UNIVERSITAS INDONESIA

FEATURE GROUPING USING ABSTRACT BEHAVIORAL SPECIFICATION LANGUAGE

PROPOSAL TESIS

REZA MAULIADI
1506782404

FAKULTAS ILMU KOMPUTER
PROGRAM STUDI MAGISTER ILMU KOMPUTER
DEPOK
JUNI 2016

UNIVERSITAS INDONESIA

FEATURE GROUPING USING ABSTRACT BEHAVIORAL SPECIFICATION LANGUAGE

PROPOSAL TESIS
Diajukan sebagai salah satu syarat untuk memperoleh gelar
Magister Ilmu Komputer

REZA MAULIADI
1506782404

FAKULTAS ILMU KOMPUTER
PROGRAM STUDI MAGISTER ILMU KOMPUTER
DEPOK
JUNI 2016

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF LISTINGS

CHAPTER 1
INTRODUCTION

This chapter describes about the background of the research, research questions, research goals, benefits of research, scopes of research, and outline of this thesis proposal.

1.1 Background

Software is a very essential need for some people or company in this modern era. It is not only used to support a large processes, but also to support the daily activities of its users. Software contains several features which describe the functions of the software. The need for the features could be vary among users. Even for the same type of software, not all features are really needed by the user. That is a challenge for software developers to create software which fits the needs of each users.

Instead of building software from scratch, it will be great if the developers can build software from reusable components. It can reduce the effort of developers to build a software for a user. One concept which relies on building software from reusable components is Software Product Line (SPL) [18].

In Software Product Line (SPL), a software is built from reusable components instead of from scratch so that software development can be done more efficiently [4]. The software will be built according to the specific needs of each user by using components that are already prepared before. It can reduce the effort of developers in building software for each user. One language which supports SPL is Abstract Behavioral Specification (ABS).

Abstract Behavioral Specification (ABS) is a modeling language which supports variability by using the concept of SPL [7, 10, 1]. ABS uses feature model to express variability and to organize features [1]. Feature model is a set of logical constraints that are used to express the dependencies between features. Feature model is represented as a tree of nested features. The example of feature model represented in feature diagram [16] shown in Figure 1.1.

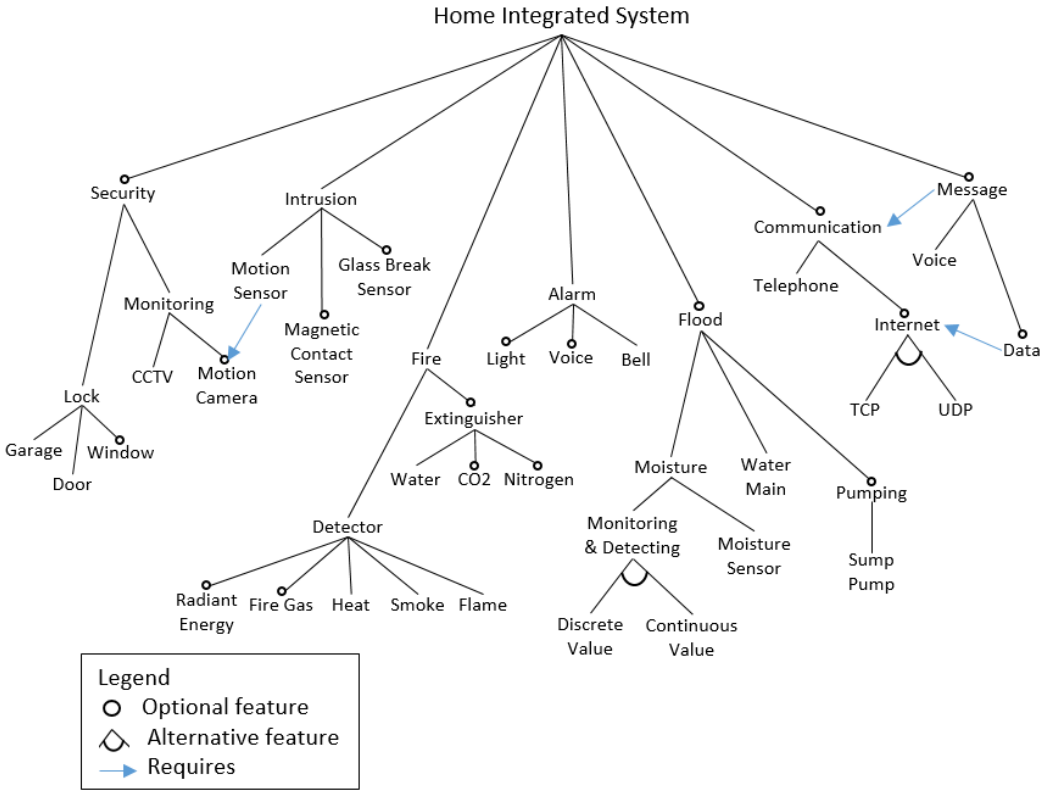

Figure 1.1: Home Integration System Feature Diagram

Source: [17] (with additional changes)

The figure above shows a set of features for Home Integrated System (HIS) [7]. There are several features which can be chosen by users, such as *Fire, Flood, Alarm, Intrusion*, etc. Some features are general feature which consist of or implemented by the features followed. An *Alarm* feature consists of *Bell, Voice*, or *Light* feature. Then, an *Internet* feature implemented by *TCP* or *UDP* feature.

A feature model is organized based on the visible characteristics of software [18]. It can be very large if the number of features available is also increasing. If a feature model has a large number of features, such as Linux kernel with more than 10.000 features [4], it can cause the selection of features process be more complex. To address this problem, the features in the feature model can be grouped based on their functions in general. According to [17, 18], features in the feature model can be grouped into feature binding unit (FBU). FBU is a group of interconnected features based on the relationships that exist in the feature model. By grouping the features, the complexity of the feature model can be reduced. Figure 1.2 shows the example of grouped features represented using feature diagram.

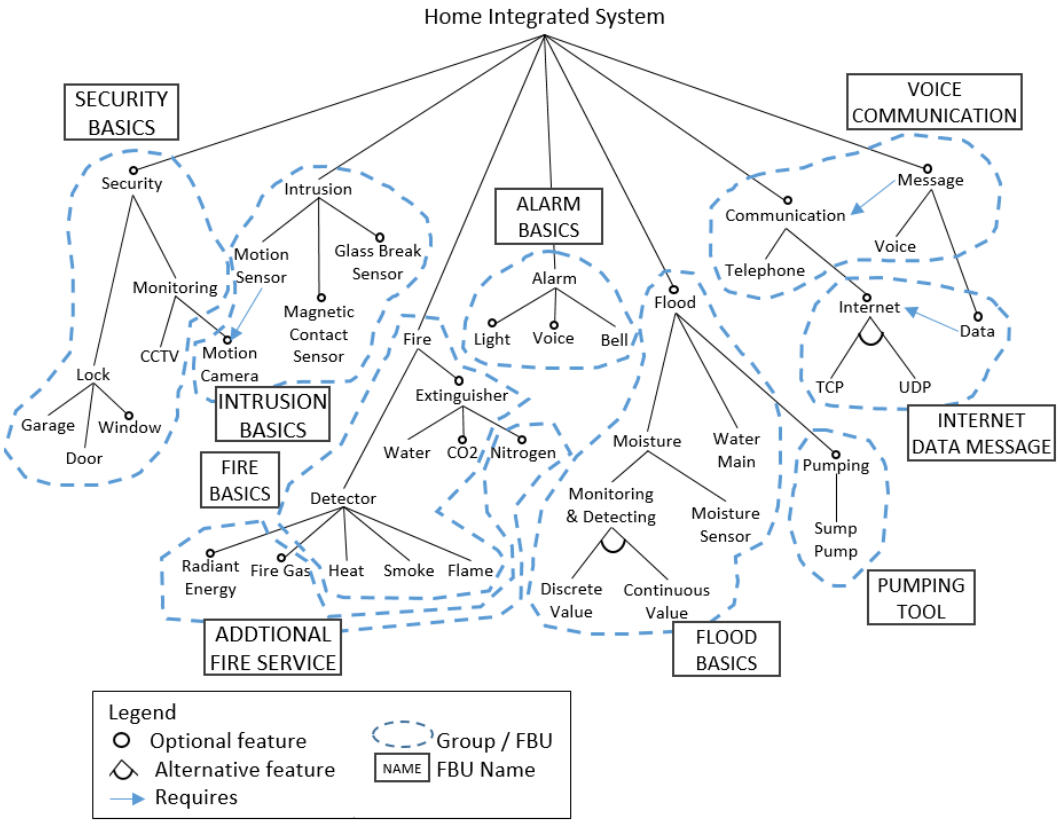

Figure 1.2: Home Integration System Feature Diagram with Grouping
Source: [17] (with additional changes)

There are several groups of basic features, as shown on the figure, such as *Security Basics, Intrusion Basics, Fire Basics, Flood Basics,* and *Alarm Basics*. Moreover, there are features grouped to be additional features, such as *Additional Fire Services, Voice Communication, Pumping Tool,* and *Internet Data Message*. The features grouped by the common services they provide and the relationship they have.

Until now, the selection of features to be included in a software product in ABS is done by listing the features chosen using product selection language (PSL) [10]. It could make the selection of features difficult to do, even more for a user. Feature grouping can help users in selecting which functions they want to implement. Another issue is about the graphical visualization for the selection of features. It also could make a selection of features be more difficult to do by the user.

In this research the feature grouping mechanism will be made and visualized. The visualization is done by using a simple web application as a tool support. The visualization of features is intended for users so they can make a selection of features with easily. Moreover, the purpose of the feature grouping is to reduce the complexity while doing the features selection. The features that have been selected

by the user will also be able to be directly generated into product selection, thereby reducing the effort to make product selection manually.

1.2 Research Questions

Based on the research purpose, the research questions which will be answered by this research are as follows:

1. How to implement feature grouping from feature model in ABS tools?
2. How to make a visualization of the feature grouping?
3. How to improve the efficiency of building software from the grouped features?

1.3 Research Goals

There are several research questions that have been stated. Based on the background of this research and those research questions, there are several goals which will be the target of this research. The research goals are as follows:

1. The feature grouping mechanism from feature model in the ABS tools is implemented.
2. The visualization of the result of feature grouping mechanism is implemented. Thus, this visualization could be a way to improve the efficiency of building software from the grouped features.

1.4 Benefits of Research

Many researchers are contributing in the development of ABS. This research will contribute in the development of ABS because this research goals are to provide the mechanism to group the features and visualize the result.

In practical term, the grouping of features, hopefully, can reduce the complexity while doing the features selection. The visualization of the features selection also gives ease for users to select the features. Then, as stated in the research goals, the visualization could be a way to improve the efficiency of building software from the grouped features.

1.5 Scope of Research

The aim of this research is to make a grouping mechanism for features, visualize them, and generate the selected features using product selection language. To limit the topics discussed in this research, there are scopes of this research as follows:

1. The focus of the work is to implement grouping mechanism, visualization using web application, and generating the selected features. The grouping mechanism will be implemented in the ABS code generator (compiler back-end). Then, the visualization and the product selection generator will be implemented in the web application.

2. Analyze the grouping mechanism, visualization and product selection generator using a case study. The case study used is ERP sales module features. The features are Odoo ERP sales module features with additional exclusions. Several features which are specific to the vendor will be ignored. It is because the ignored features could be not common features in a sales module of ERP.

1.6 Outline

The outline of this thesis proposal is describe as follows:

- Chapter 1 INTRODUCTION
  Chapter 1 describes the introduction of the thesis proposal that includes the background, research questions, research goals and scope, and the systematic of writing.

- Chapter 2 LITERATURE REVIEW
  Chapter 2 contains the literature review and theories which used in this research as the basis to support. This chapter explains software product lines, Abstract Behavioral Specification (ABS), and feature grouping.

- Chapter 3 PROPOSED GROUPING TECHNIQUE
  Chapter 3 discuss about the grouping technique from the feature model which can be used for selecting the features and generating the selected features.

- Chapter 4 CASE STUDY
  Chapter 4 explains about the case study used in this research which is an ERP sales module features.

REFERENCES

[1] ABS-Tools. *The ABS Language Specification*. 2013.

[2] ANTLR2.org. Antlr tree construction. http://www.antlr2.org/doc/trees.html. [Online; accessed 6-April-2016].

[3] ANTLR.org. About the antlr parser generator. http://www.antlr.org/about.html. [Online; accessed 6-April-2016].

[4] S. Apel, D. Batory, C. Kästner, and G. Saake. *Feature-Oriented Software Product Lines*. Springer, 2013.

[5] N. F. Apriani. *Delta-Relational Mapping using the Abstract Behavioral Specification Language*. 2015.

[6] S. Chung and Y.-S. Lee. Modeling web applications using java and xml related technologies. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pages 10–pp. IEEE, 2003.

[7] D. Clarke, R. Muschevici, J. Proença, I. Schaefer, and R. Schlatte. Variability modelling in the abs language. In *Formal Methods for Components and Objects*, pages 204–224. Springer, 2010.

[8] J. Conallen. *Building Web applications with UML Second Edition*. Addison-Wesley Longman Publishing Co., Inc., 2002.

[9] K. Czarnecki and M. Antkiewicz. Mapping features to models: A template approach based on superimposed variants. In *Generative Programming and Component Engineering*, pages 422–437. Springer, 2005.

[10] R. Hähnle. The abstract behavioral specification language: A tutorial introduction. In *Formal Methods for Components and Objects*, pages 1–37. Springer, 2013.

[11] M. Haverbeke. *Eloquent JavaScript: A Modern Introduction to Programming*. No Starch Press, 2014.

[12] G. Hedin and E. Magnusson. JastaddâĂŤan aspect-oriented compiler construction system. *Science of Computer Programming*, 47(1):37–58, 2003.

[13] JastAdd.org. Jastadd concept overview. http://jastadd.org/web/documentation/concept-overview.php. [Online; accessed 6-April-2016].

[14] JastAdd.org. Reference manual for jastadd 2.2.2. http://jastadd.org/web/documentation/reference-manual.php#Aspects. [Online; accessed 6-April-2016].

[15] E. B. Johnsen, R. Schlatte, and S. L. T. Tarifa. Deployment variability in delta-oriented models. In *Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change*, pages 304–319. Springer, 2014.

[16] C. Kästner and S. Apel. Feature-oriented software development a short tutorial on feature-oriented programming, virtual separation of concerns, and variability-aware analysis, 2013.

[17] J. Lee and K. C. Kang. Feature binding analysis for product line component development. In *Software Product-Family Engineering*, pages 250–260. Springer, 2003.

[18] J. Lee and D. Muthig. Feature-oriented variability management in product line engineering. *Communications of the ACM*, 49(12):55–59, 2006.

[19] R. Nixon. *Learning PHP, MySQL, JavaScript, and CSS: A step-by-step guide to creating dynamic websites*. " O'Reilly Media, Inc.", 2012.

[20] K. Pohl, G. Böckle, and F. J. van Der Linden. *Software product line engineering: foundations, principles and techniques*. Springer Science & Business Media, 2005.

[21] L. Rosen and L. Shklar. *Web Application Architecture: Principles, Protocol and Practices*. John Wiley & Sons Ltd.: West Sussex, England, 2009.

[22] S. Schulze, O. Richers, and I. Schaefer. Refactoring delta-oriented software product lines. In *Proceedings of the 12th annual international conference on Aspect-oriented software development*, pages 73–84. ACM, 2013.

[23] P. Y. Wong, E. Albert, R. Muschevici, J. Proença, J. Schäfer, and R. Schlatte. The abs tool suite: Modelling, executing and analysing distributed adaptable object-oriented systems. *International Journal on Software Tools for Technology Transfer*, 14(5):567–588, 2012.