Goals and Scenarios to Software Product Lines: the GS2SPL Approach

Gabriela Guedes, Carla Silva, Jaelson Castro

Centro de Informática – Universidade Federal de Pernambuco (UFPE)
CEP 50740-540, Recife/ PE – Brasil
{ggs, ctlls, jbc }@cin.ufpe.br

Abstract. Goal-oriented requirements engineering (GORE) approaches for Software Product Lines (SPLs) offer a natural way to capture not only stakeholders' goals but also similarities and the variability of an SPL. Goals to Software Product Lines (G2SPL) is an approach that guides the systematic creation of an SPL feature model from i* models with cardinality. However, by using G2SPL it is not possible to specify the behavioral characteristics of an SPL. In order to capture the system's behavior, it is common to use a scenario specification technique. In this paper, we present GS2SPL (Goals and Scenarios to Software Product Lines), an approach for the Requirements Engineering phase of SPL development that combines G2SPL and PLUSS use case scenarios. Our approach also includes a sub-process for configuring specific applications of an SPL based on the priority given to non-functional requirements.

Keywords: Requirements Engineering, Software Product Lines, Goal Models, Feature Model, Scenarios.

1 Introduction

In Requirements Engineering (RE) for Software Product Lines (SPL), feature models are used to capture similarities and the variability of product families. However, according to Silva, Borba and Castro [1] it is a challenge to establish a relationship between features of a software product and stakeholders' goals, since feature models do not capture which stakeholder's need originated each feature.

In this context, some Goal-Oriented Requirements Engineering (GORE) approaches were proposed to model requirements variability in SPL [2-5], these approaches can trace a relationship between features and goals. A comparison of them presented in [6], motivated the definition of the G2SPL (Goals to Software Product Lines) approach [1]. It relies on i*-c (i* with cardinality) language, which is used to (i) structure requirements according to stakeholders' intentions for the SPL, (ii) facilitate the gathering of features that define the SPL and (iii) aid the configuration of an individual product.

However, none of the approaches compared in [6], nor G2SPL captures dynamic or behavioral aspects of the SPL. This could be done using a scenario specification technique. Scenarios describe the behavior of the system functionality and are widely

used in requirements engineering because stakeholders easily understand them [7]. PLUSS (Product Line Use case modeling for Systems and Software engineering) [8] is an SPL approach that combines feature models and use case scenarios. It captures both common and variable behavior of the SPL. In PLUSS, both use cases and scenario steps are annotated with the features to which they are related.

In this paper, we present a requirements engineering approach for SPL that combines goal models, feature models and use case scenarios. The combination of these three models should provide a more complete requirement specification of the SPL, modelling stakeholders' goals, the SPL's functionality and its behavior.

2 Objectives of the Research

The main goal of our study was to define a requirements engineering approach for SPL that integrates i* models, features models and use case scenarios. Moreover, this approach should provide guidelines to derivate feature models and use case scenarios with variability from i* models.

Our goal was achieved by extending G2SPL [1], an approach where the feature model of an SPL is generated from i*-c (i* with cardinality) models. We have added new activities to guide the generation of use case scenarios with variability from i*-c models. We have also added a sub-process for configuring the SPL's artifacts for a specific product. This new approach was called GS2SPL [9] (Goals and Scenarios for Software Product Lines).

3 Scientific Contributions

The GS2SPL process consists of eight activities, most of them are part of the Domain Engineering process and only the last one is part of the Application Engineering process. The first four activities were inherited from G2SPL [6], the rest of the process consists in the addition of new activities or adaptation of G2SPL activities. The GS2SPL process is explained below:

1- Creation of SR (Strategic Rationale) Model: This activity consists of modeling stakeholders' goals using i* framework and it is optional if the SR model is already available. The output of this activity is a SR Model.

2- Identification of Candidate Elements To Be Features: The Domain Engineer identifies the elements of the SR Model that could represent features. According to Silva et al. [6], features are extracted from Tasks and Resources. Therefore, all internal tasks and resources of the actor that represents the SPL should be highlighted, as well as task and resource dependencies connected to this actor.

3- Reengineering the SR Model: in this activity, we add cardinality to the SR model. Cardinality may be added to intentional elements and to means-end relationships in which the root element (*end*) has more than one sub-element (*means*). The output is a SR model with cardinality. Fig. 1 presents part of the SR model with cardinality for Mobile Media [10], an SPL that will be used in this paper as a running example. The main purpose of Mobile Media is to manage media files in mobile devices.

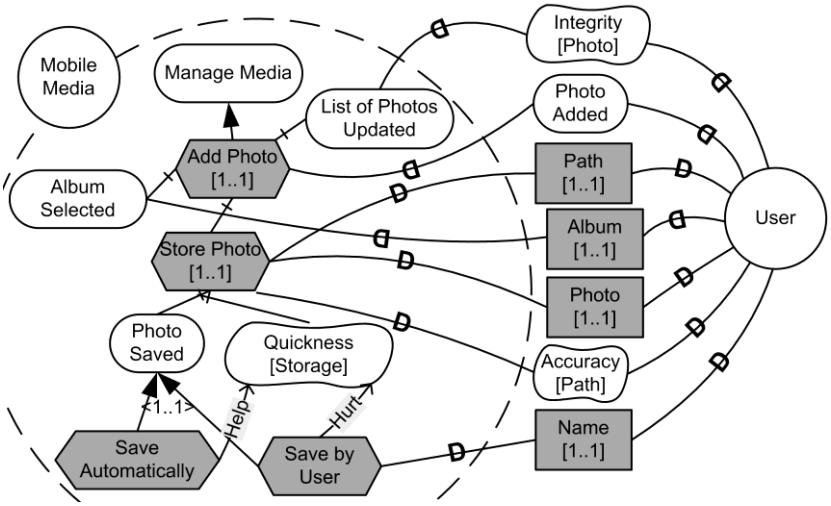

Fig. 1 i*-c model of Mobile Media

4- Elaboration of the Feature Model: This activity is concerned with the derivation of the SPL's feature model, this derivation uses the SR model with cardinality and is guided by the application of some heuristics. According to the heuristics defined in this activity, optional features are obtained from elements with cardinality [0..1], while mandatory features are obtained from elements with cardinality [1..1]. Elements involved in a means-end relationship with cardinality become alternative features with equivalent cardinality. Fig. 2 depicts the FM obtained for Mobile Media.

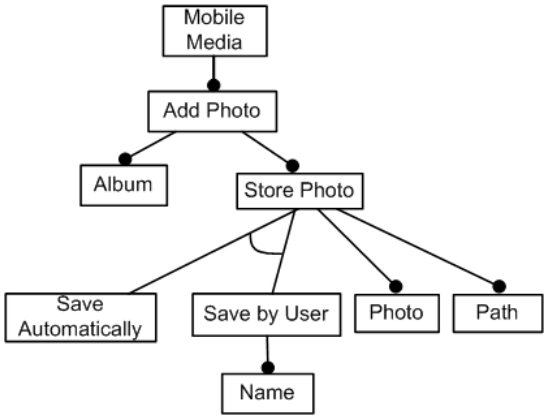

Fig. 2 Feature Model of Mobile Media

5- Feature Model Refinement: This is an optional activity and it is executed if the feature model needs to be reorganized or if there are that were not captured in the SR

model. If the feature model has repeated features, sub-features with more than one parent or different features with the same meaning, reorganization is required. This activity can be performed as many times as the domain engineer believes it is necessary. Our running example is quite simple and did not require the execution of this activity.

6- Elaboration of Use Case Scenarios: The SPL use case scenarios are specified according to an adaptation of the guidelines defined by Castro et al. [11]. This activity uses the SR Model with cardinality and the feature model as input to generate the PLUSS [8] scenarios description of an SPL. The guidelines proposed by Castro et al. in [11] are a mapping between i* models and use case scenarios that are not specific for dealing with SPL variability. We propose guidelines to map i*-c models to PLUSS use case scenarios.

The guidelines are divided in three steps. Step 1 (Discovering Actors) is composed by Guidelines 1 to 5, that determine which i* actors should be mapped to use case actors. Basically, i* actors that have dependencies with the SPL actor should be mapped to use case actors, unless all dependencies between them are softgoal dependencies. In our example, there is only one external actor, "User", and, according to the presented guidelines, it can be mapped to a use case actor.

Step 2 (Discovering use cases for actors) is composed by Guideline 6, that analyzes dependencies between the actor that represents the SPL and those i* actors that were mapped to use case actors in order to determine which dependencies should be mapped to use cases. In summary, we map goal dependencies to use cases; task and resource dependencies are mapped to use cases if they require many steps; and, finally, softgoal dependencies cannot be mapped to use cases, because they represent non-functional requirements. Applying Step 2 to the example, we discovered that only the "Photo Added" goal dependency can be mapped to a use case.

Step 3 (Discovering and Describing Use Case Scenarios) is composed by Guidelines 7 to 12, that guide the elaboration of scenarios descriptions through the analysis of the intentional elements and their relationships inside actors' boundary. In summary, sub-elements of task decomposition links are mapped to primary scenario steps, while sub-elements of means-end links are mapped to alternative steps (creating alternative scenarios). The cardinality of intentional elements must be analyzed to determine if the step derived from the element is mandatory or optional. Applying Step 3 to the Mobile Media example, we obtained the description for the "Add Photo" use case (Table 1).

7- Use Case Scenarios Refinement: Scenarios obtained on the previous activity may be succinct or written on a very high level; it will depend on the level of refinement achieved in the SR model. Hence, we suggest the refinement of scenarios descriptions until they reach the desired level of details.

8- Product Configuration: This is the configuration sub-process and it will be executed every time a new product of the SPL has to be derived. It represents the Application Engineering process of GS2SPL and consists of three activities:

Choice of Specific Configuration: Here the client chooses the goals for the new product. Depending on the client's choices, there may be more than one possible product configuration. In our running example, there are two alternatives: one with "Save Automatically" task (A1) and another with "Save by User" task (A2).

Table 1 "Add Photo" use case scenario

Use Case 1: Add Photo

CHARACTERISTIC INFORMATION

Primary Actor: User
Feature: Add Photo
Scope: MobileMedia
Pre-condition: -
Success Condition: Photo added to album

PRIMARY SCENARIO

ID	User Action	System Response
1	Select "Add Photo" option [Add Photo]	
2	Select album [Album]	
3	Provide path of photo [Path]	
4	Select photo to be added [Photo]	
5	-	Photo is automatically saved [Save Automatically]
5	Choose for photo [Name] [Save by User]	Photo is saved with the chosen name
6	-	List of photos is updated

SECONDARY SCENARIOS

RELATED INFORMATION

Non-functional requirements: Integrity [Photo], Accuracy [Path], Quickness [Storage]

Prioritization of Variants: Alternatives obtained according to the client's choices are ranked based on the priority the client gave to the softgoals (modelled in the SR model). The alternative with the highest priority value represents the most suitable configuration for the client's desires. The function to calculate the priority of each variant will not be presented in this paper due to the lack of space.

Product Artifacts Configuration: First, the configuration model is generated by eliminating, from the FM, all features that are not related to elements in the SR model of the chosen alternative. Then, all cardinality indications must be removed from the SR model, thus the i* model of the product is obtained. Finally, only use cases that are related to selected features will be present on the product's artifacts. Scenario descriptions must also be configured by eliminating the steps that are annotated with features that were not chosen.

4 Conclusions

In this paper we presented GS2SPL (Goals and Scenarios to Software Product Lines), a GORE approach for the requirements engineering phase of SPL development.

GS2SPL guides the creation of an i*-c model for a software product line, which is used to systematically generate the SPL's feature model and then its use case scenarios.

The advantage of using GS2SPL is that the most relevant features and use cases for satisfying the stakeholders' goals are obtained in a systematic way from the i*-c model. GS2SPL also provides a sub-process that guides the configuration of the SPL's requirements models for a specific product, that is based on the softgoals' priority. Unfortunately, our approach does not have a supporting tool yet, making it difficult for its adoption in an industrial context.

5 Ongoing and Future Work

As future work, we plan to: (i) perform an empirical validation of GS2SPL to evaluate its strengths and weaknesses; (ii) develop tool support for our approach; (iii) investigate how to identify feature model constraints from i* models; and (iv) investigate how to take feature interactions into account when generating use case scenarios.

References

1. Silva, C., Borba, C., Castro, J.: A Goal Oriented Approach to Identify and Configure Feature Models for Software Product Lines. In: Proc. of the WER'11, Rio de Janeiro, Brazil (2011)
2. Yu, Y., Leite, J.C.S.P., Lapouchnian, A., Mylopoulos, J.: Configuring features with stakeholder goals. In: Proc. of the ACM SAC'08, Fortaleza, Brazil, pp. 645-649 (2008)
3. Mussbacher, G., Amyot, D., Araújo, J., Moreira, A. Modeling Software Product Lines With AoURN. In: Ws on Early Aspects at AOSD'08, Brussels, Belgium. ACM (2008)
4. Silva, C., Alencar, F., Araújo, J., Moreira, A., Castro, J.: Tailoring an Aspectual Goal-Oriented Approach to Model Features. In: Proc. of the 20th Intl. Conf. on Software Engineering and Knowledge Engineering (SEKE'08), San Francisco Bay, USA (2008)
5. Silva, L., Batista, T., Soares, S., Santos, L.: On the Role of Features and Goals Models in the Development of a Software Product Line. In: Ws on Early Aspects at 9th Annual Aspect-Oriented Software Development Conference (AOSD'10), Rennes, France (2010)
6. Borba, C., Silva, C.: A comparison of goal-oriented approaches to model software product lines variability. In: LNCS, Vol. 5833, pp. 244-253, Springer-Verlag (2009)
7. Maiden, N, Alexander, I.: Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle. 1 ed., Wiley (2004)
8. Eriksson, M., Börstler, J., Borg, K.: Managing requirements specifications for product lines – an approach and industry case study. In: Journal of Systems and Software, 82(3), 435-447 (2009)
9. Guedes, G., Silva, C., Castro, J., Soares, M., Dermeval, D., Souza, C.: GS2SPL: Goals and Scenarios to Software Product Lines. In: Proc. of the SEKE'12, Redwood City, USA, pp. 651-656 (2012)
10. Figueiredo, E. et al.: Evolving software product lines with aspects: an empirical study on design stability. In: Proc. of the 30th International Conference on Software Software Engineering (ICSE'08), Leipzig, Germany, pp. 261-270 (2008)
11. Castro, J., Alencar, F., Santander, V., Silva, C.: Integration of i* and Object-Oriented Models. In: Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J. (eds). Social Modeling for Requirements Engineering. 1st Ed., MIT Press, pp. 457-483 (2011)