Towards a Search-based Interactive Configuration of Cyber Physical System Product Lines[1]

Kunming Nie[1], Tao Yue[2], Shaukat Ali[2]

[1] Software Engineering Institute, Beihang University, Beijing, China
niekunming@cse.buaa.edu.cn
[2] Certus Software V&V Center, Simula Research Laboratory, Oslo Norway
{tao,shaukat}@simula.no

Abstract. Product Line Engineering (PLE) is a technique to improve the quality and productivity of developing (via configuration) Cyber Physical Systems (CPSs). A CPS often contains many heterogeneous components with complex constraints relevant to product configuration in the context of PLE. Manual configuration is error-prone and has low productivity since managing and manipulating such constraints in a real industrial context is very complicated and thus warrants an automated solution. However, fully automated solution is often impossible for CPSs since some decisions must be made manually by users, thus requiring an interactive configuration solution. Therefore, we propose a semi-automated and interactive configuration solution for CPSs. We started our research by analyzing the characteristics of three industrial CPS product lines and constraints required for supporting such a configuration solution. Then we conducted some pilot experiments on applying search algorithms to find optimal decision orders for configuring a product. In this poster, we describe the whole research idea and, discsusion on the work we have completed, the initial results, and the future plan.

1 Introduction

Modern society is increasingly dependent on Cyber-Physical Systems (CPSs), which rely on software to control many individual systems and complicated coordination of those systems [1][2]. Such systems include communications and control systems, avionics, oil and gas production platforms, with the characteristics being large-scale, heterogeneous and distributed. Product Line Engineering (PLE) had been used in different CPSs, including communication systems, intelligent traffic systems, industrial automation systems, aerospace, and distributed weather station network [3-6], to enhance the reusability [7]. Based on our experience of working on three commercial CPSs product lines (i.e., Subsea Production Systems (SPSs), Video Conferencing Systems (VCSs) and Vessel Prognostics and Health Management Systems (VPHMS)), we summarize the characteristics of CPS PLE: *Dynamic Configuration (also called Runtime Configuration)* [3], *Temporal Variability* [4], *Feature Interaction* and *Subsystem/Component Interaction* [5]. These characteristics bring new challenges to the variability modeling and product configuration of CPSs.

In the context of CPS PLE, there are a large number of variation points that have to be configured correctly by conforming to a large number of constraints. Product configuration is therefore an

[1] We thank the Research Council of Norway under the Certus SFI project, the Ministry of Industry and Trade of Norway, and the National Natural Science Foundation of China (No. 61170087, 61370058) for funding the research.

error-prone and time consuming activity if it is not automated [8] [9]. For one system that has n variation points, each of which has m variants, it is then theoretically possible to derive m^n products without considering constraints among variation points and variants. To derive a correct product, constraints should therefore be satisfied during configuration [10]. Therefore, there is a need for an interactive and semi-automated configuration solution for CPS PLE configuration. Note that it is often impossible to automate all configuration steps in the context of CPS PLE, hence user interaction is required in the configuration process.

We provide an overview of such an approach is Fig. 1. There are mainly two steps involved: variability modeling and product configuration. In the domain engineering phase, a variability modeling approach should be applied to capture the commonalities and variabilities of products in a product line. In the domain engineering phase of CPS PLE, product line architecture (PLA) modeling and constraint specification approaches are required to capture commonalities and variabilities on the system architecture and design and constraints relevant to configure a correct product. For different purposes or in different contexts, different modeling and specification approaches can be used. For example, an extension of UML for PLA in the form of a profile together with an existing profile e.g., MARTE can serve the purpose of PLA modeling. When combined with UML, Object Constraint Language (OCL) is often used to specify constraints. In the application engineering phase, a configuration tool with the functionalities of decision inferring, decision ordering and consistency checking is expected to be applied to assistant configuration engineers to configure a product.

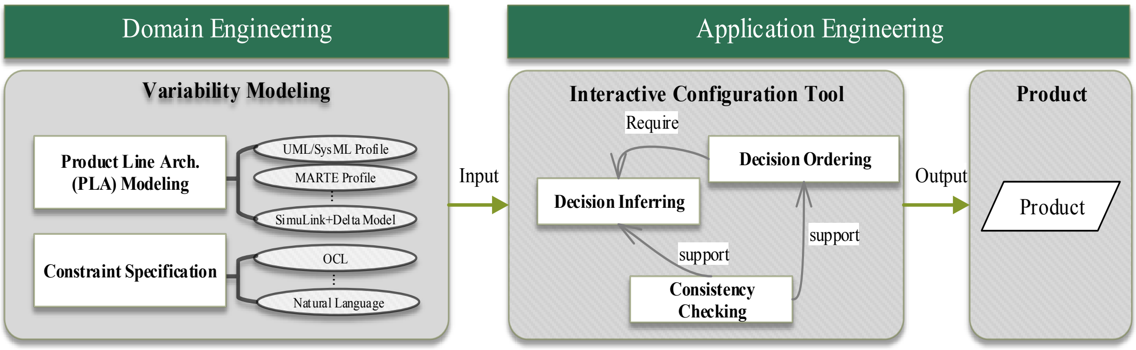

Fig. 1. Overview

Regarding our solution, SimPL [11] will be used as the PLA modeling methodology, which was designed for describing the commonalities and variabilities of CPSs and has been applied to model the PLA of subsea production systems. As SimPL is a UML profile, constraints will be specified using OCL. Note that we have proposed a classification of constraints in CPS PLE, based on our experience of working with three industrial partners, for supporting semi-automated and interactive configuration with five functionalities: decision ordering, decision inferring, reverting decision, consistency checking and collaborative configuration [6]. Among these five functionalities, `Decision Ordering`, `Decision Inferring` and `Consistency Checking` are dependent on each other. Therefore, in our research, we will mainly focus on these three functionalities. Our ultimate objective is to find an efficient solution to guide configuration engineers through the complicated and error-prone process such that eventually the overall quality and productivity of the product development can be improved. As the first step towards this goal, we devised a search-based solution for addressing the decision ordering functionality in an efficient and scalable way, which will be discussed in detail in the next section.

2 Search-based Configuration Solution

Although there are many interactive configuration solutions for the PLE of CPSs [6], most of them just consider relatively simple dependencies among variabilities. However, CPSs product lines generally have hundreds and thousands of variation points and constraints. The interactive configuration solution should find an optimized decision ordering that satisfies the constraints. To efficiently address such an optimization problem, we recommend search-based solutions, which obtained very positive results when addressing other software engineering optimization problems such as model-based testing [12].

Our solution aims to guide configuration engineers through the configuration process but not fully automates all configuration steps, which are often impossible in CPS PLE, as we discussed previously. Therefore, the configuration tool recommends an order to make decisions. This order is mixed of manual configuration steps and automated configuration steps. Automation refers to another functionality of the tool, i.e., Decision Inferring. This functionality infers decisions based on dependencies of variation points, various types of constraints, and decisions made. When a configuration engineer makes a manual configuration on a variation point, the tool will automatically check the correctness of the configuration and identify possibilities to automatically infer decisions. This cycle is repeated until all variation points are configured. Note that either a manual or automated configuration step will trigger the tool to find an optimal order as the optimization space is changed. As shown in Fig. 2, our solution contains three main steps: variability modeling, transformation to intermediate representation, and search-based decision ordering optimization, which will be discussed in details in the following sub-sections.

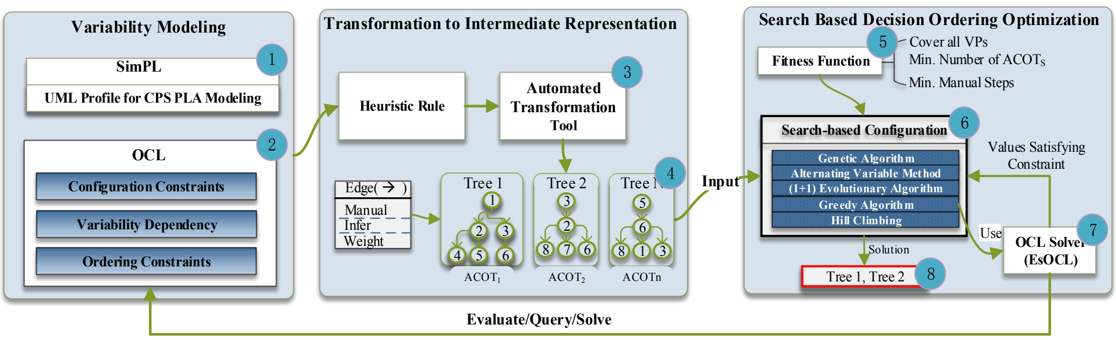

Fig. 2. Search-based Interactive Configuration Solution

2.1 PLA Modeling and Constraints Specification

In our previous work [6], we reported that CPSs are heterogeneous systems that typically combine mechanical, electrical, and software components and are large-scale both with respect to the diversity of the types of their hardware and software components. Most of the current variability modeling method does not take the characteristics of CPSs into consideration. SimPL [11] (box(1) of Fig. 2), is however a PLA modeling methodology with a UML profile for CPS product lines. In SimPL, UML and its extensions are used to create both the base and the variability models. In particular, UML constructs such as classes and relationships are used to model software, four stereotypes from MARTE are used together with UML constructs to model hardware, and UML templates and packages together with three stereotypes from a newly introduced profile are used to model variabilities.

A semi-automated and interactive configuration solution heavily relies on a large number of constraints that should be formally specified to facilitate the five functionalities of automatic configuration (Section 1). Therefore a classification of such constraints according to how they are specified, manipulated and enforced, and relate them in a systematic manner to the functionalities (e.g., decision ordering) of such an automated solution is required. In our previous work [6], we presented such a comprehensive classification of constraints. In our solution, these constraints will be specified using OCL (box(2) of Fig. 2). We are planning to use EsOCL [12] to solve OCL constraints, whereas DresdenOCL [13] to query and evaluate constraints. We selected these tools based on our previous experience of working with them.

2.2 Transformation to Intermediate Representation

To have an independent solution for decision ordering optimization, we propose to transform PLA models and constraints into an intermediate representation as shown in the middle part of Fig. 2. This provides the flexibility to use different PLA modeling and constraint specification methods other than UML, UML profiles and OCL. When a different approach is used, one only needs to write a transformation and the optimization part remains untouched.

We refer to the intermediate representation of PLA models and constraints as configuration model, which captures all the information of variation points and constraints. Configuration model can be used as the input for the configuration tool to support, for example, decision ordering.

Hence, in our solution to support decision ordering, a configuration model is a set of trees, which forms a *Forest* (box(4) of Fig. 2). In the forest, one tree corresponds to a constraint constraining the configuration sequence of a set of variation points. A node of a tree represents a variation point and an edge of the tree describes which variation point should be configured before or after which other variation point. In order to support decision ordering, a configuration model should specify additional information for the edges of the tree. Each edge has a set of attributes: *Manual* and *Infer* of type *Boolean*, respectively indicating whether a configuration step is manual or can be automatically inferred. Each edge of a tree is also characterized with another attribute *Weight* with different categories, each of which respectively indicates the strength of dependency of two variation points.

In order to transform PLA models and constraints to the configuration model, a set of transformations will be implemented. In our solution, we need to transform UML class diagrams with the SimPL profile applied and OCL constraints into the configuration model.

2.3 Decision Ordering Optimization

In order to find an optimal ordering of configuration steps for the configuration engineers, we propose a search-based solution. After the optimization, an optimal solution (*Tree 1* and *Tree 2*, in box(8) of Fig. 2) will be provided to configuration engineers. To configure a correct product, all constraints should be satisfied for a solution. Any constraint requiring solving is solved using EsOCL [12] shown in box (7) of Fig. 2, since we use OCL for constraint specification.

The optimization objective and corresponding fitness function used by search algorithms is provided in [14]. The optimization objective contains three parts: 1) minimizing overall manual configuration steps, 2) configuring most constraining decisions first and 3) satisfying ordering dependencies among variabilities. We evaluated five search algorithms including both local search algorithms (`Alternating Variable Method, Genetic Algorithm, Hill Climbing`) and global search algorithms (`(1+1) Evolutionary Algorithm, Greedy Algorithm`) (box (6) of Fig. 2).

We carefully designed and conducted a series of experiments by using the above mentioned algorithms combined with the fitness function on 65 problems. The 65 problems were generated forests with varied sizes with randomly populated attribute values. Results of the experiments were carefully analyzed using statistical analysis techniques, based on which, we concluded that Greedy Algorithm significantly outperforms the rest of the algorithms. In the future, we plan to run the experiments on an industrial case study.

References

1. Cyber-Physical Systems, http://cyberphysicalsystems.org/ (Last accessed July 2013)
2. Lattmann, Z., Nagel, A., Levendovszky, T., Bapty, T., Neema, S., Karsai, G.: Component-based Modeling of Dynamic Systems using Heterogeneous Composition. In: 6th International Workshop on Multi-Paradigm Modeling (2012)
3. Ortiz, Ó., García, A.B., Capilla, R., Bosch, J., Hinchey, M.: Runtime variability for dynamic reconfiguration in wireless sensor network product lines. In: Proceedings of the 16th International Software Product Line Conference, vol. 2, pp. 143-150. ACM, New York (2012)
4. Haber, A., Rendel, H., Rumpe, B., Schaefer, I.: Evolving delta-oriented software product line architectures. Large-Scale Complex IT Systems. Development, Operation and Management, pp. 183-208. Springer (2012)
5. Juarez-Dominguez, A.L., Day, N.A., Joyce, J.J.: Modelling feature interactions in the automotive domain. In: Proceedings of the 2008 international workshop on Models in software engineering, pp. 45-50. ACM, New York (2008)
6. Nie K., Y.T., Ali S., Zhang L. and Fan Z.: Constraints: The Core of Supporting Automated Product Configuration of Cyber-Physical Systems. Proc. Accepted at ACM/IEEE 16th International Conferene on Model Driven Engineering Languages and Systems (MODELS), Corresponding Technical Report: http://simula.no/publications/Simula.simula.2020, 2013. ,
7. Tao Yue, L.B., Bran Selic: Experiences with Model-based Product Line Engineering for Developing a Family of Inte-grated Control Systems: an Industrial Case Study. Technical Report, (2012)
8. Rabiser, R., O'Leary, P., Richardson, I.: Key activities for product derivation in software product lines. Journal of Systems and Software, vol. 84, 285-300 (2011)
9. Rabiser, R., Grünbacher, P., Dhungana, D.: Requirements for product derivation support: Results from a systematic literature review and an expert survey. Information and Software Technology, vol. 52, 324-346 (2010)
10. Rabiser, R., Grünbacher, P., Lehofer, M.: A qualitative study on user guidance capabilities in product configuration tools. In: Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering, pp. 110-119. ACM, (2012)
11. Behjati, R., Yue, T., Briand, L., Selic, B.: SimPL: A Product-Line Modeling Methodology for Families of Integrated Control Systems. Information and Software Technology, vol. 55, 607-629 (2013)
12. Ali, S., Iqbal, M., Arcuri, A., Briand, L.: Generating Test Data from OCL Constraints with Search Techniques. IEEE Transactions on Software Engineering, 1-47 (2013)
13. Dresden OCL, http://www.dresden-ocl.org/index.php/DresdenOCL. (2012)
14. Yue T., A.S., Nie K.: Towards A Search-based Interactive Configuration of Cyber-Physical System Product Lines, Simula Research Laboratory. Technical Report 2013-50, (2013)