DyMMer-NFP: Modeling Non-functional Properties and Multiple Context Adaptation Scenarios in Software Product Lines

Anderson G. Uchôa[1(✉)], Luan P. Lima[1], Carla I.M. Bezerra[1,3], José Maria Monteiro[2], and Rossana M.C. Andrade[2,3]

[1] Federal University of Ceará, UFC Quixadá, Quixadá, Brazil
andersonguchoa@gmail.com, luan_pereira_lima@hotmail.com, carlailane@ufc.br

[2] Computer Science Department, Federal University of Ceará, UFC Fortaleza, Fortaleza, Brazil
{monteiro,rossana}@ufc.br

[3] Group of Computer Networks, Software Engineering and Systems (GREat), Federal University of Ceará, Fortaleza, Brazil

Abstract. In Software Product Lines (SPLs), the modeling of non-functional properties (NFPs) and context adaptation scenarios are important activities, once they make possible the identification of inter-dependencies constraints between functional requirements (FR) and NFP, according to a specific adaptation context scenario. However, there are few tools to help domain engineers to represent NFPs and context adaptation scenarios. To deal with this problem, we propose DyMMer-NFP, an extension of the DyMMer tool to support the modeling of NFPs and multiple contextual adaptation scenarios in feature models. DyMMer-NFP uses a catalog with 39 NFPs. Each NFP in this catalog were mapped according to each quality characteristic and sub-characteristics presented in the ISO/IEC 25010 SQuaRE product quality model. To specify the interdependencies between NFPs and features, DyMMer-NFP has used the concept of contribution links. In order to make it easier to evaluate DyMMer-NFP two datasets, called AFFOgaTO and ESPREssO, were made available for free.

Keywords: Feature models · Non-functional properties · Dynamic Software Product Lines

1 Introduction

One of the main concepts in Software Product Lines (SPLs) is the variability, which can be defined as the possibility of setting or as the ability of a system or software artifact has to be changed, customized or configured for a particular context [3]. SPLs treat variability using a design artifact called feature model,

R.M.C. Andrade—Research Scholarship - DT Level 2, sponsored by CNPq.

G. Botterweck and C. Werner (Eds.): ICSR 2017, LNCS 10221, pp. 175–183, 2017.
DOI: 10.1007/978-3-319-56856-0_12

which captures the similarities and variabilities between the possible configurations of products in a particular domain [9]. This artifact represents the variations in software architecture. In SPLs, variants are typically set during the SPL development cycle at design time. However, recently, Dynamic Software Product Lines (DSPLs) were proposed as a way to promote runtime variability, where variants can be selected and change their behavior at runtime [3]. DSPL aims to produce software capable to adapt according to the needs of users and resources constraints at runtime. In a DSPL, variation points are firstly bound when the software is released, matching initial environment settings. However, at runtime, whether the context changes, it is possible to rebind the variation points, in order to adapt the system to the new, changed, environment settings [4]. Thus, in a DSPL feature model, the features can be added, removed and modified at runtime in a managed way [1].

Nevertheless, feature models do not capture Non-Functional Properties (NFP) explicitly neither influence these properties to achieve alternative configurations of a product variant. Representing NFP in SPL or DSPL might be a rather complex activity. Given a particular context, configuration rules, features constraints and preferences of stakeholders, must all be considered [10]. Although the NFP represent an important aspect related to the quality of a SPL, modeling NFP techniques are not completely suitable for SPL engineering, as indicated in an extensive literature survey [7].

A major challenge in DSPLs is to enable the support of scenarios with context adaptation and non-functional properties (NFP). The modeling of multiple context adaptation scenarios in DSPL feature models is an important mechanism for DSPL engineering, specifically in Domain Engineering where system features must be specified to accommodate context variations, and therefore, depending on the context, one set of features will be activated and the other will be deactivated [5].

In our previous work, we developed a tool called, DyMMer [2], to automatically support the evaluation of SPL and DSPL feature models based on a catalog of 40 quality measures. Currently, DyMMer is able to import a feature model in XML format, represent visually a feature model and edit it, adding context adaptations and context rules to the activation and deactivation of features. Besides, DyMMer is also capable of computing 40 quality measures, in which 8 are specific for DSPLs and 32 can be used in both SPLs and DSPLs [2].

In this work, we present an extension of the DyMMer tool, called DyMMer-NFP, to support the modeling of NFPs and multiple contextual adaptation scenarios in feature models. DyMMer-NFP uses a catalog with 39 NFPs (showed in Fig. 1), identified initially by [11]. Next, the 39 NFPs in this catalog were mapped according to each quality characteristic and sub-characteristics presented in the ISO/IEC 25010 SQuaRE product quality model [8]. Finally, to specify the interdependencies between NFPs and features we have used the concept of goal-oriented modeling, in particular the concept of contribution links [12].

The remainder of this paper is organized as follows. Section 2 the DyMMer tool is presented in detail, which includes the main features and architecture.

Section 3 describe the main features added to the DyMMer-NFP and presents a NFPs catalog to assist in the identification and NFPs modeling in DSPL feature models. Section 4 describes how to use the DyMMer-NFP to NFPs modeling through a set of tasks. Finally, Sect. 5 the conclusion and future work are described.

2 Background

DyMMer (*Dynamic Feature Model Tool Based on Measures*) is a tool originally developed to extract measures from different feature models, which can be described using the file format (XML) proposed by the S.P.L.O.T[1] feature models repository. So, DyMMer receives as input a XML file and creates an in memory representation for this feature model. The DyMMer was developed using JAVA and offers a series of functionalities, organized in three main layers [2]:

- **Feature Model Viewer -** allows the visualization of feature models according to the selected context;
- **Feature Model Editor -** allows the insertion, exclusion and edition of context adaptations. DyMMer considers that context and non-context features are represented in a single feature model; and
- **Data Exportation -** allows to export the values generated from the quality measures applied to the feature models with and without context for a spreadsheet in *Microsoft Office Excel* format.

Although DyMMer presents functionalities that allow domain engineers to extract quality measures and to edit feature models, it does not support the addition and removal of features and cardinality groups. Besides, DyMMer does not support the managing of NFPs, as well as multiple context adaptations scenarios in a same feature model. Thus, in order to surpass the lack of these functionalities, we extend the DyMMer giving rise to a new tool called DyMMer-NFP. The DyMMer-NFP code and its documentation are available *online*[2]. The new features incorporated by DyMMer-NFP are presented in the next section.

3 DyMMer-NFP Main Features

In this section we present the main features added in the DyMMer-NFP.

Creating a Feature Model. In addition to importing existing feature models which must be described in the S.P.L.O.T format, DyMMer-NFP makes it possible to create new feature models using the FODA method notation [6]. It is possible to create mandatory and optional features, XOR and OR cardinality groups, as well as inclusion and exclusion constraints (see Fig. 2. ③).

[1] S.P.L.O.T - http://www.splot-research.org/.
[2] https://github.com/anderson-uchoa/DyMMer.

Modeling Non-functional Properties. DyMMer-NFP makes it possible to represent NFPs in a feature model. For this, we have used a NFPs catalog (showed in Fig. 1), identified initially by [11]. Next, the 39 NFPs in this catalog were mapped according to each quality characteristic and sub-characteristics presented in the ISO/IEC 25010 SQuaRE product quality model [8] (see Fig. 2 ①). This catalog works as a guide for the identification of NFPs that emerge at runtime. Thus, the NFPs catalog was organized in several levels. The 1^{st} level represents the root of our classification schema. The 2^{nd} level is composed of each quality characteristic and the 3^{rd} level represents the quality sub-characteristics related to each quality characteristic. Finally, the 4^{th} level represents the NFPs identified in [11]. The quality characteristics mapped in this catalog are: *Functional suitability*, *Performance efficiency*, *Reliability*, *Security* e *Usability*. Currently, the DyMMer-NFP supports the modeling of 39 NFPs, which can also be used in both SPLs and DSPLs feature models.

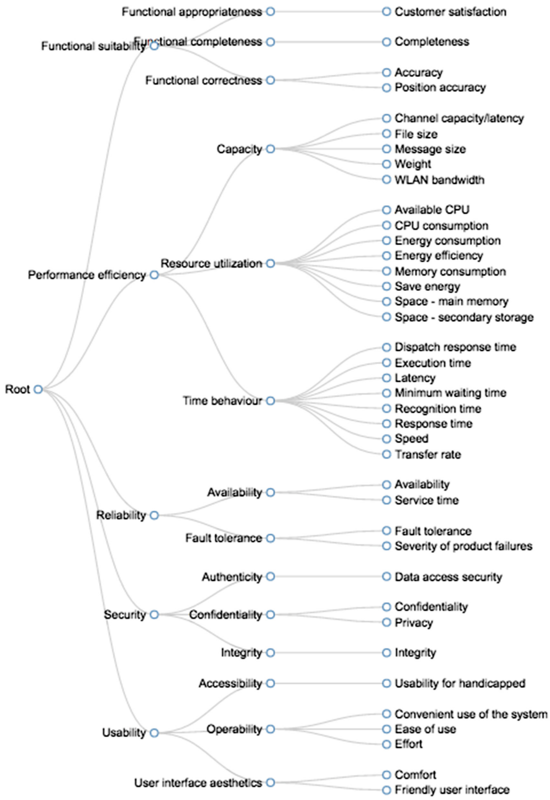

Fig. 1. The NFPs catalog.

Defining Context Adaptation Scenarios. DyMMer-NFP makes it possible to edit a specific feature model, adding or removing context adaptation scenarios. The addition of context adaptation scenarios is represented by a hierarchical list consisting of two specification levels (see Fig. 2. ②). Thus for each added context adaptation scenario a domain engineer must add context information with their respective variations, specify which features should be activated or deactivated, and define constraints. A DSPL feature model may have one or more context adaptation scenarios. DyMMer-NFP enables the domain engineer to handle DSPLs feature models. It is important to emphasize that the S.P.L.O.T does not support DSPLs modeling.

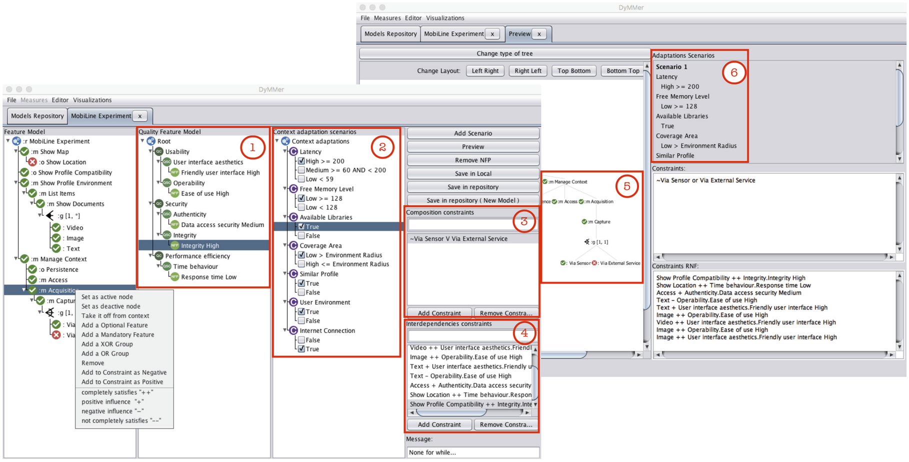

Fig. 2. An overview of the DyMMer-NFP tool: ① quality feature model editor, ② context adaptation scenarios editor, ③ cross-tree constraints editor, ④ interdependencies constraints editor, ⑤ visualizing feature model configuration, ⑥ selecting context adaptation scenarios.

Specifying Interdependencies Constraints Between Features and NFPs. DyMMer-NFP makes it possible to specify the interdependencies between NFPs and features (see Fig. 2. ④). To specify this constraints we have added a concept of goal-oriented modeling, in particular the concept of contribution links [12]. In this way, we assign interdependence constraints for each feature in a given context. These features may have four types of interdependence constraints over an NFP:

- **"++" -** the feature completely satisfies an NFP if it is activated;
- **"- -" -** the feature does not completely satisfy an NFP if it is activated;
- **"+" -** the feature has a positive influence on an NFP if it is activated; and
- **"-" -** the feature has a negative influence on an NFP if it is activated.

Visualizing Feature Model Configuration. DyMMer-NFP makes it possible to view the feature model configuration (see Fig. 2 ⑤–⑥). The visualization of the feature model configuration can be done in two ways: (i) after adding a context adaptation scenario in editing; and (ii) by means of the feature model viewer layer. The visualization of the feature model configurations after adding a context adaptation scenario, for example, is possible through the preview functionality, allowing the domain engineer to easily observe the feature model configuration during its editing. In this way, after each addition of a new context adaptation scenario the domain engineer can view one or more feature model configurations.

4 Using DyMMer-NFP to Modeling Non-functional Properties

The use of the DyMMer-NFP to modeling NFPs in feature models consists in running a set of tasks showed in Fig. 3 and in the tool demonstration video[3]. These task are discussed next.

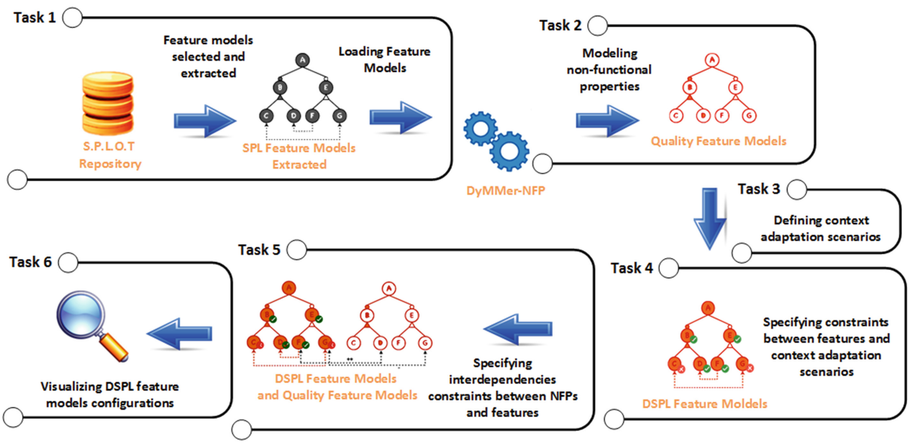

Fig. 3. Using DyMMer-NFP.

Task 1 - Importing Feature Models from S.P.L.O.T. The first task is to import a feature model from S.P.L.O.T repository or create a new feature model. In order to make it easier to evaluate DyMMer-NFP two datasets can be used, they are: (i) AFFOgaTO dataset is composed of a set of 218 feature models fetched from S.P.L.O.T repository; and (ii) ESPREssO dataset is composed of a set of the 30 DSPLs feature models extracted from literature and represented in the XML format, following the S.P.L.O.T specifications. The AFFOgaTO[4]

[3] Demonstration video - https://youtu.be/FCn0zEfAEBs.
[4] AFFOgaTO - https://goo.gl/gye5ma.

and ESPREssO[5] datasets are available for free download and can be used by the software engineering community.

Task 2 - Modeling Non-functional Properties. After to import or create a new feature model, we must represent the non-functional properties that are relevant for the feature model. These NFPs are specified according to the NFPs catalog and represented visually in DyMMer-NFP.

Task 3 - Defining Context Adaptation Scenarios. Next, a set of context adaptation scenarios can be defined for a given feature model. Context adaptation scenarios are added in a hierarchical list, where context information are added with their quantifiers and qualifiers. These quantifiers are defined by relational operators: greater than (>), less than (<), greater than or equal to (>=), less than or equal to (<=), equal (=) and different (<>). Followed by a value of type: *string*, integer, *float* or boolean. If this quantifier is defined by a numeric range, a logical operator is added that can be of the type: (OR) or (AND), followed by another value of type *string*, integer, *float* or boolean.

Task 4 - Specifying Constraints Between Features and Context Adaptation Scenarios. The next task is to analyze and specify the constraints between features and context adaptation scenarios. This task aims to verify which features should be activated and deactivated in a given context adaptation scenario. To perform this task we must specify which features are activated and deactivated for each context adaptation scenario.

Task 5 - Specifying Interdependencies Constraints Between NFPs and Features. After specifying the context adaptation scenarios, that is, which features are activated and deactivated in each context adaptation scenario, the interdependencies constraints between NFPs and the context features can be defined. Thus, for each context adaptation scenario, the user needs to specify the contribution links between each feature, activated or deactivated in this current scenario, and one or more NFPs.

Task 6 - Visualizing DSPL Feature Models Configurations. After modeling the context adaptation scenarios and the interdependencies constraints between NFPs and context features, we can visualize the configuration of a specific feature model.

5 Conclusion and Future Work

This paper presents DyMMer-NFP, an extension of the DyMMer tool to support the modeling of NFPs and multiple contextual adaptation scenarios in feature models. DyMMer-NFP uses a catalog with 39 NFPs (showed in Fig. 1), identified initially by [11]. Next, the 39 NFPs in this catalog were mapped according to each quality characteristic and sub-characteristics presented in the ISO/IEC

[5] ESPREssO - https://goo.gl/ONfTL3.

25010 SQuaRE product quality model [8]. Finally, to specify the interdependencies between NFPs and features we have used the concept of goal-oriented modeling, in particular the concept of contribution links [12]. The modeling of multiple context adaptation scenarios in DSPL feature models is at the same time an important mechanism and a challenging task. So, DyMMer-NFP can help domain engineers in creating better adaptive software, guiding the specification of the feature model, besides making it easier to model the NFPs established by the stakeholders and to represent the relationship (including constraints) between features and NFPs. Thus, DyMMer-NFP enables the analysis of the impacts that a feature have on one or more NFPs in different context adaptations, ensuring that products derived from the DSPL continue serving in a satisfactory way the NFPs previously established. In this way, the domain engineers are guided to build DSPLs that satisfies the specified NFPs: adding features that increases the possibility of achieving a particular NFP and removing (or minimizing) features that reduces this probability, ensuring an improvement in the quality of the DSPL. As future work, we intend to formally verify whether the NFPs in a specific DSPL feature model match the expected results of the process of runtime reconfiguration. In addition, we plan to add new NFPs in the catalog and to identify thresholds for each NFP.

References

1. Bencomo, N., Hallsteinsen, S., De Almeida, E.S.: A view of the dynamic software product line landscape. Computer **45**(10), 36–41 (2012)
2. Bezerra, C.I., Barbosa, J., Freires, J.H., Andrade, R., Monteiro, J.M.: DyMMer: a measurement-based tool to support quality evaluation of DSPL feature models. In: Proceedings of the 20th International Systems and Software Product Line Conference, pp. 314–317. ACM (2016)
3. Bosch, J.: Software variability management. In: Proceedings of the 26th International Conference on Software Engineering, pp. 720–721. IEEE Computer Society (2004)
4. Capilla, R., Bosch, J., Kang, K.C.: Systems and Software Variability Management. Springer, Heidelberg (2013)
5. Capilla, R., Bosch, J., Trinidad, P., Ruiz-Cortés, A., Hinchey, M.: An overview of dynamic software product line architectures and techniques: observations from research and industry. J. Syst. Softw. **91**, 3–23 (2014)
6. Czarnecki, K., Helsen, S., Eisenecker, U.: Formalizing cardinality-based feature models and their specialization. Softw. Process: Improv. Pract. **10**(1), 7–29 (2005)
7. Hammani, F.Z.: Survey of non-functional requirements modeling and verification of software product lines. In: 2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS), pp. 1–6. IEEE (2014)
8. ISO/IEC: 25010: Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models. Technical report, ISO/IEC, Switzerland (2011)
9. Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., Peterson, A.S.: Feature-oriented domain analysis (FODA) feasibility study. Technical report, DTIC Document (1990)

10. Sanchez, L.E., Diaz-Pace, J.A., Zunino, A., Moisan, S., Rigault, J.P.: An approach based on feature models and quality criteria for adapting component-based systems. J. Softw. Eng. Res. Dev. **3**(1), 1–30 (2015)
11. Soares, L.R., Potena, P., Carmo Machado, I., Crnkovic, I., Almeida, E.S.: Analysis of non-functional properties in software product lines: a systematic review. In: 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications, pp. 328–335. IEEE (2014)
12. Van Lamsweerde, A.: Goal-oriented requirements engineering: a guided tour. In: 2001 Proceedings of the Fifth IEEE International Symposium on Requirements Engineering, pp. 249–262. IEEE (2001)