

# Towards an Extension of the Theme/Doc Approach to Specify Variability in SPL Development

Inês Simão and João Araújo  
CITI, FCT/Universidade Nova de Lisboa,  
Quinta da Torre, 2825-516 Caparica, Portugal  
nessimao@gmail.com, ja@di.fct.unl.pt

## Abstract

In Software Product Lines (SPL), relying only on feature models is difficult to explain why a particular feature is required, therefore information about relations between requirements and features are needed. Another problem is that features can be repeated in the model making it more complex to understand, characterizing the crosscutting problem. Aspect-oriented requirements engineering approaches, such as Theme/Doc, help building more modularized feature models by addressing the crosscutting feature problem. In this paper we adapt Theme to cope with SPL development and we define an initial set of heuristics to map the proposed Theme-SPL models into feature models.

## Keywords

Software Product Lines, Feature Model, Aspect-Oriented Requirements Engineering.

## 1. Introduction

An SPL is a set of software systems which share common characteristics, satisfying the needs of a particular segment of the market and are developed from a common set of core assets [2]. Part of the current research on requirements for SPL considers defining and structuring requirements artifacts as modularized as possible to avoid the repetition of features in their models. The aim is that these artifacts are able to serve as the basis for deriving cost-effective products and also to facilitate their development. A feature model is a hierarchical representation, which aims to capture the structural relationships between the features of an application domain. The model also represents the common and variable features and dependencies between those features.

Aspect-oriented approaches are used to obtain more modularized models. One approach that addresses efficiently the modularization of requirements is the Theme/Doc approach [1], which is characterized by identifying, modeling and composing crosscutting requirements by identifying themes and their relationships. However, the use of the Theme approach to describe software product lines has not been sufficiently explored yet. But we believe that Theme models, like other aspect-oriented approaches, offer a natural way to identify commonalities and variabilities at early requirements stages, and can be used to help specifying a corresponding feature model, therefore providing a more expressive approach for SPL requirements engineering.

Thus, this work aims to develop an approach (Theme-SPL) that investigates how the development of SPL can be benefited from the integration of an aspect-oriented approach, namely, the Theme approach, by defining an initial set of heuristics to map Theme-SPL models into feature models.

## 2. The Theme-SPL Approach

The process of the approach is divided into two main activities. Initially elicitation and analysis of requirements and themes are realised, where the requirements documentation is examined;

then the main themes and their relationships are identified. After this analysis the feature model is built, where the features are primarily identified and the variability analysed, using the action view defined in the previous step. In the last activity of the process we will refine the feature models, where the aspectual themes will be identified, the crosscutting view will be built, and based on this the feature model is refined. The process will be described in the next section using a simple example, a mobile phone system. The aim is to develop software components to make calls, put phone calls on hold, insert contacts in a contacts list, send and receive SMS and MMS, take pictures, and transfer data between mobile phones.

### 2.1 Applying Theme-SPL to an example

Let us assume that the identification and modeling of themes of the SPL are already performed and a list of themes is provided. The themes for our problem are: Make call, put phone call on hold, receive MMS/SMS, send MMS/SMS, take pictures and transfer data. Table 1 shows the requirements list for the themes "Make call" and "Call waiting".

Table 1. Reqs. for "Make Call" and "Call Waiting" themes

R No.	Requirements description
R4	To make a call a user should search the receiver contact in, contacts application, and press the call key.
R5	To make a call we need to check the balance on the phone.
R6	If there is no balance in phone, displays a message "Out of balance".
R7	If there is balance, the phone call is made.
R8	When the user is making a call, your call may be on hold, showing on the display the message "Call Waiting".

We propose the following heuristics to extend the Theme approach with concepts of SPL.

**H.1 Identify the root.** Find a requirement that says what the system is. Model parts (possibilities) that make up the root and each component will lead to a root theme. The link type is explained in heuristic H.2 and H.3.

**H.2 Identify optionality.** If it appears in a requirement descriptions that indicate optionality, such as "when X may, or may not, run Y", obtain the themes X and Y, where Y is optional and a link with the stereotype «alternative» is defined indicating that when the theme X occurs, Y does not have necessarily to be executed (optional). This is the case of the theme "Put call on hold" related to the theme "Make call".

**H.3 Identify mandatory.** If in a requirement it appears options to indicate a mandatory situation, such as "when X, there must be Y", "X requires Y", the theme Y is mandatory and is represented by a connection with the stereotype «obligatory» that indicates that when the theme X runs Y must be executed.

**H.4 Identify themes that can be grouped.** If two or more themes are closely related, then it is created a new theme with an aggregate to the themes that prompted it. The themes "Send SMS" and "Send MMS" are related, so we create a new theme

"Send Message", whose sub-themes will be those which originated them.

**H.5 Identify OR alternatives.** If a requirement is related to a theme, and several alternatives appear in expressions such as "selects at least one of several", then the theme will be a ThemeGroup that contains a minimum and maximum number of options, and the alternatives will be a GroupedTheme each, whose link has the stereotype «alternative\_or». For example, for the ThemeGroup "Transfer Data" you can choose between one and three GroupedThemes ("Bluetooth", "Infrared" and "USB").

**H.6 Identify aspectual themes.** An aspectual theme is a theme that is present into two or more requirements and the same theme is responsible for all requirements where it appears. Define a relationship for the aspectual theme to the base theme with the stereotype «crosscutting». Later, when mapping to features we need to define whether the relationship between the aspectual theme and the base theme is mandatory or optional. In this example, the aspectual theme is "Manage Contacts" since it is needed to "Make Call" and to "Send Message".

After applying the heuristics we obtain an action view model with stereotypes that extend Theme with SPL concepts. The result is that the mapping to a feature model is straightforward.

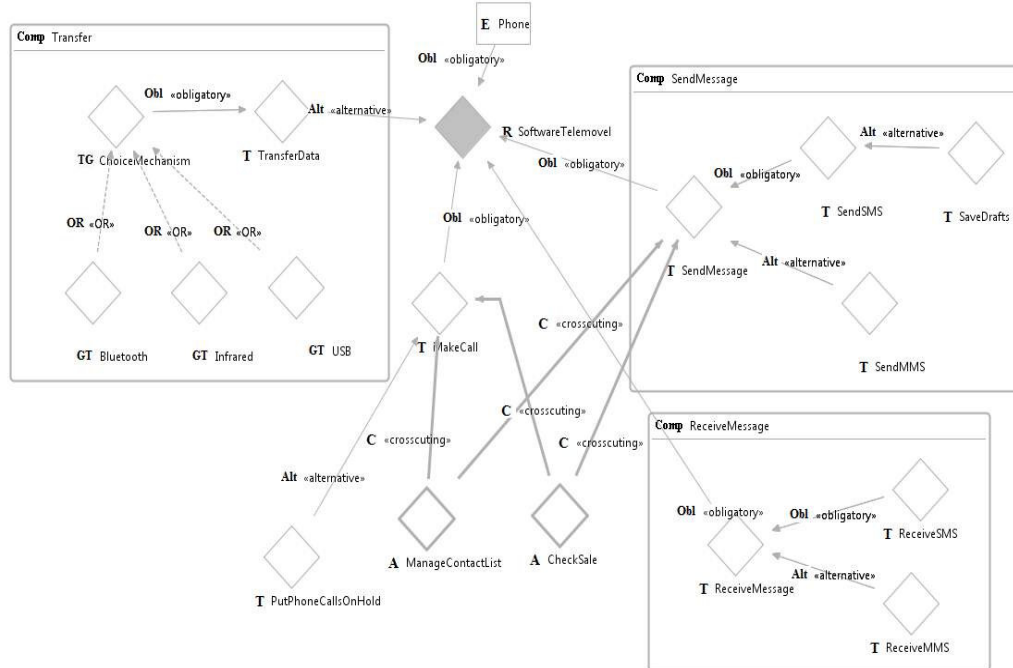


Figure 2. Theme-SPL action view

#### 4. Conclusions and future work

Our purpose in this paper is to show that the Theme approach can be used to identify and specify variability of software product lines. We enriched the Theme approach to capture the variability by defining a set of heuristics and stereotypes. This facilitated the derivation of feature models. As future work we will extend the set of heuristics to include feature interaction detection and apply the approach to other real case studies. Also, we will define and implement automatic rules to transform Theme-SPL models into feature models.

#### Acknowledgements

This work was partially funded by FCT MCTES.

Each theme becomes a feature and the relationships between the themes are directly mapped to existing relationships between features in a feature model. The Theme-SPL approach has a tool developed in Eclipse platform with the plug-ins EMF and GMF. Figure 1 presents the action view model for the Theme-SPL after applying the previous heuristics.

#### 3. Related work

Weston et al. [4] present an approach which helps to construct feature models from a set of documents where requirements are expressed in natural language. The feature models obtained can be complemented with other requirements models, to better illustrate the features behavior and particular kinds of relationships between features and other elements of those new models. This approach is more adequate to text based requirements, while ours relies on model specifications.

In [3] the authors present an approach based on UML to maintain the separation of features during the modeling, using a composition language based on graph transformation (which leads to an easy re-use of feature models), and an approach to detection of undesirable structural interactions between different features models. Here heuristics are not defined to help obtain a feature model.

#### References

- [1] Clarke S., Baniassad E., L. A. *Aspect Oriented Analysis and Design*, Addison-Wesley Professional, April 2005.
- [2] Clements P., Northrop L. *A Framework for Software Product Line Practice*, 1st edition Boston, MA, USA: Addison-Wesley, 2002.
- [3] Jayaraman P, Whittle J, Elkhodary AM, Gomaa H., Model Composition and Feature Interaction Detection in Product Lines Using Critical Pair Analysis, MoDELS 2007, LNCS, Springer, 2007.
- [4] Weston N., Chitchyan R., and Rashid A., A Framework for Constructing Semantically Composable Feature Models from Natural Language Requirements. SPLC, USA, 2009.