Goals and Scenarios for Requirements Engineering of Software Product Lines

Gabriela Guedes[1], Carla Silva[2], Jaelson Castro[1]

[1] Centro de Informática – Universidade Federal de Pernambuco (UFPE)
CEP 50740-540, Recife/ PE – Brasil
{ggs, jbc}@cin.ufpe.br
[2] Departamento de Ciências Exatas, Centro de Ciências Aplicadas e Educação
Universidade Federal da Paraíba (UFPB) – CEP 58297-000, Rio Tinto/ PB – Brasil
carla@dce.ufpb.br

Abstract. Goal-oriented requirements engineering (GORE) approaches offer a natural way to capture similarities and the variability in software product lines (SPLs) development. Besides, they can effectively capture both the stakeholders' objectives and the system requirements. From i* models, for example, it is possible to systematically obtain feature models. To complement the requirements specification of SPLs, their behavioral characteristics can be captured by using a scenario specification technique. This paper presents a process. An extension of i* that includes cardinality is used in connection with feature models and a use case scenarios to support the requirements engineering phase in SPLs development. This process also includes activities to aid the configuration of requirements artifacts for a specific product in the SPL. The paper also presents the case study being used to illustrate the proposed process.

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

1 Introduction

Requirements Engineering (RE) is the phase of software development concerned with producing a set of software systems specifications that satisfy the *stakeholders* needs and can be implemented, deployed and maintained [1].

In RE for Software Product Lines (SPL), feature models are used to capture similarities and the variability of product families. However, according to Borba and Silva [2] it is a great challenge to establish a relationship between features of a software product and the objectives of the stakeholders. In this context, we proposed a Goal-Oriented Requirements Engineering (GORE) approach that provides a systematic way to discover the features that will be part of a SPL and also allows the systematic selection of the features for a particular product [3].

It is worth to complement the requirements specification obtained with this GORE approach. The dynamic aspect of a SPL may be described by a scenario specification technique. Scenarios describe the behavior of the system functionality and are widely used in requirements engineering because they are easily understood by stakeholders [4]. In this paper, we present a process that integrates a GORE approach for SPL, feature modeling and a scenario specification technique.

2 Objectives of the Research

Many goal-oriented approaches were proposed to model requirements variability in SPL [5, 6, 7, 8]. A comparison of these approaches was presented in [2] and motivated the definition of the G2SPL (Goals to Software Product Lines) approach [3]. It relies on the i*-c (i* with cardinality) language, which is used to (i) structure requirements according to the stakeholders intentions for the SPL, (ii) facilitate the gathering of the features that define the SPL and (iii) aid the configuration of an individual product.

In SPL, specifying non-trivial features can cause the scattering of the SPL variation points on the line's artifacts. Moreover, some feature specifications combine, in their artifacts, information from the SPL variants and the product configuration. The scattering and tangling of features related concerns can also be observed in the scenario specifications of the SPL. These concerns are, therefore, crosscutting and may compromise the maintainability and understanding of the SPL artifacts [9].

Crosscutting concerns are requirements which may impact multiple modules or components. Thus, the crosscutting concerns (representing functional or non-functional requirements) are properties that affect various parts of the system. The importance of their proper handling is evident. We must take into account the way in which the crosscutting concerns interact with other concerns, otherwise there is the risk that the nature of these interactions only becomes clear in later stages of software development. This can cause a higher cost in solving problems related to the system evolution and maintenance [10].

One of the studies concerned with the separation of crosscutting concerns in scenario specifications is the technique MSVCM (Modeling Scenario Variability as Crosscutting Mechanisms) [9]. This technique improves the separation of concerns between the variability management and the scenario specifications of the SPL. It deals with scenario variability as a composition of different artifacts such as use case specifications, feature models, product configuration and configuration knowledge.

Another study that is concerned with the separation of crosscutting concerns in scenario specifications is MATA (Modeling Aspects using a Transformation Approach) [10]. MATA is an aspect-oriented modeling approach that uses graph transformations for specifying and composing aspects. Scenario specification in MATA is performed as follows: a non aspectual base scenario may be specified by a sequence diagram, while an aspectual scenario is described by a sequence diagram enhanced with roles. These roles work as variables that must be instantiated when the aspectual scenario and the base scenario are composed. Recently, MATA was integrated with a GORE approach to obtain a systematic identification of crosscutting concerns in the use case scenario specification [11].

This paper proposes the definition of a RE process for SPL that integrates a GORE technique and a scenario specification technique with separation of crosscutting concerns. In particular, we are extending the G2SPL approach to include activities related to the generation and configuration of scenarios specifications for SPL.

3 Scientific Contributions

The extended G2SPL process, shown in Fig. 1, was modeled using the BPMN (*Business Process Modeling Notation*) [12]. It consists of eight activities, explained as follows:

1. Creation of the SR (Strategic Rational) Model: this activity consists of modeling the stakeholders' goals using i* framework. The output of this activity is a SR Model.
2. Identification of the Candidate Elements to be Features: in this activity, the Domain Engineer identifies the elements of the SR Model that could represent features. According to Silva et al. [3], features can be extracted from Tasks and Resources.
3. Reengineering the SR Model: in this activity, cardinality is added to the SR model. Restructuring is based on some heuristics tailored for i*-c language [3]. The output is a SR Model with cardinality.
4. Elaboration of the Feature Model: this activity is concerned with the derivation of the Feature Model of a SPL. The input artifacts are some heuristics and the SR Model with cardinality and the output is the Feature Model.
5. Reorganization of the Feature Model: this activity is considered optional. If the feature model has repeated features, sub-features with more than one father 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 [3].
6. Elaboration of the Use Case Scenarios: the SPL use case scenarios are specified according to an adaptation of the guidelines defined by Castro et al. [13]. This activity uses the SR Model with cardinality as input and the output is the use case scenarios of the SPL.
7. Generation of Use Case Scenarios with Separation of Crosscutting Concerns: in this activity, both the use case scenario specification and the feature model are used to generate use case scenarios with separation of crosscutting concerns. In order to accomplish this, we should choose between the MSVCM (Modeling Scenario Variability as Crosscutting Mechanisms) [9] and the MATA (Modeling Aspects Using a Transformation Approach) techniques [10].
8. Configuration of the Product Artifacts: the purpose of this activity is the derivation of the artifacts for a specific product of the SPL. The outcomes of this activity are the use case scenario description, the configuration model (containing the chosen features) and the SR model of a particular product.

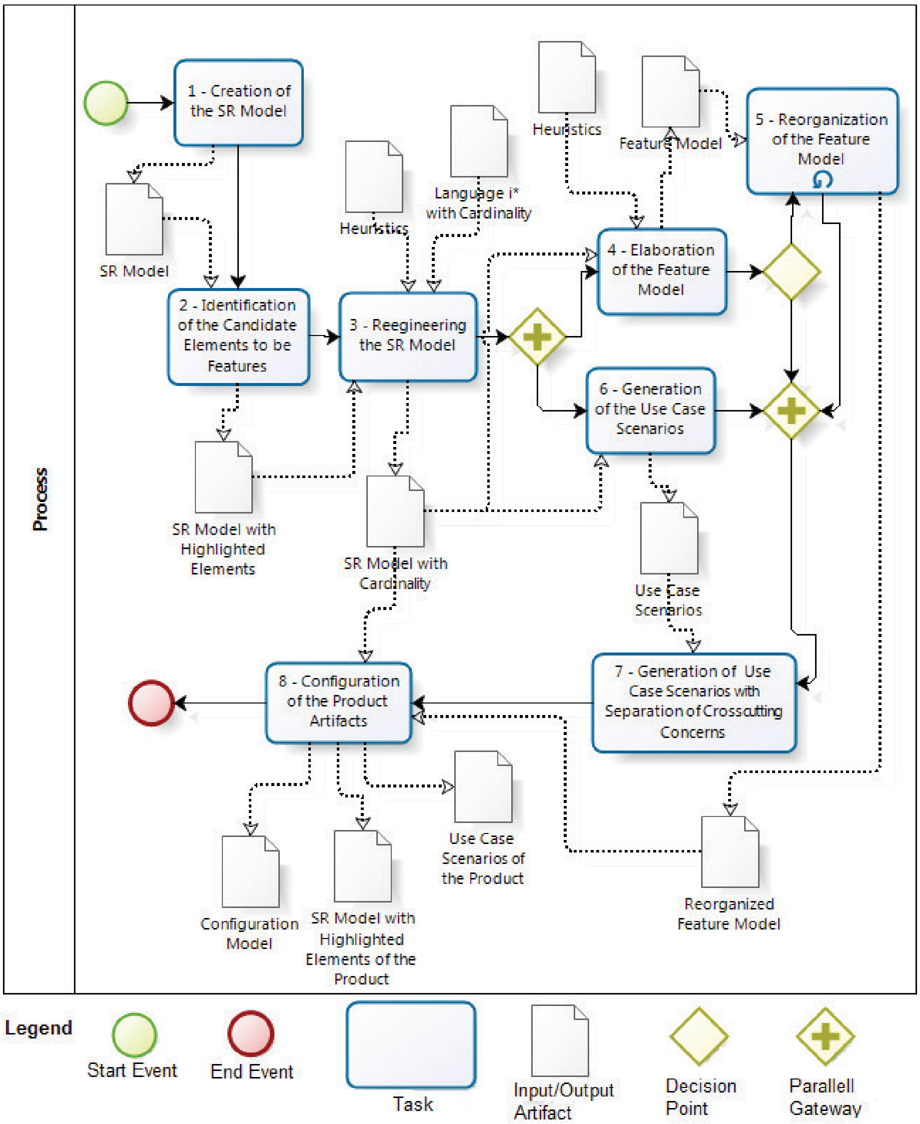

Fig. 1 Extended G2SPL process model

3.1 Case Study

We chose the Motorola TaRGeT (Test and Requirement Generation Tool) project [14] as our case study. TaRGeT is a SPL whose products are tools that automatically generate tests suites from scenario specifications written in a given template. In this case, the productivity is increased, since it is only necessary to generate Tests Suites from the Scenarios Description.

The SR model of TaRGeT SPL is shown in Fig. 2. Note that we are using the i*-c notation to represent some optional elements. The optional elements are “Detect Scenario Changes and Update Test Cases” and “Verify Scenarios Syntactically” tasks, since they have cardinality [0..1]. The tasks involved in means-end relationships are optional too.

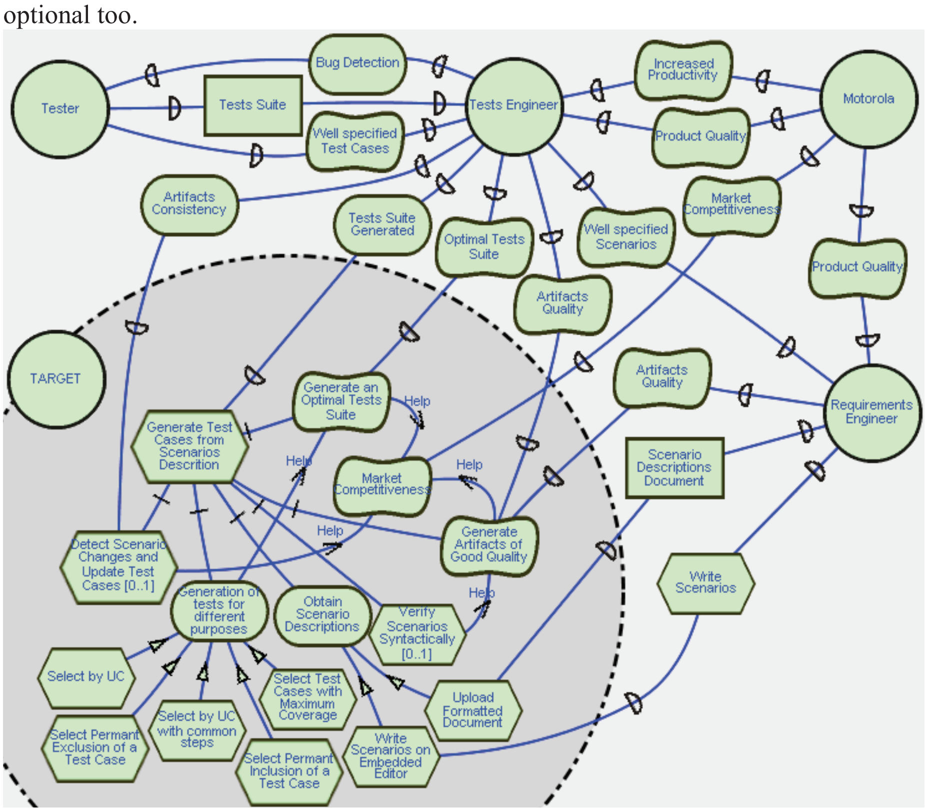

Fig. 2 TaRGeT SR model

4 Conclusions

We presented a process to support RE of SPL in regard of the elaboration of requirements artifacts. The proposed process aims to (i) provide the development of more complete requirements artifacts, (ii) enable the systematic construction of model features, (iii) allow the systematic generation of artifacts (goal models, feature models and scenarios specification) for a specific product, and (iv) support the systematic configuration of the artifacts of a product.

Regarding to the case study, we have performed the first three activities of the process and, as a result, we have produced the TaRGeT SR Model (see Fig.2).

5 Ongoing and Future Work

So far, we have held meetings with members of TaRGeT project for requirements elicitation and validation purposes. Currently, we are carrying out the remaining activities of the process. As future work, we suggest the development of a tool to support the whole process, since only two activities have tool support ("Creation of the SR Model" and "Elaboration of the Use Case Scenarios") [15, 16].

References

1. Lamsweerde, A.: Goal-Oriented Requirements Engineering: A Guided Tour. In: RE '01: Proceedings of the Fifth IEEE International Requirements Engineering Conference. Washington, DC, USA. IEEE Computer Society, p.249-263, 2001.
2. 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.
3. 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)
4. Maiden, N, Alexander, I.: Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle. 1 ed., Wiley (2004)
5. 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)
6. 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)
7. 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)
8. 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)
9. Bonifácio, R. and Borba, P.: Modeling Scenario Variability as Crosscutting Mechanisms. In: AOSD'09, Charlottesville, Virginia, USA (2009)
10. Whittle, J., Araujo, J.: Scenario modelling with aspects. Software, IEE Proceedings. 151(4): p. 157-171 (2004)
11. Oliveira, C., Araújo, J., Silva, C. Integração de KAOS com Cenários Aspectuais (In English: Integration of KAOS with Aspectual Scenarios). In: XV Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2010), Valencia, Spain (2010)
12. Business Process Modeling Notation, V1.1. OMG Available Specification, 2008. Available at: http://www.omg.org/spec/BPMN/1.1/PDF/. Last access: June 2011.
13. 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, 2011. Chapter 13, pp. 457-483
14. Test Research Project (Motorola Brazil Test Center), CIn-UFPE/Brazil and UFCG/Brazil, http://twiki.cin.ufpe.br/twiki/pub/LabPS/ModulosApredizagem/TreinamentoTaRGeT.pdf
15. IStar Tool - IStar modeling Tool support. Available at: http://portal.cin.ufpe.br/ler/Projects/IStarTool.aspx. Last access: June 2011.
16. Vicente, A., Santander, V., Castro, J., Freitas, I., Matus, F.: JGOOSE: A Requirements Engineering Tool to Integrate I* Organizational Modeling with Use Cases In UML. Ingeniare. Revista Chilena de Ingeniera 17(1) (2009) 6-20. Available at: http://andvicente.googlepages.com/aav(undergraduate). Last access: June 2011.