ReMINDER: An Approach to Modeling Non-Functional Properties in Dynamic Software Product Lines

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

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

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

[3] Computer Science Department, Federal University of Bahia, UFBA Salvador, Salvador, Brazil
ivanmachado@dcc.ufba.br

Abstract. This paper presents a systematic approach to modeling NFPs in DSPL feature models. In our proposed approach, feature models are annotated with the representation of NFPs, rules for the activation and deactivation of features, constraints between NFPs and features, and context adaptation scenarios. To evaluate the applicability of the proposed approach we carried out an empirical evaluation. The approach yielded good results at identifying NFPs in DSPLs.

Keywords: Dynamic Software Product Lines · Non-Functional Properties · Feature models

1 Introduction

DSPL engineering is an approach that extends the concept of conventional SPL to allow the generation of software variants at runtime [1]. 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 [2].

One of the most important assets of a DSPL is the feature model. This artifact captures the similarities and variabilities between the possible configurations of products of a DSPL in a given context. In a DSPL feature model the system characteristics can be added, removed and modified at runtime

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

G. Botterweck and C. Werner (Eds.): ICSR 2017, LNCS 10221, pp. 65–73, 2017.
DOI: 10.1007/978-3-319-56856-0_5

in a managed way [1]. However, the feature model does not capture NFP explicitly neither influence these properties to achieve alternative configurations of a product variant.

Representing NFP in DSPL might be a rather complex activity. Given a particular context, configuration rules, features constraints and preferences of stakeholders, particularly the Functional Requirements (FR) and NFP [8], must all be considered. Although the NFP represent an important artifact associated to the quality of an SPL, modeling NFP techniques are not completely suitable for SPL engineering, as the approaches are commonly based on FR rather than NFP, as indicated in an extensive literature survey on the field [4].

In this paper, we present an approach to modeling NFPs in DSPL feature models. The approach is capable to identify, in a systematic fashion, the set of NFPs, constraints and context adaptation scenarios. To evaluate the approach we carried out an empirical evaluation, aimed to identify the relevant NFPs and context adaptation scenarios, with their respective constraints, to support the features modeling and the representation of NFPs in DSPL feature models. As results we obtained 70% accuracy in relation to the identification of NFPs, and 90% accuracy to identify context adaptations scenarios. Besides, we could observe that the approach could be useful to provide software engineers with an adequate support at identifying NFPs and their relation to the behavior of the feature models through interdependence constraints.

2 ReMINDER: An AppRoach to Modeling Non-FunctIoNal Properties in DSPL

The ReMINDER approach aims to provide a systematic way for identifying NFPs and context adaptation scenarios, with their respective constraints, as a means to support feature modeling and the representation of NFP in DSPL feature models. The proposed approach could be considered an endeavor to bridge an important gap, concerning to the lack of existing approaches to enable the representation of both NFP and context scenarios in feature models.

Figure 1 provides an overview of the ReMINDER approach. The ReMINDER approach encompasses two major phases: (1) identification and representation of NFPs in feature models; and (2) identification of constraints and context adaptations scenarios. The first phase aims to identify the NFPs that are critical for a DSPL and represent them in an quality feature model. The input of this phase are SPL or DSPL feature models without context, modeled with the FODA notation [3]. The output of this phase is a quality feature model. The second phase aims to identify the constraints between NFPs and context adaptations with the guidance of a Domain Engineer. The input of this phase is the SPL or DSPL feature models without context and the quality feature model built in the previous phase, while as output DSPL feature models with context, the quality feature model and context adaptation scenarios. The NFPs catalog, the quality feature model and the templates used by ReMINDER are available online[1]. Each step of the ReMINDER approach is discussed along the next sections.

[1] http://bit.ly/2lx3Ik2.

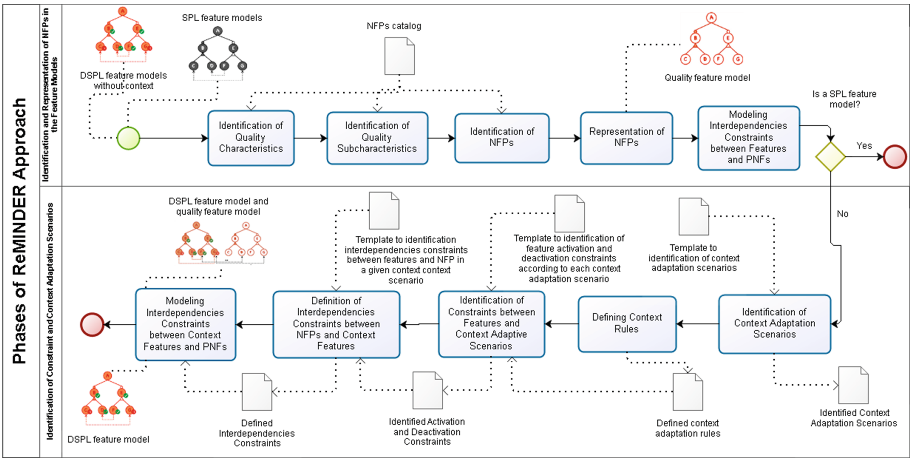

Fig. 1. Process of ReMINDER approach.

2.1 Phase I - Identification and Representation of NFPs in Feature Models

The first phase of the ReMINDER approach aims to identify the NFPs that are critical for a DSPL and represent them in a quality feature model. The quality feature model classifies these NFPs according to the quality characteristic and sub-characteristics presented in the product quality model of the ISO/IEC 25010 SQuaRE [6]. To assist in the identification of these NFPs, we created a NFPs catalog from [9]. This catalog also classifies the NFPs according to the SQuaRE quality characteristic and sub-characteristics. This phase encompasses four steps, as follows:

Step I - Identification of Quality Characteristics. It aims to identify the relevant quality characteristics based on the stakeholders needs.
Step II - Identification of Quality Sub-characteristics. Next, for each quality characteristic previously identified, it aims to identify the relevant quality sub-characteristics.
Step III - Identification of NFPs. This step consists of adding the NFPs established by the stakeholders, for each relevant sub-characteristic previously identified. The relevance of a NFP for a specific sub-characteristic is represented by qualifiers - high, medium or low. For example, for the quality sub-characteristics operability, the NFP effort can be measured as low.
Step IV - Representation of NFPs. The NFPs could be represented through a quality feature model, composed of quality characteristics, sub-characteristics, and NFPs. This artifact together with the DSPL feature model is given as input to the next phase.

2.2 Phase II - Identification of Constraint and Context Adaptation Scenarios

The second phase of the ReMINDER aims to identify the activation and deactivation constraints between features and adaptation context scenarios, based on the definition of context rules. In addition, define how an activated and deactivated context feature might influence a NFP in a particular context adaptation scenario. This phase has as input the quality feature model together with the DSPL feature model. For this phase we identified four steps, as follows:

Step I - Identification of Context Adaptation Scenarios. In this approach we classify the context adaptations in three categories: computing contexts, user contexts, and physical contexts. According to [5], *computing context* refers to the hardware configuration used, such as the processors available for a task, the devices accessible for user input and display, and the bandwidth. *User context* represents all the human factors, such as the user's profile, calendars, and profiles. *Physical context* refers to the non-computing-related information provided by a real-world environment, such as location, lighting, etc. The definitions of each context adaptation scenario describe situations relevant to the domain of an DSPL, based on a scenario ID and the following properties:

- **Contexts -** used to inform the type of context, computing context, user context, and physical context.
- **Context informations -** used to identify the information that varies according to each type of context, for example, battery level is a computational context information.
- **Qualifiers -** used to identify the qualifications of each context information. A qualification can be either boolean or classified in a qualitative scale (high, medium and low). For example, the context information "free memory level" can assume the values high or low. On the other hand, context information "internet connection" can assume the values true or false.
- **Quantifiers -** used to describe the values associated with each type of qualifier. 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. For example, high free memory level >= 128 or low free memory level < 128.
- **Status of quantifiers by scenario -** Used to indicate which of the values of each context information is valid in the scenario definition.

Step II - Defining Context Rules. After the identification of context adaptation scenarios, we can define the context rules that specify how a context information impacts on the configuration of the products of a DSPL. It indicates, for example, which features should be activated and deactivated.

These rules have as properties: an identifier and an expression. The expression is formed by an antecedent, the operator *implies*, and a consequent. The antecedent is an expression that can contain context information with its variation activated and deactivated. The operator *implies* determine that if the antecedent is true in the adaptation scenario, then the consequent should be or not included in the configuration of the feature models. The consequent is an expression that can contain features and logic operators. The expression determines which features need to be activated or deactivated according to each context adaptation scenario. For example, RC1 - Low Memory Level *implies* NOT (Persistence); RC2 - NOT (Internet Connection) *implies* Via Sensor.

Step III - Identification of Constraints Between Features and Context Adaptive Scenarios. The next step is to identify the constraints of activation and deactivation features according to the context rules that will be executed based on the definitions of context adaptation scenarios and the features present in the initial configuration of the product. As exemplified previously, for a given context adaptation scenario, in which just the antecedent Low Memory Level is true, RC1 must be executed, i.e., Persistence must be deactivated.

Step IV - Definition of Interdependence Constraints Between NFPs and Context Features. After identifying the constraints between features and context adaptation scenarios, we need to identify the relations of interdependencies between the identified NFPs and the features, according to each context adaptation scenario. To specify and measure this interdependence constraints, we have added a concept of goal-oriented modeling, in particular the concept of contribution links [10]. In this way, we assign interdependence constraints between context features and NFPs, in a given context. These features can be represented visually (see footnote 1) and 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.
- **"–" -** the feature has a negative influence on an NFP if it is activated.

3 Empirical Evaluation

An observational study was executed to analyze the process defined in the ReMINDER approach. The DSPL used in this study is called DSPL MobiLine [7]. The MobiLine's feature model, used in the observational study is available online (see footnote 1). Two M.Sc. students and one PhD participated in this observation study.

3.1 Execution

For this empirical study, a questionnaire was applied to characterize each participant. Then, a training was held for participants on the steps used in the

ReMINDER approach. An observation activity occurred in a single session. Initially, participants had to assume the role of domain engineer in charge of specifying the relevant NFPs for the DSPL mobile visit guide. The effort expected to accomplish this task was medium. We consider a scale between high, medium and low. The participants identified an average, respectively, 50%, 100% and 62% of the expected NFPs. These results show that it is possible to identify and specify NFPs using ReMINDER.

Next, the engineers should identify which features should be activated and deactivated based on the context rules that are performed and the definitions of the context adaptation scenarios. The effort expected to accomplish this task was low. A set of 11 contextual rules and two contextual adaptation scenarios were defined. In the first scenario, 11 activated features and 2 deactivated were provided, whereas in the second scenario, 9 activated and 4 deactivated features were provided. Only one participant did not identify all features.

Finally, the participants had to identify the interdependence constraints between the NFP that were identified in the first part of the observation study and activated and deactivated features according to context adaptations scenarios identified in the second part. The effort expected to accomplish this task was medium or high. In this activity, we analyzed whether the constraints identified by each engineers were consistent. Only one of the engineers presented inconsistency in their constraints.

3.2 Results and Findings

After the observation activities, we performed a interview with the participants to contribute with the evaluation approach and answer the research questions.

Q1: What is the effort expended to identify the NFPs according to the quality characteristics and sub-characteristics of the SQuaRE standard, through the NFPs Catalog?

In general, the effort spent to identify the NFPs according to the quality characteristics and sub-characteristics was medium. However, prior knowledge about the quality characteristics and sub-characteristics of SQuaRE standard is required. Regard the number of correct answers in the NFP identification task, we verified 70% accuracy in relation to the expected result.

Q2: What is the effort expended to specify the interdependence constraints between identified NFP and features?

When it is necessary to specify the interdependence constraints between NFP and features, the overall effort spent was medium. The participants took on average about 29 min to specify the constraints, even with the support of the proposed template. Initially, the participants took about 20 min to specify the interdependence constraints in the first context adaptation scenario. In the second adaptation scenario, they took less time to accomplish the task, which indicates a reduced effort necessary.

Q3: What is the accuracy and effort expended to specify the adaptation scenarios and their constraints?

The use of the templates facilitated the specification of context scenarios adaptations and their constraints. According to the participants, with the templates it was possible to delimit the values of the context information, and manage their features and constraints. In addition, the ease of specifying context adaptation scenario is one of the greatest benefits of the approach. However, during the observation activity, the engineers made a high effort to identify which context rules should be executed. In relation to the number of correctness in specifying scenarios of context adaptations and their constraints, we verified 90% accuracy in relation to the expected result for the first scenario of context adaptation, while 89% accuracy regarding the expected result for the second context adaptation scenario.

Q4: What are the drawbacks and benefits of the approach?

The participants mentioned the approach was both intuitive and easy to understand, the phases are well divided and with specific objectives. The approach facilitated the identification of NFP and their relationship to the features of the product line. In addition, the templates and the NFP catalog were a useful support at identifying NFP in DSPL. However, they reported a greater effort to accomplish the task to specify the interdependence constraints between identified NFPs and context features.

3.3 Threats to Validity

We next discuss the threats to the validity of our empirical assesssment. **Construct Validity.** As the main researcher of this study is part of the same experimenter's research group, he had a strong influence on the conclusions. To mitigate this threat, other participants played the role of a domain engineer, during the process of using the approach. **Internal Validity.** The approach has a couple of phases and steps. It is possible that some concepts have been misinterpreted. To mitigate this threat, the researcher was, during an observation activity, all the time close to the participants. The Mobiline DSPL used in this study may not be the most appropriate. In order to mitigate this threat, we aim to continue to investigate the approach in other application scenarios. **External validity.** As the empirical study was executed in one small academic DSPL, it is difficult to generalize the findings. Hence, the discussions are limited for this DSPL context. Despite the limitations, researchers can extend the study by replicating it in different DSPL contexts following the design of this study.

4 Related Work

Zang et al. [11] presents an approach to modeling quality attribute and quality-aware product configuration in SPLs. Similar to the objective of this work, the authors proposed an approach for modeling quality attributes in the feature models leading to a quality-aware product configuration. However, in this work, we defined an approach for identification and representation of NFPs and identification of constraints and context adaptation scenarios to the DSPL feature models. Soares et al. [9] identified in a systematic review, a set of 52 NFP that may emerge at runtime. This set of 52 NFPs that emerge at runtime, by application domains that suffer adaptations at runtime, can also emerge in autonomic systems and context-aware systems. From this classification we have created a NFP catalog as one of the main artifacts of our approach. Sanchez et al. [8] presents an approach for self-adaptive systems, based on the process of specification, measurement and optimization of quality attribute properties in feature models. In addition, instead of extending the feature models in two submodules, we propose a quality feature model to support the NFP representation.

5 Conclusion and Future Work

This paper presented the ReMINDER approach that support the modeling of NFP in DSPL. We carried out an empirical evaluation to evaluate the applicability of the approach. The evaluation consisted of observation and interview activities with domain engineers. Among the observations, we examined the role of the approach as a guide to identification and specification of NFP. From an NFP catalog, stakeholders can find NFP related to the quality characteristics and sub-characteristics of a DSPL, according to the SQUaRE standard. As future work, we plan to extend the ReMINDER approach to formally verify whether the NFP in DSPL feature models match the expected results of the process of reconfiguration at runtime. In addition, we plan to add new NFPs to the NFPs catalog, which may emerge at runtime and develop a tool to assist in operationalizing the approach.

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. Capilla, R., Bosch, J., Kang, K.C.: Systems and Software Variability Management. Springer, Heidelberg (2013)
3. Czarnecki, K., Helsen, S., Eisenecker, U.: Formalizing cardinality-based feature models and their specialization. Softw. Process: Improv. Pract. **10**(1), 7–29 (2005)
4. 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)
5. Hong, D., Chiu, D.K., Shen, V.Y.: Requirements elicitation for the design of context-aware applications in a ubiquitous environment. In: 7th International Conference on Electronic Commerce, pp. 590–596. ACM, Xi'an (2005)
6. 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)

7. Marinho, F.G., Andrade, R.M., Werner, C., Viana, W., Maia, M.E., Rocha, L.S., Teixeira, E., Ferreira Filho, J.B., Dantas, V.L., Lima, F., et al.: MobiLine: a nested software product line for the domain of mobile and context-aware applications. Sci. Comput. Program. **78**(12), 2381–2398 (2013)
8. 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)
9. 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)
10. Van Lamsweerde, A.: Goal-oriented requirements engineering: a guided tour. In: Proceedings of the 2001 Fifth IEEE International Symposium on Requirements Engineering, pp. 249–262. IEEE (2001)
11. Zhang, G., Ye, H., Lin, Y.: Quality attribute modeling and quality aware product configuration in software product lines. Softw. Qual. J. **22**(3), 365–401 (2014)