Configurable system or product line?

Pascal Krapf, Nicole Levy, Sébastien Berthier

Configurable system or product line?

Pascal Krapf
Syscience
Villebon sur Yvette, France
pascal.krapf@syscience.fr

Nicole Levy
CEDRIC-CNAM
Paris, France
nicole.levy@cnam.fr

Sébastien Berthier
Syscience
Villebon sur Yvette, France
sebastien.berthier@syscience.fr

I. INTRODUCTION

A configurable system defines and describes a set of systems. There are several type of configurable systems:

- a system can have typed parameters that are values belonging in a finite set of values. Some logical constraints can be defined to ensure the coherence of the configuration.

- a system can offer a varying set of features. Some logical constraints can be defined to ensure the coherence of the chosen features.

- a system can be implemented or executed on different platforms that have to be chosen.

The question we want to address is to compare the process and the activities of system configuration and product definition within a product line. The problem is to identify common practices and possible differences between both approaches.

The activity of making a system configurable is therefore firstly to identify the possible variabilities, and secondly to select or define the identified variants. While this activity is often solved by introducing parameters, there are other alternatives.

We propose to use the well-known concept of a product line [1] to describe a set of defined products or a set of similar software systems that share a common, managed set of features and are developed from a common set of core assets [2]. A product line allows to abstract the construction from the configuration of the reusable components.

In the rest of this abstract, we will present the qualities targeted when configuring a system, the platform that manages variability as a product line in the manner of FODA's feature models [3] and some advantages of our approach.

II. QUALITIES TARGETED WHEN CONFIGURING A SYSTEM

Companies need a product line management framework (process and tool) that has the following characteristics:

- Operability: the number of actions to select parts is reduced and available to human decision, both for the first setup and for later updates,
- Evolutivity: it is possible to add features and parts to the product line and continue to use former versions. When the product line is enriched, new features are added along with new added constraints.,
- Reusability: parts and groups of parts can be reused securely without modification in new products or new product lines,
- Simplicity: no deep knowledge about the system design is necessary to select a variant,
- Modularity: Architects can select coherent subsets of the product line by the selection of sets of variants,
- Consistency: Compliance with design rules constraining the choice of variants is ensured. These design rules can be of a norm, regulations or chosen method to be followed.

To guarantee these properties, we introduce product lines as a way of developing configurable systems.

III. PBS AND FEATURE MODEL

A product line is composed of 2 elements: library of reusable assets associated to a feature model. This latter describes the reasonings and the decisions to be taken to develop a system.

A. *Library of reusable assets*

System engineering [4,5,6,7,8] is the general framework used to develop complex products. A product is broken down into systems. Each system is broken down into subsystems, and so on until reaching individual parts that can be subcontracted and called Product Breakdown Structure (PBS). Parts are organized in a tree structure.

Software engineers generally build their software with software components. The software components library is similar to the PBS. General software qualities like cybersecurity, energy consumption efficiency, human machine interfaces, etc. are often managed as possible variants. The random selection of software components does not ensure the quality of the final product.

In both domains, the library of reusable assets can contain elements that are part of models, documents, tests reports, etc.

B. *Feature model*

Variants cannot be summarized as a single PBS due to the different kinds of assets likely to be involved. Companies describe the variants in a feature model. It describes variable features that can be selected for an individual product within the product line. Feature Diagrams were first introduced by Kang as part of the FODA (Feature Oriented Domain Analysis) method back in 1990 [9]. They are a family of modelling languages such the one described by K. Czarnecki [3], used to address the description of products in product lines.

A feature model can be represented as a directed acyclic graph (DAG), which may be more expressive than a tree [10]. Nodes are features that can be selected. They represent decisions that can be taken: selection of functionalities, non-functionalities, but also actors or elements of the context. Il is possible to express within the tools the different properties a feature can represent: a mandatory or an optional choice, an exclusive or an inclusive option. Constraints are “require” and “exclude” relations between features. The challenge of a well-defined product line is to design it so that the feature model contains constraints that are made simple, with no circular reference. If this condition is satisfied, then any path in the feature model allows to valuate variables into a coherent set defining an individual product.

IV. PROPOSITION

To be efficient, companies need a framework of combined and coherent processes, methods, and tools. We developed a tool to manage a product line. As for system engineering, the key success factor is to make people working in different domain understand each other and communicate efficiently, it is crucial to provide graphical intuitive views. The platform under development enables both, to construct a product line, with its associated PBS and then to be easy to use when choosing a configuration, by users with basic system knowledge. Configuration is easy as the visual is graphical and allows to describe a complete product line with its different possibilities as described in [6]. Our platform aims also at creating system product lines that satisfy the properties given above.

V. REFERENCES

[1] A. Le Put, L’ingénierie système d’une ligne de produits, Cépadues Editions.

[2] Paul Clements and Linda Northrop, Software Product Lines : Practices and Patterns, Addison-Wesley Professional, 2001.

[3] Czarnecki, Krzysztof; Eiseneckerr, Ulrich W. (2000). Generative Programming : Methods, Tools, and Applications. Addison-Wesley.

[4] IEEE1220 (ISO1220): Standard for Application and Management of the Systems Engineering Process.

[5] IEEE15288 (ISO15288): Systems Engineering - System Life Cycle Processes.

[6] IEEE1471 (ISO1471): Recommended Practice for Architectural Description of Software-Intensive Systems.

[7] EIA 632: Processes for engineering a system.

[8] NASA SEH: NASA Systems Engineering Handbook.

[9] Kang, K.; Cohen, S.; Hess, J.; Nowak, W.; Peterson, S. (1990). Feature-Oriented Domain Analysis (FODA) Feasibility Study (PDF) (Report). Pittsburgh : Software Engineering Institute, Carnegie Mellon University.

[10] P-Y Schobbens, Patrick Heymans, Jean-Christophe Trigaux, Yves Bontemps, Generic Semantics of Feature Diagrams, Computer Networks, February 2007.