# TOWARDS A CONFIGURATION MANAGEMENT INTEGRATION TO FEATURE MODELS IN MODEL-BASED PRODUCT LINE ENGINEERING

**Lameh, José (1,2);**
**Dubray, Alexandra (2);**
**Jankovic, Marija (1)**

1: Université Paris Saclay - CentraleSupélec, Laboratoire Genie Industriel, Gif-sur-Yvette, France;
2: Renault, Technocentre, 1 Av. du Golf 78288 Guyancourt, France

## ABSTRACT

In parallel with the industry 4.0 revolution, customers' demand and number of product requirements are increasing, inducing increased product variability. System complexity is growing with the new technologies and system architectures. Hence, maintaining a system consistent and in the desired state becomes crucial. This paper tackles two problems: a product's variability and its temporal evolution. Regarding variability, Model Based Systems Engineering (MBSE) methods and Product Line Engineering (PLE) techniques become essential to configuring products by selecting and arranging features in a combination. In parallel, versioning and temporal evolution are managed with the configuration management (CM) principles. A versioned feature model (FM) is proposed. In PLE methodology, the first step is to model variability through traditional FM. This research presents an extension of the FM by integrating CM into it. The versioned FM includes not only features' variants but also their versions, making it evolve in time. This work is presented as a base for a work that will study the application of CM for PLE. This study is done in an automotive industry context at Renault Group, and the model proposed is applied to Renault's systems.

**Keywords**: Configuration Management, Product families, Product line Engineering, Product modelling / models, Systems Engineering (SE)

**Contact**:
Lameh, José
CentraleSupélec Université Paris Saclay
France
joselameh@hotmail.com

**Cite this article:** Lameh, J., Dubray, A., Jankovic, M. (2023) 'Towards a Configuration Management Integration to Feature Models in Model-Based Product Line Engineering', in *Proceedings of the International Conference on Engineering Design (ICED23),* Bordeaux, France, 24-28 July 2023. DOI:10.1017/pds.2023.359

## 1 INTRODUCTION

Today, while the emergence of industry 4.0 and the convergence of the virtual world exploit the advantages of both customisation and mass production, resulting in a cost-effective production system (Bossen et al., 2014), companies are under increasing pressure from their customers, who want a greater variety of products with the latest technological options. If this technological boom is a source of wealth, it is a head of challenges to be met above all. On one side, the product diversity and hence, the existence of different variants increases the complexity of component configurations (Beuche, 2017). Product variability management that controls the dependencies between variants and coordinates variability between assets becomes essential (Forlingieri, 2022). In this context, Product Line Engineering (PLE) development tools and methodologies help streamline and systematise the creation of a single product to be able to reuse most of the common assets between the different variants for its applications (Lindohf et al., 2021). PLE focuses on creating different variations of the same product, making production management as effective and efficient as possible, allowing to reduce costs and time, and getting a high-quality product. On another side, the complexity of electrical and software components makes traditional configure-to-order approaches even more challenging. Current solutions struggle to support versioning across multiple domains, resulting in duplicate data, errors, and overwhelmed resources (Saniuk and Waszkowski, 2016; Wolff et al., 2021). The problem of the existence of different versions due to the temporal evolution of the features arises during divergent phases of product development. This complicates the coordination of the product's development and its maintenance. This is where configuration management (CM) is typically used to help maintain control of a product or service throughout its life cycle (Peng, Zhao, and Zhu, 2006). This is achieved by maintaining the baseline describing the attributes of a system at a point in time, and by managing changes (Conradi and Westfechtel, 1998). Previously mentioned issues are both important in terms of product management. First, Model-Based Product Line Engineering (MBPLE), which is a combination of PLE and Model-Based Systems Engineering (MBSE), targets the issue of variability often through Feature Modelling (FM). Second, CM or version management focuses on the temporal evolution of the product's assets.
This research explores the possibility of integrating CM into MBPLE and therefore managing versions of several product line variants. Although PLE techniques usually focuses on software (SPLE), in this research we focus on the product from a systemic approach, tackling several types of product's assets, such as physical components or engineering features and functionalities while following CM principles. The paper is organised in the following order. After the current introduction, a state of art is presented. Materials and methods are presented in section 3. Then, the results where the versioned feature model are described along with some examples, and application practices are shown. As an application, an adapted approach to address complex technical systems adjusted for Renault will be proposed. Finally, before concluding, future perspectives are expressed.

## 2 LITERATURE REVIEW

Along with the two identified issues, the literature review will be organised in two parts: The first axis is for variability management, where the concept of diversity is studied, and the notions of core assets and variants are detailed in the Systems Engineering domain. The second axis concerns temporal evolutions and products' life cycle. The notions of versioning or CM (baselines, change) are evoked.

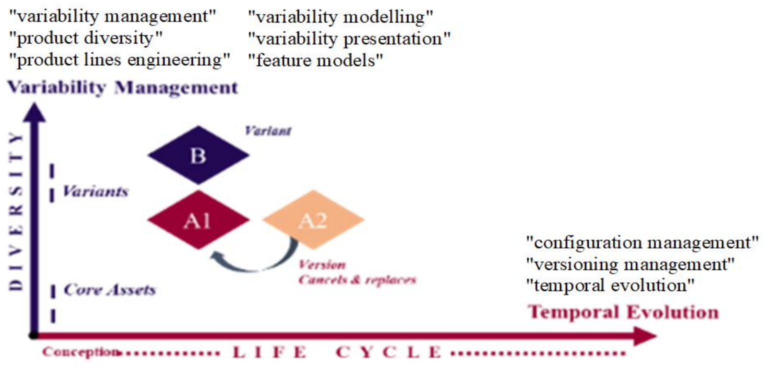

Figure 1. Research axes

## 2.1 Variability management

### 2.1.1 *Variability and product line engineering*

When the American industrialist Henry Ford declared that customers could have a car painted any colour, so long as it is black, the concept of Variability Management (VM) and its importance were not yet considered. In fact, VM refers to the ability to configure, customise, and exchange an artifact (any entity of a product) (Bachmann and Clements, 2005), throughout the lifecycle (Schmid and John, 2004). Today, products developed at a rapid pace and exploiting a rapidly changing technology base are to accommodate a growing demand for features across a wide spectrum of feature categories (Díaz et al., 2011). Several variants of the same products are created to respond to the same functions of use, but in diverse ways according to the customers' tastes and needs. However, this needs to align with business goals such as improving efficiency, quality, and productivity while limiting costs, time, and the use of resources, and optimising flexibility and configurability. This is where PLE is needed.
In production, PLE development tools and methodologies helped streamline and systematize the creation of a specific product, so in many cases, there was a need to reuse most of the common assets between the different versions of the applications. PL focus on creating different variations of the same product, making production management as effective and efficient as possible, allowing for to reduction of costs, time, and high-quality product (Dumitrescu, Salinesi, and Dauron, 2011). In PLE, products are divided into two main categories, core assets, and variable parts:

- First, the core assets that are shared between all the products of the line. "A core asset is an artifact or resource built to be used in the production of some products in a product line" (Clements and Northrop, 2002). Shared assets could be but are not limited to (i) architecture, (ii) documentation, (iii) specifications, (iv) software components, (v) tools such as component or application generators, (vi) performance models, (vii) schedules, (viii) budgets, (viva) test plans, test cases, work plans, (x) process descriptions… (Clements and Northrop, 2002).
- Second, the variable parts that once applied to the product, make the different variants of the product differentiable. Several definitions are given and established for variable parts. For Ouali et al., a variable part is "the position in an asset where a variation can take place. Everything else is a common part" (Ouali, Kraiem, and Ben Ghezala, 2022). For Grup, it is "the ability of a system to be efficiently extended, changed, customised or configured for use in a particular context (Gurp, 2000). Others are more precise and mention that it is the ability "to support the production of a set of artifacts that differ from each other in a pre-planned fashion" (Bachmann and Clements, 2005).

The most commonly accepted definition of a Product Line comes from Clements (2001) where PL is defined as "a set of systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way" (Clements and Northrop, 2002). With PLE, a faster way to market demands is obtained. PLE allows manufacturers to deliver the diversity of customers' demands while increasing profitability. Variability is defined and shared across all domains and the entire lifecycle will be created. Product options are managed independently. This creates a common source of configuration data that can be leveraged across mechanical, electrical, and software domains. The integration of PLE within PLM supports the entire product life cycle, allowing us to take advantage of robust product planning and system modelling capabilities in all areas. One can guarantee the accuracy of the configuration information and its proper understanding by each of the stakeholders. This could be summarised in the following illustration:

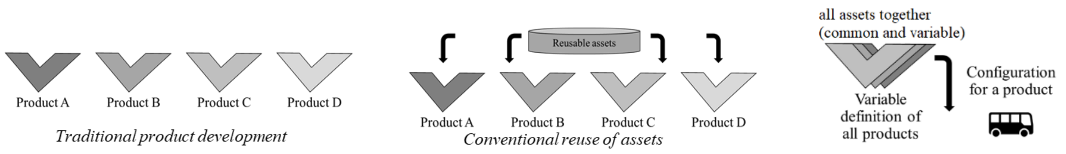
Product A
Product B
Product C
Product D
Traditional product development
Reusable assets
Product A
Product B
Product C
Product D
Conventional reuse of assets
all assets together (common and variable)
Variable definition of all products
Configuration for a product

*Figure 2. Traditional product development vs Conventional reuse of assets vs PLE*

A product line is a family of similar products with some variations in features. (1) However, traditionally, several products were developed independently. (2) Conventional reuse of assets was a first step to optimize the development of the different products. (3) PLE refers to the engineering of a portfolio of the related process using a shared set of engineering assets and defining variables of all the products in an efficient means of production to proceed to the configuration for a product.

### *2.1.2 PLE methodology and modelling variability*

PLE's methodology is expressed in (ISO/IEC 26580, 2013). At this stage, modelling variability through the feature model lies under domain engineering which is responsible for developing the common elements of the domain: studying it, defining its scope, defining the features, and implementing the reusable core assets and their variability mechanism. It is to be noted that several variability modelling techniques exist. A systematic review was realised (Chen, Ali Babar, and Ali, 2009), and the kinds of approaches reported in the papers studied were diverse. One of the most common approaches to model variability is Feature Model (Wahyudianto, Budiardjo, and Zamzami, 2014). Feature Models - established in the domain of engineering, more specifically in the problem space - describe the products of a product line in terms of the features that are common to those products and the features that vary between those products. Each feature in a Feature Model represents a property of a product that will be visible to the user of that product. In other words, all products and their features in a product line are represented. Some features are mandatory and are present obligatory for all the products in the line. Some of them are variant, so optional (Ouali et al., 2013). An FM is hence a model that defines features and their dependencies, in the form of a diagram which is visual notation or an and-or tree. This latter depicts the features of a solution in groups of increasing levels of details. An FM is composed of features and constraints (Kang et al., 1990; ter Bee, Gnesi, and Njima, 2011).

1. **Features:** A feature is a "characteristic of a product in a product line that distinguishes it from other member products in the product line" (ISO/IEC 26580, 2013). Features can express (a) the customer-visible or end-user-visible diversity among the products in a product line, or (b) distinguishing implementation diversity not directly visible to a customer or end-user except through non-functional differences such as price, performance, noise, weight, energy, and more. In PLE, features describe differences among products. A capability or other characteristic common to all products in the product line is not modelled as a feature (ISO/IEC 26580, 2013).
2. **Features' properties:** Features can have one of the two UML (Unified Modeling Language) properties: (i) A Boolean: Each feature that can be chosen or not (yes/no features), or (ii) an enumeration: Each feature that can have multiple choices. Features can be: (i) optional: they may be present in a product or not, (ii) mandatory: they should be present in a product, or (iii) alternative features: a set of features among which one and only one is present (Kang et al., 1990). Example 1: Adaptive cruise control is a feature that can have several feature values: Not existing, Stop and Go, and Predictive. Example 2: Countries are considered as a feature that can have several values: Europe, China, MEA, USA…
3. **Constraints :** The two categories of constraints are: (i) "requires" which is a unidirectional relation between two features indicating that the presence of one feature requires the presence of the other, and "excludes" which is a bidirectional relation between two features indicating that the presence of either feature is incompatible with the presence of the other (ter Beek, Gnesi and Njima, 2011). Constraints are added to a feature diagram.
4. **Parents and relationships:** Features can be a parent for other features, we will call them feature groups. The root feature group is the first parent and the highest level of our feature model. It is represented at the top of the tree.
5. **FODA notation :** According to the Feature-Oriented Domain Analysis (FODA) notation, a feature model would be illustrated as follow (Kang et al., 1990).

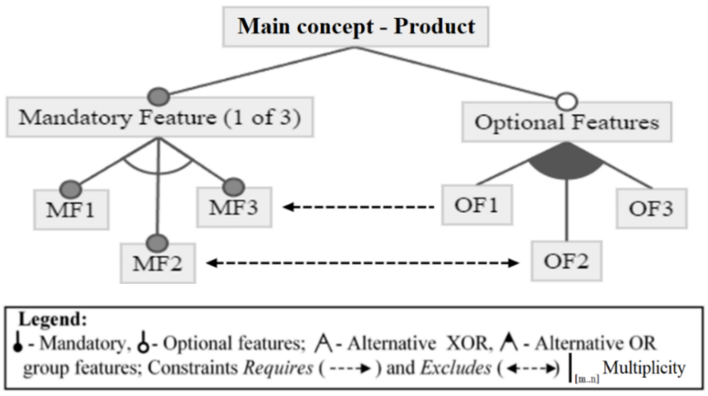
Main concept - Product
Mandatory Feature (1 of 3)
Optional Features
MF1
MF3
OF1
OF3
MF2
OF2
Legend:
- Mandatory, - Optional features; - Alternative XOR, - Alternative OR group features; Constraints *Requires* ( ---→ ) and *Excludes* (←--→) [m..n] Multiplicity

*Figure 3. Feature model notation*

https://doi.org/10.1017/pds.2023.359 Published online by Cambridge University Press

Once modelled, choices of variability are made, and customization is decided: a configuration is created. It is a set of features that describes a member of a Product Line. In UML it is called an Instance model. A member contains a feature if and only if the feature is in its configuration. A configuration is valid if and only if it does not violate constraints imposed by the model. If all the features are defined (instanced), we have a complete product configuration. In the car case, a final vehicle physical configuration is done (Dumitrescu, 2011). If not, it is a partial configuration i.e., a car family. **Product configuration is not to be confused with configuration management presented in 2.2.**

## 2.2 Configuration management

With time, the different parts of a product evolve i.e., multimedia system upgrade. Maintaining consistency is hence essential. In fact, change, baselining, product structure, and verification of a product are factors that are present throughout the life cycle of a product (Xing, 2010). Therefore, it is a priority to keep rigorous control and record of it. This control must be aimed at reducing errors and increasing quality and productivity. It should also avoid the inconveniences that can cause a bad synchronization (adjustment of clock causing a set of data or files to remain identical in more than one location.) of the configuration elements. Having a temporal and versioning model for this change and evolution is essential (Galante et al., 2005). The Configuration Management of a product is the discipline of managing the product structure and its describing information. It consists of a set of methods that make it possible to trace the configuration of this product and to control its conformity throughout its life cycle (van Ommering, 2001). CM is a product and information management discipline that indicates technical procedures for controlling product's information consistency and quality particularly when changes occur. The sub-processes or phases that make it up are (i) Configuration Identification, where the product is broken down into elementary elements with their relationships, (ii) Baselining or product configuration release is the approved selection of mutually consistent configuration item variant versions, organised according to a product structure. The maturity level of the records it contains must be consistent with the expectations at the milestone. It serves as a reference for the activities conducted from its formalization until the formalization of the next reference configuration. (iii) Physical and functional product configuration compliance verification checks if all the items are built correctly and are homogeneous to the characteristics and specifications. (iv) Change Management or change control is the operation of modifying (deleting, adding, modifying) configuration through a standard process (Sheng, 2019). Each of the phases has parameters for control processes that seek the integration of activities related to product development from the first phases, assigning roles and responsibilities to the working staff, to obtain a quality product; in this way, it is expected to be able to carry out a change in an efficient manner (Sheng, 2019). In an organization, developers, testers, project managers, system quality personnel, and customers can all benefit from the product configuration management process; among the following (Keyes, 2004): (i) it provides the ability to track changes during development, be it sequential or parallel, (ii) it organizes the tasks and activities that maintain the integrity of the product, (iii) it ensures correct product configuration, (iv) it ensures that engineers correctly implement changes to the baseline, (v) it helps reduce the cost of product life cycle maintenance, , (vi) it provides information for reports that can be easily generated, (vii) it allows quick and easy audits, (viii) it helps in the production of a higher quality product.

## 3 GAP AND RESEARCH QUESTION

The literature review underlines the importance of both VM and CM. However, if the FM presented could be the base for PLE and hence manage variability, it is still not a solution for versioning problems. In fact, this variability must be accompanied in engineering by a strong approach to standardization and modularization. Several research proposed system architecture frameworks to address these issues such as product line architectures (Clements and Northrop, 2002). Moreover, the notion of temporal evolution should also be considered. Siddiqi et al. identify Multi-ability, evolvability, and survivability as three main drivers for the need for reconfigurability (Siddiqi and de Weck, 2008). These approaches are only starting to be addressed in large complex systems such as vehicle development as they need to address both modelling and knowledge capture to represent and manage product variability, and the CM process. With this background, this research aims to integrate CM to variability PLE techniques by answering: How can configuration management principles be integrated to feature models in Model-Based System Engineering and hence MBPLE? How can aspects of the product other than its software be modelled?

## 4 METHODOLOGY

The state of the art was a fundamental step to do a synthetic state of work and understanding models and theoretical advances previously made. Drawing up a state of the field consists in conducting targeted and in-depth research, of all the pre-existing information concerning this said field, and producing a synthesis. Literature reviews not only develop knowledge but create guidelines, provide evidence, and help the generation of new ideas, and concepts for a particular field (Snyder, 2019). In parallel, the discovery and exploitation of the computer tools are as important as the theoretical part, especially to be able to validate hypotheses and verify that what is conceivable to be carried out can see the light of day. Handling an MBSE tool (MagicDraw) was therefore an essential part of the research. In this context, the focus group qualitative method was used (Masadeh, 2012). Engineers and experts in their fields meet to progress around a specific topic and axis of research. The data and the information collected, the several hypotheses established before making conclusions and validating results were the fruit of this method, of structured and focused daily and weekly discussions between systems engineering experts, and configuration managers. The collected information and outcomes of the meeting answer a limited number of questions defined in advance. It helps in exploring uncharted lands, identifying, and deepening understanding of a specific problem and complex (factors difficult to measure objectively), and providing a range of ideas and experiences. Moreover, workshops and proof of concept were done. Interviewing and contacting experts was a huge source of validation and verification. Several qualitative interviewing methods were applied (Meuser and Nagel, 2009; Döringer, 2021). These helped investigate experts' perspectives for a better understanding of modelling and problem-solving.

## 5 RESULTS, DISCUSSIONS, AND APPLICATIONS

In this part, the feature model that includes CM is presented, followed by an application to Renault.

### 5.1 Two properties of a feature: the revised feature model

A feature model usually represents the first axis of our research since it reflects variability and helps to manage it. The added value of the research is that the model will evolve on the second axis by following the principles of CM. Variability modelling is one of the key issues that need to be discussed and decided. Several possibilities are available. For instance, one central model versus several local models could be modelled. We could have a flat versus a tree model. From another perspective, in terms of tools, several options are available, and several ways can be used to achieve the same – or a different – solution. In the case of MBPLE, we work on a whole large model which is the system. This could be decomposed into many sub-systems. MagicDraw Tool by Dassault Systems was utilised for testing. In this platform, the root of the Feature Model is a Class with the «RootFeatureGroup» stereotype applied. Features can be placed in the root feature group (class) directly or can be placed into a subgroup. In the latter case, a subgroup is modelled as a separate UML class with the stereotype «FeatureGroup» applied. The subgroup is connected (linked) to the root feature group with composition. The composing property must have the stereotype «Feature» applied. In our case, from an engineering perspective, the features are the alternatives of technical solutions, on their engineering perimeter (system/subsystem). By integrating the time dimension, thus temporal evolution, and life cycle, we will have the versions. A version of a system variant (subsystem) is characterised by the set of versioned documents containing requirements and specifications. It corresponds to a temporal evolution to converge its maturity with other disciplines or to consider patches or improvements that have no connection with a new criterion. Note that a version can be applicable for several vehicle or platform milestones (Baselines). Figure 4 is an example of defining a new variant. At milestone 6, a new variant (B) of a feature is decided, resulting from variant A but different from it, by its functionalities or its solutions.

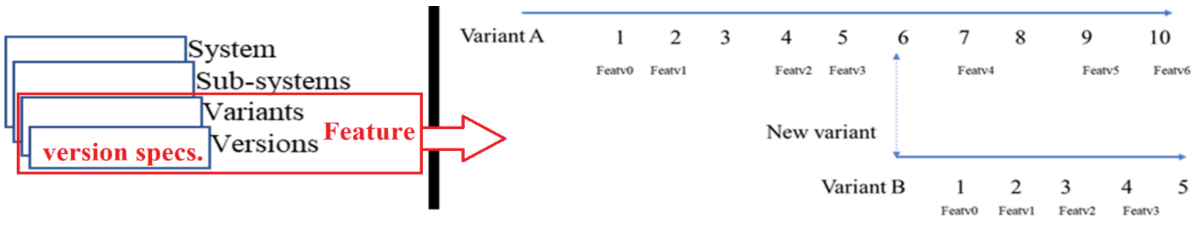

*Figure 4. Features and an example of feature variants and versions*

https://doi.org/10.1017/pds.2023.359 Published online by Cambridge University Press

This being described, the new layer which is a version of a feature will be included in the model. Hence, the model will be versioned and updated whenever one or several components change in the model. This is done according to the CM principles described in part 2. Baselines and change process guidelines are respected. The software alerts the user each time an element of the model evolves since the last update. Each time a feature is added, modified, or deleted; a new version of the whole model could then be generated. An example is given in figure 6. Here, after an update, a feature is added (FeatD), a feature is deleted and thus obsolete (FeatZ), and a feature is updated (FeatC v1 to FeatC v2). Instead of changing the whole feature model, the model is also versioned according to the CM rules to a version1.1 updated.

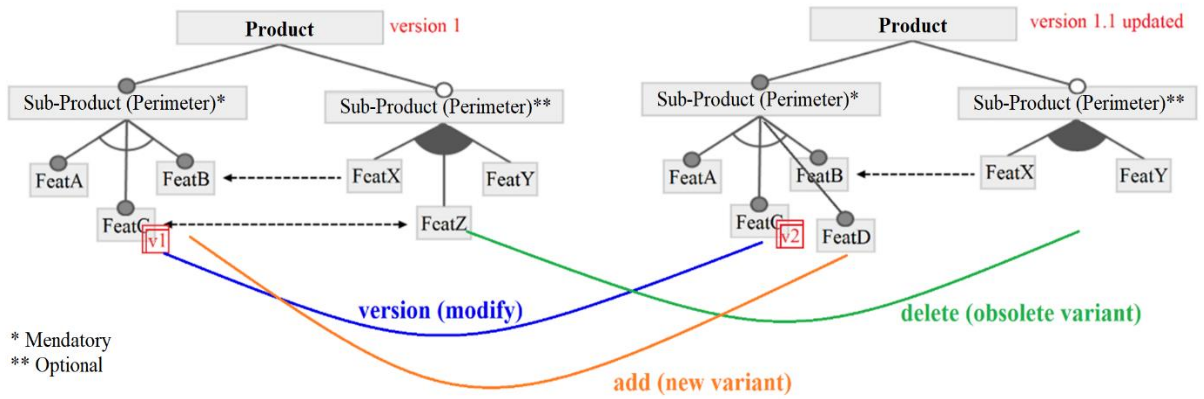

*Figure 5. example of feature variants and versions*

## 5.2 Case study: Renault

The history of the car manufacturer Renault is punctuated with innovations, from its creation in 1898 to the present day. Engineering has always been at the heart of transformations. Today, the automotive and mobility adventure continues and reinvents itself, even though Renault's products are more and more complex (Renault, no date). There is an exponential use of calculators and software, in addition to reinforcement of regulatory constraints. Until now, to meet these challenges, each "métier" used its own tools. However, these tools have shortcomings concerning sharing and access to information, but above all, they have a real lack in terms of traceability. Indeed, the robust mode of operation must evolve to adapt to new challenges, particularly related to autonomous vehicles and new regulations. In particular, the physical component mesh approach is no longer sufficient, and engineering must strengthen its systems engineering and requirements engineering practices. The intangible components (non-physical i.e., software, engineering features…) represent a significant part of the value of the vehicle, and these require different management modes. In this context, Renault is developing advanced and present-time ways to manage products' life cycle that includes CM techniques, systems engineering, and PLE.

## 5.3 Application at Renault and deployment

In the car industry, variability modelling problems are crucial (Xu et al., 2021). At Renault, the Variability Management is based on the variability model that describes the diversity for all vehicle projects with a lexicon of technical diversity drivers. This model also describes the constraints between the several criteria of the lexicon. It is called the Configuration Management Lexicon or CML, and it shows the technical variability in engineering projects. The CML is composed of the lexicon with objects and criteria. One object is a "customer or engineering or regulation need" and the criteria attached to an object represent all the possibilities that engineering must satisfy. Moreover, the CML contains object trees that show the dependencies between objects. Constraint tables reflect the links between the criteria of these objects (compatibility, combinations…). The CML is versioned and translation tables between criteria of different versions are built so the continuity in the description of the technical applicability of configuration items is assured. This is where the second axis plays a role. Cross-functionality at the product family level has been studied for quite some time. The concept of the product line at the level of several vehicle families and therefore a multi-brand product line is new.
As an application to the work done in section 5.1, the feature model will be created based on the CML to contain the whole diversity model. This model will hence include variants and versions of the

https://doi.org/10.1017/pds.2023.359 Published online by Cambridge University Press

features and will be then used in the MBSE approach and PLE methodology. The full MBPLE steps and hence the use cases of our FM are out of the scope of this paper and will be tackled at later stages. Once modelling realised, the variation points will be placed, and the diversity configuration would be done with the creation of feature profiles. These are lists of desired profiles that are defined according to possible instantiations of the feature model. System variants of sub-systems could be used to create thousands of combinations together and hence generate more possible instantiations. Users could check how a choice of a specific feature will affect several systems' functionalities and engineering features. Using the latest model's version, we ensure that systems requirements are respected and that the features chosen are not outdated. An update in the feature and hence in the feature model would also require an update in the system and hence a new version. Otherwise, an obsolete configuration will be created.
Moreover, the users could have a look on the whole exhaustive model (figure 6) or could work by perimeter and hence visualise and check what changes in a specific parameter. This could work in a systemic approach i.e., perimeter ss2, or in another organisation i.e., the impacts of featX. Many use cases (Feature Owner, System Architect, Vehicle System Architect points of view…) could be studied. This will be the content of future research.

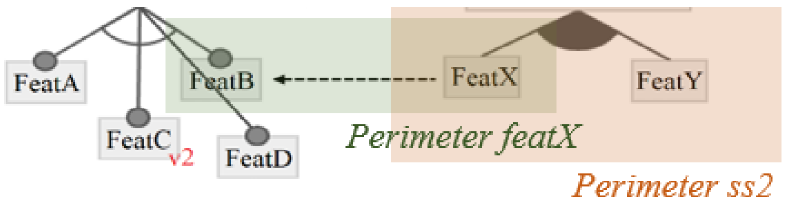

*Figure 6. visualising perimeters*

This section shows that the model will contain not only software components but also engineering features. As an example, Adaptive Cruise Control (ACC) is an engineering feature that can or cannot be present in a vehicle with or without the Stop-and-Go feature. Moreover, this feature can evolve in terms of variability creating more variants, and in time creating different versions. The model will hence be updated as explained in 5.1. When doing a configuration, users will be able to select what variant of ACC they need and will make sure to take the last version compatible with the product version. Baselines and CM principles would also be respected. Systems engineers could check ACC's influence on its perimeter and other components of the product. Therefore, integrating CM techniques in an FM that does not only tackle software will be achieved through MBPLE methodology.

## 5.4 Discussion

Today, feature-based PLE is used in several domains and various industrial sectors. Aerospace, security, aviation, and automotive manufacturers are saving millions of dollars because of these techniques. For instance, Boeing is a member of the Product Line Hall of Fame. In Bold Stroke Avionics, Boeing improved affordability, quality, and system timeliness (Ghafoor et al., 2005). Ericsson is also a member of the Product Line Hall of Fame. Through PLE, it showed adaptability to change and evolution, as well as got a significantly lower number of defects and code modifications (Mohagheghi and Conradi, 2008). Another nominee in the Product Line Hall of Fame is Robert Bosch Corporation which works on automotive gasoline systems. When applying PLE, Bosch consumed 25% less memory, reduced the calibration effort by 20% and maintenance, and reduced the resource consumption (by 20%-30%) (Nord, 2004). At Renault, the study of the PLE for the “Trafic” van product family showed that more than 10^21 configurations of the product were available (Astesana, Cosserat, and Fargier, 2022). Product line adoption involves moving from some form of developing intensive systems with a single-system mentality to developing them as a product line. Many companies have not adopted PLE yet and still have not a core asset base and supportive processes and organizational structures. This is usually due to many barriers to product line adoption such as cost, and time needed to launch, educate, address cultural barriers, develop a core asset base, and lead the organization while continuing to do the business. More barriers could exist also such as lack of knowledge, the need for organizational change, cultural resistance, and incompatible development processes. A major challenge remains: to move from the traditional "opportunistic" reuse to a new innovative strategical, managerial, and organisational way of handling product life management along with PLE and CM techniques.

## 6 CONCLUSION

To conclude, this work aimed to integrate CM principles and mechanisms into variability management and PLE through the versioned feature model. The transformation process allowing for the integration of different functionalities in the feature model has been presented. The versioned FM allows for representing product variability through MBSE practices, and temporal evolution through configuration and versioning management. When a change occurs (adding, subtracting, or modifying) a feature, CM principles will be applied inside the new layer of the feature in the model. This model could be used as a solid base for PLE where fixing variability points, instantiating, configurations and then filtering the model could then be done. This shall be done at the next stage, where the finalization of the technique, the validation of the complete method through a complete tangible case study, and the comparison with other tools could be realized. The case study showed that not only the method proposed is a need, but also that CM integrated into PLE can take on today's main engineering challenges globally and on Renault's challenges specifically. The difficulty remains in the complexity of the systems, the time to implement these potential solutions, the need for collaboration between all engineering teams, and of course the writing of rules to be adapted and respected by the different actors of the project.

## REFERENCES

Astesana, J.-M., Cosserat, L. and Fargier, A. (2022), “Constraint-based Modeling and Exploitation of a Vehicle Range at Renault’s: Requirement analysis and complexity study”

Bachmann, F. and Clements, P.C. (2005), *Variability in Software Product Lines*. report. Carnegie Mellon University. Available at: https://doi.org/10.1184/R1/6585860.v1

ter Beek, M.H., Gnesi, S. and Njima, M.N. (2011), “Product Lines for Service Oriented Applications - PL for SOA”, *Electronic Proceedings in Theoretical Computer Science*, 61, pp. 34–48. Available at: https://doi.org/10.4204/EPTCS.61.3

Beuche, D. (2017), “Using pure: variants Across The Product Line Lifecycle”, in *Proceedings of the 21st International Systems and Software Product Line Conference - Volume B. SPLC ’17: 21st International Systems and Software Product Line Conference*, Sevilla Spain: ACM, pp. 50–50. Available at: https://doi.org/10.1145/3109729.3109754

Bossen, J. et al. (2014), “An Engineer-To-Order Mass Customization Development Framework”, in E. Bayro-Corrochano and E. Hancock (eds), *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Cham: Springer International Publishing (Lecture Notes in Computer Science), pp. 116–123. Available at: https://doi.org/10.1007/978-3-662-44733-8_15

Chen, L., Ali Babar, M. and Ali, N. (2009), “Variability management in software product lines: A systematic review, *Proceedings of the 13th International Software Product Line Conference*, p. 90. Available at: https://doi.org/10.1145/1753235.1753247

Clements, P. and Northrop, L. (2002), *Software product lines: practices and patterns*. Boston: Addison-Wesley (The SEI series in software engineering)

Conradi, R. and Westfechtel, B. (1998), “Version Models for Software Configuration Management”, *ACM Comput. Surv.*, 30, pp. 232–282. Available at: https://doi.org/10.1145/280277.280280

Díaz, J. et al. (2011), “Agile product line engineering - A systematic literature review”, *Softw., Pract. Exper.*, 41, pp. 921–941. Available at: https://doi.org/10.1002/spe.1087

Döringer, S. (2021), “The problem-centred expert interview”. Combining qualitative interviewing approaches for investigating implicit expert knowledge”, *International Journal of Social Research Methodology*, 24(3), pp. 265–278. Available at: https://doi.org/10.1080/13645579.2020.1766777

Dumitrescu, C., Salinesi, C. and Dauron, A. (2011), “Formalization and Exploitation of the Coupling between Systems Engineering Methods and Product Lines”, INSIGHT - *International Council on Systems Engineering (INCOSE)*, 14(4), pp. 12–13

Forlingieri, M. (2022), “The four dimensions of Variability and their impact on MBPLE: How to approach variability in the development of aircraft product lines at Airbus”, in *Proceedings of the 16th International Working Conference on Variability Modelling of Software-Intensive Systems*. New York, NY, USA: Association for Computing Machinery (VaMoS ’22), pp. 1–4. Available at: https://doi.org/ 10.1145/3510466.3511275

Galante, R. et al. (2005), “Temporal and versioning model for schema evolution in object-oriented databases”, *Data Knowl. Eng.*, 53, pp. 99–128. Available at: https://doi.org/10.1016/j.datak.2004.07.001

Ghafoor, A. et al. (eds), (2005), *Eight IEEE International Symposium on Object-Oriented Real-Time Distributed Computing: proceedings: ISORC 2005: 18-20 May, 2005, Seattle, Washington. International Symposium on Object-Oriented Real-Time Distributed Computing*, Los Alamitos, Calif: IEEE Computer Society van Gurp, J. (2000), *Variability in software systems: the key to software reuse*. Dept. of Software Engineering & Computer Science, Blekinge Institute of Technology.

ISO/IEC 26580 (2013), ISO/IEC 26580:2021, ISO. Available at: https://www.iso.org/cms/render/live/fr/sites/isoorg/contents/data/standard/04/31/43139.html (Accessed: 29 July 2022)

Kang, K.C. et al. (1990), *Feature-Oriented Domain Analysis (FODA), Feasibility Study*: Fort Belvoir, VA: Defense Technical Information Center. Available at: https://doi.org/10.21236/ADA235785

Keyes, J. (2004), Software configuration management. Boca Raton, FL: Auerbach Publications. Available at: http://www.books24x7.com/marc.asp?isbn=0849319765 (Accessed: 28 July 2022)

Lindohf, R. et al. (2021), "Software product-line evaluation in the large", *Empirical Software Engineering*, 26(2), p. 30. Available at: https://doi.org/10.1007/s10664-020-09913-9

Masadeh, M. (2012), "Focus Group: Reviews and Practices", *International Journal of Applied Science and Technology*, 2, pp. 63–68

Meuser, M. and Nagel, U. (2009), "The Expert Interview and Changes in Knowledge Production", in, pp. 17–42. Available at: https://doi.org/10.1057/9780230244276_2

Mohagheghi, P. and Conradi, R. (2008), "An empirical investigation of software reuse benefits in a large telecom product", *ACM Transactions on Software Engineering and Methodology*, 17(3), pp. 1–31. Available at: https://doi.org/10.1145/1363102.1363104

Nord, R. (ed.), (2004), Software product lines: *Third International Conference, SPLC 2004, Boston, MA, USA, August 30-September 2, 2004: proceedings. Software Product Lines Conference*, Berlin ; New York: Springer (Lecture notes in computer science, 3154)

van Ommering, R. (2001), *Configuration Management in Component Based Product Populations*, p. 23. Available at: https://doi.org/10.1007/3-540-39195-9_2

Ouali, S. et al. (2013), "A Model Driven Software Product Line Process for Developing Applications", in X. Franch and P. Soffer (eds), *Advanced Information Systems Engineering Workshops*. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Business Information Processing), pp. 447–454. Available at: https://doi.org/10.1007/978-3-642-38490-5_40

Ouali, S., Kraiem, N. and Ben Ghezala, H. (2022), "A Flexible Security Process for SPL construction'.

Peng, X., Zhao, W. and Zhu, C. (2006), *Integrating Configuration Management and Process Management Based on Life Cycle Control*, p. 196. Available at: https://doi.org/10.1109/CIT.2006.108

Renault (no date), *Renault's Internal documentations: internet and intranet websites*

Saniuk, A. and Waszkowski, R. (2016), "Make-to-order manufacturing - new approach to management of manufacturing processes", *IOP Conference Series: Materials Science and Engineering*, 145(2), p. 022005. Available at: https://doi.org/10.1088/1757-899X/145/2/022005

Schmid, K. and John, I. (2004), "A customizable approach to full lifecycle variability management", *Science of Computer Programming*, 53(3), pp. 259–284. Available at: https://doi.org/10.1016/j.scico.2003.04.002.

Sheng, R. (2019), "Chapter 6 - Managing People, Product, and Process (P3), implementation", in R. Sheng (ed.), *Systems Engineering for Aerospace*. Academic Press, pp. 75–112. Available at: https://doi.org/10.1016/B978-0-12-816458-7.00006-5

Snyder, H. (2019), "Literature review as a research methodology: An overview and guidelines", J*ournal of Business Research*, 104, pp. 333–339. Available at: https://doi.org/10.1016/j.jbusres.2019.07.039.

Wahyudianto, Budiardjo, E. and Zamzami, E. (2014), "Feature Modeling and Variability Modeling Syntactic Notation Comparison and Mapping", *Journal of Computer and Communications*, 02, pp. 101–108. Available at: https://doi.org/10.4236/jcc.2014.22018

Wolff, K. et al. (2021), "Effects Of Different Order Processing Strategies On Operating Curves Of Logistic Models: A Comparison Of Make-to-Order And Make-to-Stock'. Available at: https://doi.org/10.15488/11532

Xing, T. (2010), "Software configuration management of change control study based on baseline", in 2010 *International Conference on Intelligent Control and Information Processing. 2010 International Conference on Intelligent Control and Information Processing*, pp. 93–96. Available at: https://doi.org/10.1109/ICICIP.2010.5565288

Xu, H. et al. (2021), "Optimisation for the product configuration system of Renault: towards an integration of symmetries", in *Proceedings of the 25th ACM International Systems and Software Product Line Conference - Volume B. SPLC '21: 25th ACM International Systems and Software Product Line Conference*, Leicester United Kindom: ACM, pp. 86–90. Available at: https://doi.org/10.1145/3461002.3473948