too complicated and inappropriate for product configuration in application engineering. In order to solve this challenge, this paper introduces an Excel-based product configurator called EXConfig, which focuses on product line features in the problem space and supports staged product configuration in application engineering. This tool can be easily customized and integrated with other system design tools or variability management tools, which connects application engineering and family engineering in development. The usage of this tool and its integration have been validated several times in industry.

CCS Concepts

• Software and its engineering~Software product lines.

Keywords

Variability Configuration; Feature Model; Enterprise Architect; Excel.

1. INTRODUCTION

Software product line engineering (PLE) [10][8] is an established approach with systematic reuse to reduce time to market, achieve lower cost, and ensure high quality. To this end, common and variable requirements of product variants are specified in a variability model – typically a feature model [6] – so that each product variant can be configured and customized along the development lifecycle. As a product line evolves both in space and in time [7], variability management becomes challenging due to increasing number of features and their complex constraints [4][9].

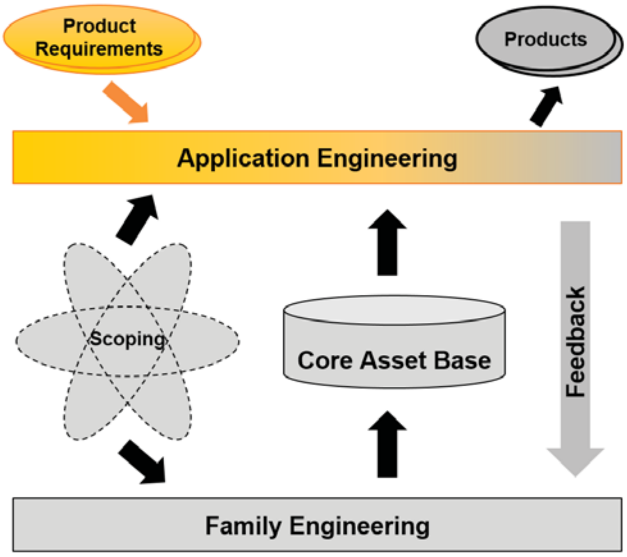
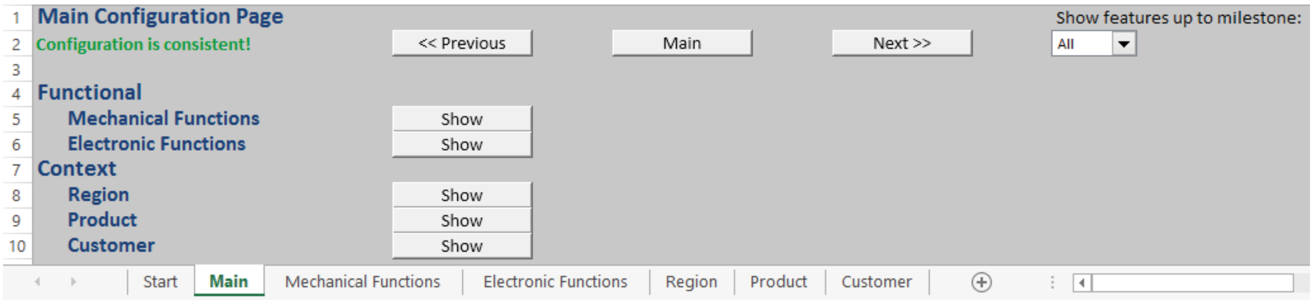


Figure 1. Our Focus in the Product Line Lifecycle (Adapted from [10])

In recent years, variability management tools (e.g., pure::variants [11] and Gears [5]) have been developed for tackling this variability complexity in feature modeling and developing core assets for reuse. While these sophisticated engineering tools are mainly used in family engineering, they might be inappropriate for product configuration in application engineering. The reason is that people who perform product configuration are often business sales and requirement engineers, who are only interested in the external product features but do not care about how to implement these features with reuse internally. Therefore, the full-fledged variability management tools would be too complicated for these people and cause unnecessary technical learning effort.

In order to solve this challenge, this paper introduces our EXConfig tool, which is an Excel-based product configurator for staged system variability configuration and its verification based on the standard feature model syntax. It allows business sales and application engineers to focus only on product line features and product configurations in the problem space (as illustrated in Figure 1 in yellow color). Moreover, as a lightweight tool it can be easily customized and integrated with other system design tools such as Enterprise Architect (EA) or variability management tools. This connects application engineering and family engineering, making

SPLC'16, September 16–23, 2016, Beijing, China.
Copyright is held by the owner/author(s). Publication rights licensed to ACM. ACM 978-1-4503-4050-2/16/09…$15.00
DOI: http://dx.doi.org/10.1145/2934466.2962727



1 Main Configuration Page — Show features up to milestone: All
2 Configuration is consistent! — << Previous | Main | Next >>
3
4 Functional
5 Mechanical Functions — Show
6 Electronic Functions — Show
7 Context
8 Region — Show
9 Product — Show
10 Customer — Show

Figure 2. The Main Configuration Page of EXConfig

this tool a critical supplement to the existing variability management tools in product line development.

In this paper we provide the following contributions:

1. Application scenario of the product configurator EXConfig.
2. Implementation techniques of the product configurator EXConfig.
3. A concrete example which shows how EXConfig is integrated with Enterprise Architect and being used for staged product configuration in practice.

The paper is organized as follows: Section 2 introduces the product configuration support using the EXConfig tool, while Section 3 explains how this tool can be applied in practice and integrated with other tools. Finally, conclusion with future work are presented in section 5.

2. PRODUCT CONFIGURATION

This section introduces the product configuration support of EXConfig with an example in the automotive domain. After a valid feature model is imported in EXConfig (will be explained in Section 3), one can start the configuration process. Since this tool is developed based on Microsoft Excel with VBA macros, it is easy to use and customize especially for users with less technical background (e.g., business sales). The following sub-sections introduce detail configuration functionalities.

2.1 Configuration Page Navigation

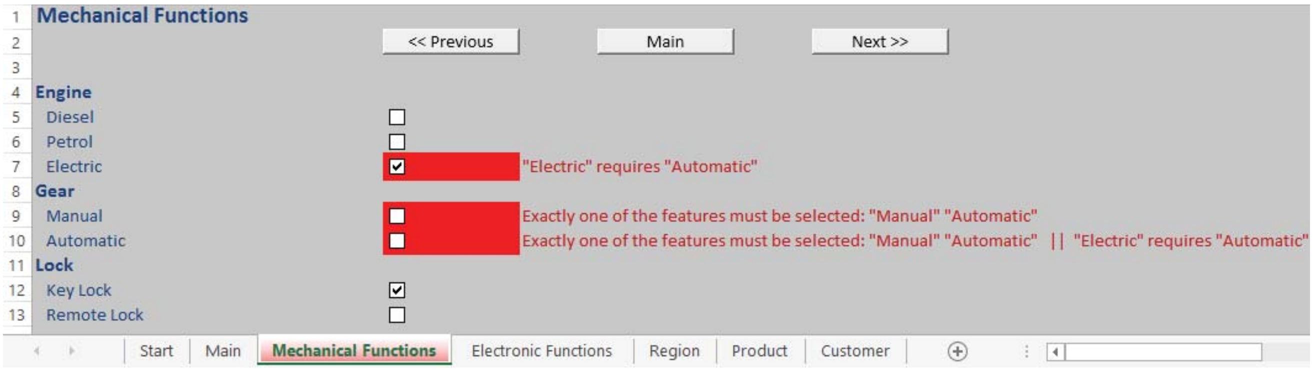
In EXConfig, a configuration process starts from the main configuration page as shown in Figure 2. This page shows the first-and-second-level features, and for each of the second-level feature a "show" button is displayed that navigates to a corresponding sub configuration page when being pressed. While a user can easily navigate to a sub configuration page from the main page, in a sub page he or she can also navigate either back to the main page or to its previous or next page by pressing corresponding buttons in the second row.

2.2 Feature Configuration

Figure 3 shows a sub configuration page, where one can select or deselect a feature as a configuration operation according to the needs of a product. Once the operation violates the feature constraints defined in the feature model, corresponding warnings will be displayed immediately in red color. For instance, in Figure 3 the feature "Electric" is selected, which causes an error in both the seventh and tenth row because its required feature "Automatic" is not selected. Here the EXConfig tool validates different cross-tree constraints including "requires", "excludes", "requiresAny" and "requiredForAll". It also validates the feature group constraints. For instance, in Figure 3 an alternative feature group constraint is violated with respect to the feature group "Gear", and corresponding warnings are displayed in the ninth and tenth row.

If a configuration page contains at least one warning, the page tab will be rendered with a red background color (such as the tab "Mechanical Functions" in Figure 3). On the contrary, if the entire configuration is validated with no constraint violation, a global message "Configuration is consistent!" will be displayed in the second row of every configuration page as shown in Figure 2. When the configuration is complete, it can be exported automatically as a CSV file, which can be used for product implementation in the next phase.

Moreover, the EXConfig tool supports staged configuration [3], which is a more realistic configuration approach (comparing with holistic configuration) especially for developing large and complex systems. In this scenario, product configuration is supposed to be done incrementally by different stakeholders (e.g., business sales and application requirements engineers), who only configure some features at a certain stage. To this end, each feature has an additional attribute "milestone" indicating the corresponding configuration stage (e.g., customer requirements review), and one can decide up to which milestone the corresponding features should be shown in this tool (it will show all features by default). In other



1 Mechanical Functions
2 << Previous | Main | Next >>
3
4 Engine
5 Diesel □
6 Petrol □
7 Electric ☑ "Electric" requires "Automatic"
8 Gear
9 Manual □ Exactly one of the features must be selected: "Manual" "Automatic"
10 Automatic □ Exactly one of the features must be selected: "Manual" "Automatic" || "Electric" requires "Automatic"
11 Lock
12 Key Lock ☑
13 Remote Lock □

Figure 3. A Sub Configuration Page in EXConfig

words, EXConfig can only show a subset of the feature model (i.e., hide irrelevant features and feature constraints), so that each user can only focus on the features he or she is interested at a certain level of abstraction. This has been proved as an effective approach in industry.

2.3 Tool Customization

As the EXConfig tool is developed based on Microsoft Excel with VBA macros, it can be easily customized and enhanced in order to meet certain special needs of a company or a user. For instance, in one project we made a customization that a mandatory will be selected automatically once its parent feature is selected (although it may cause side effects). Also, another customization we made was to shown all sub-features of a certain parent feature in a separate configuration page, and this page will be activated once this parent feature is selected. This is to prevent the case that a large number of nested features are listed in one single configuration page. In Excel all these customizations as well as interaction with other external tools can be implemented by VBA macros with minor effort, and this tool can be also easily debugged in case of any errors.

3. APPLICATION AND INTEGRATION

This section introduces how EXConfig can be applied in real product development settings and integrated with other tools in practice. In order to conduct product configuration, a feature model needs to be first established in this tool. Although one can create a feature model manually in EXConfig, it is not realistic for a large system in practice. As the feature model is typically created either using a variability management tool (e.g., pure::variants [11]) or using a general-purpose system design tool (e.g., EA), it would be convenient to integrate EXConfig with these tools and import the feature model. To this end, we have developed an additional Powershell script for importing a feature model in EA to EXConfig. The integration workflow is illustrated in Figure 4, which is be explained in details in this section. Integration with other tools besides EA can be also implemented in a similar way.

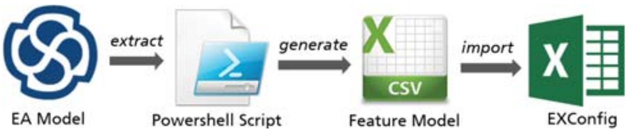


Figure 4. Importing an EA Feature Model to EXConfig

3.1 Feature Modeling in EA

While there are dedicated tools (e.g., pure::variants [11]) that support feature modeling, many companies still create feature models using general-purpose design tools like EA. This makes sense because they have probably already created other models using the same tool along the development process. For instance, a company may use EA for creating the feature model, the requirements model, the design model, and so on.

In EA a feature model can characterize product commonalities and variabilities in the form of a UML model, while it still complies with the classic feature model syntax [6][2]. Figure 5 shows the EA feature model of mechanical functions as part of the automotive example. In this EA feature model, a feature has a stereotype that indicates its feature type (e.g., mandatory, optional, or alternative).

A feature with the stereotype "Domain" is considered as a section or group name of its sub-features and is not configurable (i.e., it can neither be selected nor deselected). A feature and its sub-feature is connected by the UML aggregation relationship, while cross-tree

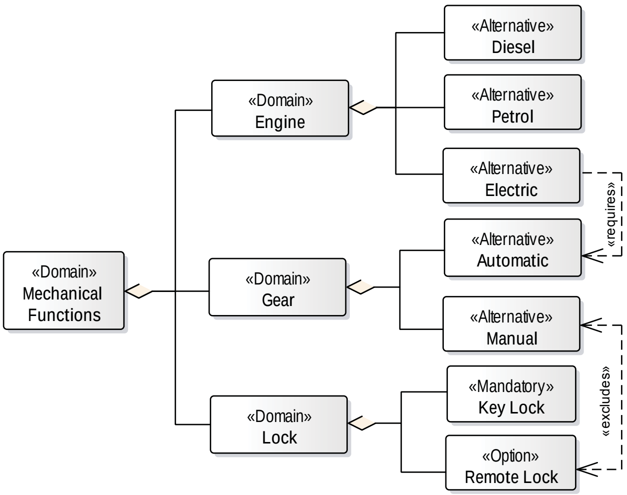


Figure 5. Mechanical Functions in an EA Feature Model

feature constraints like "requires" and "excludes" are specified by UML dependencies. For instance, in the example EA feature model as shown in Figure 5, the feature "Electric Engine" requires the feature "Automatic Gear", while the features "Manual Gear" and "Remote Lock" exclude each other.

3.2 Feature Model Extraction and Import

When a feature model has been created in an external tool (e.g., EA), it needs to be extracted and imported into our EXConfig tool in order to enable product configuration. To this end, we have developed an additional Powershell script which extracts the feature model from an EA model and save it as a CSV file. Technically, the script reads an EA model via its DOM API, conducts a few EA database queries, and identifies features and related information (e.g., feature type, constraints, description, etc.) based on predefined mapping rules (e.g., the feature type is written in the field of stereotype of an EA model). In practice, the extraction of the feature model from EA can be customized in the Powershell script according to customer's needs (e.g., only extraction EA objects with certain naming conventions as features). Typically, the feature model extraction from an EA model with thousands of elements using our Powershell script takes a couple of minutes.

Given an extracted feature model in the CSV format, it can be imported into our EXConfig tool. The imported feature model does not necessarily have to be extracted from an EA model, but also can be extracted from other tools or even created manually in theory. If the imported feature model is syntactically consistent, the corresponding configuration pages will be generated and initialized (otherwise errors will be raised). Then the EXConfig tool is ready for product configuration.

4. CONCLUSION

This paper presents our Excel-based EXConfig tool as a lightweight product configuration solution that is based on the feature model syntax and can be customized and integrated easily with other tools like EA in the development lifecycle. It has been applied several times in industry, where users can import a feature model automatically from EA and then conduct staged system variability configuration and verification. This is a significant advantage in application engineering, especially for users with less technical background (e.g, business sales).

Currently we have only implemented its integration with EA. In the future, we would like to extend its interoperability and integrate it with other software and system development tools like PTC Integrity Modeler and IBM Rhapsody.

5. ACKNOWLEDGMENTS

Our thanks to Slawomir Duszynski for his early contribution in development of the EXConfig tool.

6. References

[1] T. Berger, R. Rublack, D. Nair, J. M. Atlee, M. Becker, K. Czarnecki, and A. Wkasowski, "A survey of variability modeling in industrial practice," in Proceedings of the Seventh International Workshop on Variability Modelling of Software-intensive Systems, ser. VaMoS '13. New York, NY, USA: ACM, 2013.

[2] K. Czarnecki and U. W. Eisenecker, Generative programming: methods, tools, and applications. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000.

[3] K. Czarnecki, S. Helsen, and U. Eisenecker, "Staged configuration using feature models," in Software Product Lines: Third International Conference, SPLC 2004. Springer-Verlag, 2004, pp. 266-283.

[4] S. Deelstra and M. Sinnema, "Managing the complexity of variability in software product families," Ph.D. dissertation, Institute of Mathematics and Computing Science, University of Groningen, Jul. 2008.

[5] Gears tool. BigLever company. http://www.biglever.com/solution/product.html. (Last visit: May 2016)

[6] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Nowak, and A. S. Peterson, "Feature-Oriented domain analysis (FODA) feasibility study," Software Engineering Institute, Carnegie Mellon University Pittsburgh, PA., Tech. Rep. CMU/SEI-90-TR-21, Nov. 1990.

[7] C.W. Krueger, “New Methods behind a New Generation of Software Product Line Successes”, In K.C. Kang, V. Sugumaran, and S. Park, “Applied Software Product Line Engineering”, Auerbach Publications, 2010, pp. 39-60.

[8] F. J. van der Linden, K. Schmid, and E. Rommes, Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[9] T. Patzke, M. Becker, M. Steffens, K. Sierszecki, J. E. Savolainen, and T. Fogdal, "Identifying improvement potential in evolving product line infrastructures: 3 case studies," in Proceedings of the 16th International Software Product Line Conference - Volume 1, ser. SPLC '12. New York, NY, USA: ACM, 2012, pp. 239-248.

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

[11] pure::variants. http://www.pure-systems.com/pv (Last visit: May 2016).

APPENDIX: DEMOSTRATION SCENARIO

The demonstration of EXConfig tool will use an automotive example to show how one can extract and import a feature model from EA to EXConfig and then conduct variability configuration and verification. The working scenario is shown in Figure 1, and future details of the demonstrated example have been shown in this paper.

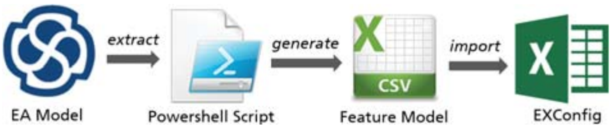
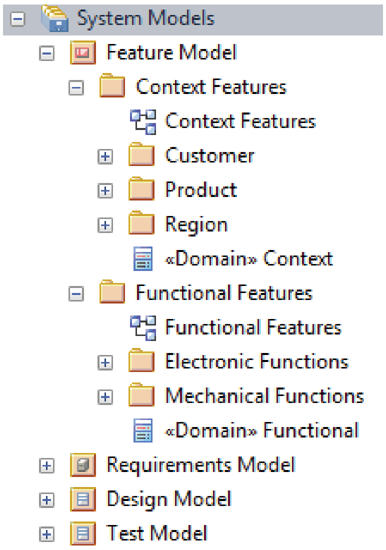


Figure 1. Importing an EA Feature Model to EXConfig

To be specific, the demonstration will follow the steps as below:

1. Introducing the basics of feature modeling and variability configuration, as well as typical configuration challenges in industry.
2. Introducing the automotive feature model described in EA (as shown in Figure 2).
3. Introducing the main GUI of EXConfig.
4. Demonstrating the automated extraction of the EA feature model as well as the import to EXConfig.
5. Demonstrating the staged variability configuration and verification using EXConfig.
6. Demonstrating the export of a complete configuration using EXConfig.
7. Demo conclusion
8. Q & A.



- System Models
  - Feature Model
    - Context Features
      - Context Features
      - Customer
      - Product
      - Region
      - «Domain» Context
    - Functional Features
      - Functional Features
      - Electronic Functions
      - Mechanical Functions
      - «Domain» Functional
  - Requirements Model
  - Design Model
  - Test Model

Figure 2. System Models using Enterprise Architect

Document Information

Title:	Supporting Product Configuration in Application Engineering Using EXConfig
Date:	May 2016
Report:	IESE-016.16/E
Status:	Final
Distribution:	Public Unlimited