Search Based Design of Software Product Lines Architectures

Thelma Elita Colanzi
Computer Science Department
Federal University of Paraná, UFPR, Brazil
thelma@inf.ufpr.br

Abstract—The Product-Line Architecture (PLA) is the main artifact of a Software Product Line (SPL). However, obtaining a modular, extensible and reusable PLA is a people-intensive and non-trivial task, related to different and possible conflicting factors. Hence, the PLA design is a hard problem and to find the best architecture can be formulated as an optimization problem with many factors. Similar Software Engineering problems have been efficiently solved by search-based algorithms in the field known as Search-based Software Engineering. The existing approaches used to optimize software architecture are not suitable since they do not encompass specific characteristics of SPL. To easy the SPL development and to automate the PLA design this work introduces a multi-objective optimization approach to the PLA design. The approach is now being implemented by using evolutionary algorithms. Empirical studies will be performed to validate the neighborhood operators, SPL measures and search algorithms chosen. Finally, we intend to compare the results of the proposed approach with PLAs designed by human architects.

Keywords—software product lines; multi-objective algorithms; software architecture optimization

I. INTRODUCTION

Software Product Line (SPL) is a systematic approach for software reuse applied to a family of specific products within a well-defined domain. Organizations have adopted SPL approach in order to shift from the reuse of individual components to the large-scale reuse of a product-line architecture (PLA) [1]. A PLA entails a design that is common to all the products derived from the product line. The design of a PLA should encompass the components realizing all the mandatory and varying features in a domain [2].

A key condition for the successful design of a PLA is the identification of architectural elements that modularize each individual SPL feature. Otherwise, the PLA is likely to be the target of early reviews and major refactorings since the design outset, thereby delaying the development process. A non-modular PLA will also suffer more changes when new products need to be accommodated, making it difficult to sustain design stability over time. Design of modular PLAs is more difficult to achieve in proactive SPL engineering, as developers often cannot depart of existing architectures [1].

Software architecting is a people-intensive, non-trivial and demanding task for software engineers to perform, and a fundamental activity of software development [3] since it involves several questions such as the best dependency balance among components, modularity maximization, and fulfillment of quality requirements. PLA design and its evolution is even more complex than software architecting because there are many inter-related variant points. In many cases, some resources such as design patterns and the use of aspect-orientation to modularize crosscutting concerns can turn the PLA more stable [4].

Solving such problem is in general a complex task for that there is no optimal and exact solution. The solution of such hard problems has been attacked by the field, named Search Based Software Engineering (SBSE) [5], [6]. In SBSE, the software engineering problems are formulated as optimization problems to be solved by search based techniques, which are derived from the fields of operations research, such as linear programming, and metaheuristics, such as Evolutionary Algorithms, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and so on.

Approaches based on Multi-Objective Evolutionary Algorithms (MOEAs) have achieved promising results because they allow the use of different factors and measures that can affect the problem [7]–[10]. Many times these factors are in conflict and usually there is no a single solution for the problem. Optimization problems with two or more objective functions are called multi-objective. The idea is to find a good trade-off of solutions that better represent the possible compromise among the objectives. In the literature we found different works that use search-based algorithms, including multi-objective ones, to solve software engineering problems [5], [11]. The area of software architecture can also benefit of SBSE, including optimization concerns related to the modularization, patterns, and aspect-orientation [6].

Therefore, the hypothesis of this research work is that the multi-objective optimization can achieve good PLAs and make its design less dependent of human architects. Furthermore, a systematic and automated approach can intensively reduce the propagation of design flaws during the SPL development process. With these goals in mind, the research described in this paper introduces a PLA design approach based on multi-objective search algorithms to optimize product-line architectures. The approach produces a set of good solutions with the best trade-off among different and conflicting objectives, such as the SPL extensibility and complexity. The main contributions of the work are to reduce SPL development efforts and costs.
II. RELATED WORKS

There are some works related to product line in general. In [12] the objectives are to maximize the market share of a company’s products, and minimize the total product development cost of a product line and the total product development cycle time. In another work [13], the objective is to maximize the number of different products that can be generated. Despite of these works are on optimization, they do not address SPL and works that focus optimization of software architectures are more related to ours. Among them, we can mention the search based ones. They address object-oriented software design, service-oriented software design, software modularization and software refactoring [11]. Grunske [14] introduce a multi-objective strategy based on architecture refactoring that performs architectural transformations and identifies alternatives considering functional and quality requirements. Two objectives are adopted: to reduce development cost and to improve the reliability on the system.

Bowman et al. [9] present a multi-objective and evolutionary approach to optimize class responsibilities assignment, considering coupling and cohesion measures. The approach works with the class diagram and the SPEA2 algorithm. Simons [3] propose a search-based approach to find the best object-oriented conceptual software design. This approach encompasses a novel object-based representation and genetic operators for evolutionary search. The genetic operators change methods and attributes from one class to another and also add new classes. Simons and Parmee [15] have also investigated the use of single and multi-objective genetic operators in this context. By using cohesion and coupling measures as objective fitness functions and genetic algorithms and evolutionary programming, they conclude that the multi-objective approach has significant potential to find good solutions in the context of their work. The MOEA used was NSGA-II. In his evolutionary approach, crossover operator was not used since it has found negative results [3].

Räähä [16] introduces an approach to synthesize software architecture by using genetic algorithms. The technique applies architectural design patterns for mutations and quality metrics for evaluating individual architectures, producing a software architecture on the basis of functional requirements given as a graph of functional responsibilities. Two quality attributes are considered, modifiability and efficiency. They stated that a mutation can be performed as either the application or removal of an architectural pattern. In this approach the fitness function is expressed in terms of quality metrics (coupling and cohesion), and crossover operation is performed by merging two architectures without breaking existing pattern instances. In an empirical study, Räähä et al. [17] observed that the crossover operator does not offer significant advantage in the software architecture optimization when the crossover operator was implemented in a random fashion. And, in other study [16], the results demonstrated that multi-objective algorithms can produce a sensible set of architectures in the efficiency-modifiability space.

The mentioned works successfully optimize software architecture, however, they do not consider specific characteristics of SPLs, such as variabilities, variation points, features assigned to architectural elements, and so on. In this sense, it is necessary to define a novel optimization approach specific to SPL context and needs, which uses different and more appropriated quality measures ([18]–[21]).

III. PROPOSED APPROACH

Several SPL development activities can be benefit by search techniques. Our focus is PLA definition and evaluation, since PLA is the most important artifact in the SPL approach. Both activities can be optimized to fulfill, in addition to basic design principles like coupling and cohesion, other quality requirements such as modularization of the features, extensibility, safety, reliability, complexity, etc. Therefore, multi-objective search algorithms seem to be the best choice to solve the PLA optimization problem since they have shown better results than mono-objective ones when there are different objectives to be achieved [22].

Thus the proposed approach to PLA optimization is multi-objective and it aims at finding PLA designs to: (a) improve the products share, and then the PLA must be as more extensible as possible; (b) reduce maintenance and evolution costs and (c) obtain high reusability. These two last goals can be achieved by means of a modular, extensible and little complex PLA. Figure 1 depicts the multi-objective optimization approach to PLA. Such approach encompasses four activities, explained below, and different measures can be used in the optimization process.

Construction of the PLA Representation: The inputs for this activity are the PLA, the features model and the list of concerns assigned to the architectural elements. A PLA representation is generated, containing all architectural elements, their inter-relationships, all variation points, variants and concerns assigned to each architectural element.

Definition of the Evaluation Model: There are different measures that allows evaluating the PLA, and in this activity the architect selects the measures to be used in the optimization. Conventional measures can be used, such as coupling, cohesion, size, etc. However, specific measures for SPL are more suitable. The extensibility measure [20] allows to obtain PLAs more extensible and, then, it is possible to have a greater number of SPL products maximizing the market share. In addition to this, there are cohesion measures specific for SPL, such as Internal-ratio Feature Dependency and External-ratio Feature Dependency [19]; Provide Service Utilization (PSU) and Required Service Utilization (RSU) [21], that allows to measure the component cohesion; and Lack of Concern Cohesion (LCC) [18]. These measures give modularity indicators and modular PLAs are
more stable, reusable and maintainable. Furthermore, the concern-driven measures [18] can be used to analyse the modularization of crosscutting concerns [4]. Finally, considering that a feature is a concern in a SPL, the concern-driven measures [18] help the architect to analyse the feature-based cohesion (LCC), the concern diffusion (CDAC, CDAI and CDAO) and the interlacing between concerns (CIBC, IIBC and OOBC) in a PLA.

**Multi-Objective Optimization:** The PLA representation obtained in the first activity is optimized according to the constraints given by the architect. Each obtained alternative PLA is evaluated following the evaluation model which contains the measures defined in the previous activity. A set of PLA representations is generated as output and they consist on the alternatives that have the best trade-off among the objectives. Possible search based algorithms that can be used in this activity are: MOEAs, ACO, PSO, etc.

**Transformation and Selection:** The set of PLA representations is converted in a legible view to the architect, who selects one alternative that prioritizes some objective(s), to be adopted as the PLA according to the organization priorities.

Independently of the used search technique, it is important to have a suitable representation to the PLA. Most techniques adopt representations that are not natural and not able to represent variabilities and features, such as the integer array used in [9] and the supergene representation proposed in [16]. Furthermore, by using such representations may turn the neighborhood operators complex.

In this sense, the object-based representation [3] seems to be more suitable and natural than the others, although, it is also not suitable to SPL characteristics. So, we proposed a new representation that comprises all the needs regarding to SPL in order to allow the PLA optimization. This representation consists on a metamodel, presented in Figure 2. A PLA contains architectural elements (components, interfaces, operations and their inter-relationships) besides to be conform to some architectural style or design pattern. Each architectural element is assigned to feature(s) and it can be common to all SPL products or variable, being present only in some product(s). And, variable elements are associated to variation points and their variants.

**IV. METHODOLOGY AND PROGRESS**

In order to address the research question, some empirical studies will be conducted, involving a combination of empirical research methods, such as exploratory case studies and controlled experiments.

In our research, we have decided to focus the instantiation of the optimization approach by using MOEAs since they have achieved good results to solve software engineering problems. In this sense, crossover operator and some mutation operators have already been proposed. One of this mutation operators include the aspect-oriented architectural style. An empirical study will be performed to evaluate the feasibility of the proposed operators, and, if it is necessary, others will be defined. Furthermore, we plan to conducted another study in order to analyse which MOEA is more appropriated to solve the problem and which kind of evolutionary operators are suitable in this case.

From the set of measures specific for SPL and the conventional measures found in the literature, we have already chosen a subset to include in the instantiation of the optimization approach: concern-driven, PSU and RSU, extensibility and complexity measures. They were chosen because fulfill the goals of the proposed optimization approach. However, we intend to perform an empirical study to confirm such choice as soon as the proposed approach be implemented.
We intend to implement the proposed approach in Java. Having an operational optimization approach allows: (i) to conduct the empirical studies mentioned above and (ii) to reduce costs and effort in the SPL development process, what is interesting to IT organizations that have adopted SPL approach.

V. EXPECTED CONTRIBUTIONS AND EVALUATION

Our research work is expected to result in the following contributions: (a) a novel and original approach to optimize PLAs, (b) empirical evidence about the set of metrics that can be used together to obtain modular, extensible and reusable PLAs, (c) new empirical evidence about the benefit of the crossover operator in the evolutionary process of PLA, (d) an automated optimization approach that can be used to avoid design flaw propagation in the PLA design and to reduce the development efforts.

Contributions b and c will be evaluated in the planned case studies mentioned in Section IV. In order to validate the proposed approach (contribution a), we intend to perform case studies to optimize PLA and then, compare the achieved optimized PLAs with PLA designed by human architects. There are at least three SPL that can be used to do this: Mobile Media [4], AGM [2] and SPL-BET [23].

ACKNOWLEDGMENT

I would like to thank CNPq, for financial support, and my supervisor Silvia Regina Vergilio.

REFERENCES


