CP-Based Framework for Software Product Lines Engineering

Angela Villota

CP-Based Framework for Software Product Lines Engineering

Angela Villota-Gomez

Centre de Recherche en Informatique, Université Paris 1 Panthéon-Sorbonne
90, rue de Tolbiac
75013 Paris

angela-patricia.villota-gomez@malix.univ-paris1.fr

MOTS-CLÉS : *Lignes de produits, Ingénierie des lignes de produits, Programmation par contraintes, Solveurs de contraintes.*

KEYWORDS: *Product lines, Product line engineering, Constraint programming, Constraint solvers.*

ENCADREMENT. *Camille Salinesi (PR) et Raúl Mazo (MCF)*

1. Context

The Product Line Engineering (PLE) and Software Product Line Engineering communities employ the term constraint referring to relations between elements in a system, non-functional requirements, feature relations, etc. It is almost natural that some authors formalize these concepts as boolean satisfiability problems or constraint satisfaction problems. Moreover, in the past years, many different works have formalized variability concepts and product line preferences in the shape of logic formulas or constraint satisfaction problems to support PLE activities like analysis, testing, verification, etc. with the purpose of taking advantage of Constraint Programming (CP) approaches. However, there is no proposal integrating CP in all the development stages of a PLE in a consistent manner. In this research project, we aim to design and develop a generic framework specially designed for PLE. With this framework, we want to consolidate the constraints, domains, and solvers to propose a more suitable approach for supporting PLE.

2. State of the art

We combine two paradigms in this project: CP paradigm and PLE paradigm. On the one hand, the CP paradigm is a programming paradigm to solve combinatorial problems modeled as a Constraint Satisfaction Problem (CSP). CSPs are defined in terms of variables, domains and constraints. A CSP may have one or more solutions, and they are produced employing one of the many consistency algorithms or by searching them using a constraint satisfaction tool called solver (Rossi *et al.*, 2006). On the other hand, a PL or system family is a collection of similar products sharing common characteristics and satisfying the requirements of a particular mission or market segment. Products in a product line are assembled from a common set of core assets in a prescribed way (Pohl *et al.*, 2005). There are previous proposals exploiting the constraint programming paradigm in PLE by assisting activities such as: analysis, configuration, derivation, synthesis, testing, simulation, and verification of product lines in different contexts (Benavides *et al.*, 2005; Czarnecki *et al.*, 2005; Mazo *et al.*, 2015). Constraints are also used to capture product line variability. Most studies propose rules for transforming variability descriptions into constraint satisfaction problems. Nevertheless, works as (Salinesi *et al.*, 2011; Mazo *et al.*, 2011) consider constraint satisfaction problem as an expressive method to describe variability. Moreover, the work of (Salinesi *et al.*, 2011) proposes a constraints language to describe product lines from a high-level meta-model general description. Additionally, (Mazo *et al.*, 2011) proposes the use of abstract constraints to represent PL with a unique notation that encompass different constraint languages (e.g., over Booleans, Integers, Reals, trees, lists, etc.).

3. Problem

We have evidence of previous works that CP is a paradigm and a viable model for supporting a PLE process. However, there is no proposal integrating CP in all the development stages of a PL in a consistent manner. This may happen because the CP approaches are modeled or adapted for a particular objective or application. Instead, we want to design and develop a generic framework gathering constraints, domains, and constraint solvers specially designed for PL. Therefore, our framework will emerge as a suitable approach for supporting PLE consisting on: (i) A constraint language for representing PL models in a more accurate way, consolidating the constraints included in previous works, plus other not included yet and. This language will progress the proposal of (Mazo *et al.*, 2011) (ii) A solving mechanism supporting the proposed constraint language.

4. Actions taken

We apply the design science paradigm in our research project. This paradigm is one of the paradigms proposed to conduct research in information systems disciplines (Peffers *et al.*, 2007). According to above methodology, this project can be divided into four stages: solution design and development, demonstration, evaluation, and communication. Table 1 shows the objectives related to each stage.

Stage	Objectives
Solution design and development	Objective 1: Elicitate the types of constraints considered relevant in the product lines engineering. The elicitation must be carried out by studying the literature and also by surveying users in the academic and industrial context Objective 2: Propose and develop a Generic Constraint System for modeling variability as a constraint satisfaction problem. Objective 3: Propose and develop a solving mechanism supporting the proposed constraint system.
Demonstration and evaluation	Objective 4: Validate this proposal with several case studies and benchmarks.
Communication	Objective 5: Publish and disseminate this project in conferences and index journals.

Table 1. Project stages and objectives

As a first step, we conducted a systematic mapping study with 6 research questions over publications containing evidence of the application of the constraint programming paradigm in the developing of product lines. With this study, we aimed to provide an overview of the research on the intersection of PLE and CP subject. As results of our mapping study, we proposed a *Classification Framework* for CP-based approaches in the context of PLE, and built a comprehensive collection of constraints for PLE. The proposed classification framework is a four-dimensional framework composed by four views: expressivity, translation, application, and support. Consequently, the proposed classification framework allows the publications characterization from four different points of view. The second contribution of our mapping study emerges as a consequence of the classification of constraints used in the PLE. To perform the classification and answer the question: *what is expressed as a constraint in CP and how?* we gather the constraints employed in the different publications regarding their semantics and their implementation. Therefore, regarding the semantics, we obtained two collections of constraints: constraints to document variability, and constraints to express product line preferences. Additionally, constraints in PLE are implemented using arithmetic constraints, boolean constraints, global constraints, and reified constraints.

4.1. *A Solution Proposal.*

One of the findings in our mapping study is the usage of different approaches for solving constraints problems, and specialized solvers. Therefore, we plan to include the different paradigms and solvers taking advantage of their strengths. In consequence, the first proposal for a CP-based general framework specially designed for PLE considers the integration of paradigms for solving constraint problems such as Logic-Formula Satisfiability Problems (LSP), and CSP. The CP-based framework for PLE is a four-level framework, as shown in Figure 1. The first level, *constraints meta-model* contains a generic constraint language gathering the meta-model for constraints relevant in product lines engineering. The second level, *constraints instantiation* transforms a generic constraint into a constraint for a particular paradigm (LSP, CSP) with the help of a compiler, and federator components. The third level, *solver paradigm meta-model* contains a component for translating a constraint (represented in a paradigm) into constraints to be used in a particular solver paradigm (PBS, CLP, SAT, SMT, BDD). Finally, the fourth level, *solver implementation* transforms the constraints obtained in the third level into a program regarding the compelling a solver's implementation syntax. In consequence, users may profit the strengths of one particular paradigm and solver.

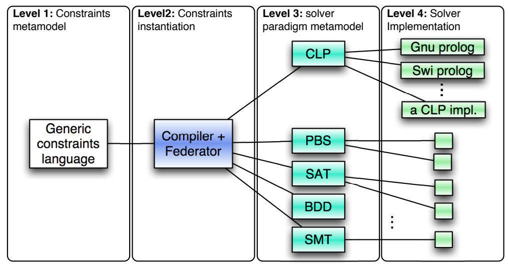

Figure 1. Framework

5. Future Work

In this paper, we presented the proposal and first results of the research project to design and develop a CP-based general framework specially designed for PLE. In the first stages of our study, we conducted a systematic mapping study to examine the state of the art, and provide an overview of the application of CP in the PLE. As a result, we proposed a classification framework, built a comprehensive collection of constraints for PLE, characterize the PLE's activities applying constraint-based methods, and designed a first proposal of our CP-based framework for PLE. As future work, we will continue the developing of our proposal, and demonstrate its application to a case of study.

6. References

Benavides D., Trinidad P., Ruiz-Cortés A., "Automated reasoning on feature models", *Advanced Information Systems Engineering*, Springer, pp. 491–503, 2005.

Czarnecki K., Kim C. H. P., "Cardinality-based feature modeling and constraints: A progress report", *International Workshop on Software Factories*, pp. 16–20, 2005.

Mazo R., Muñoz Fernández J. C., Rincón L., Salinesi C., Tamura G., "Variamos: an extensible tool for engineering (dynamic) product lines", *Proceedings of the 19th International Conference on Software Product Line*, ACM, pp. 374–379, 2015.

Mazo R., Salinesi C., Diaz D., "Abstract Constraints: A General Framework for Solver-Independent Reasoning on Product Line Models", *INSIGHT-Journal of International Council on Systems Engineering (INCOSE)*, vol. 14, num. 4, pp. 22, 2011.

Peffers K., Tuunanen T., Rothenberger M., Chatterjee S., "A Design Science Research Methodology for Information Systems Research", *J. Manage. Inf. Syst.*, vol. 24, pp. 45–77, 2007.

Pohl K., Böckle G., van der Linden F. J., *Software Product Line Engineering: Foundations, Principles and Techniques*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

Rossi F., Beek P. v., Walsh T., *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*, Elsevier Science Inc., New York, NY, USA, 2006.

Salinesi C., Mazo R., Djebbi O., Diaz D., Lora-Michiels A., "Constraints: The core of product line engineering", *Research Challenges in Information Science (RCIS), 2011 Fifth International Conference on*, IEEE, pp. 1–10, 2011.