Reuse / Variability Management and System Engineering

Olivier Renault
PLM Industry Expert
PLMSys Consulting
111, avenue Victor Hugo – 75784 Paris Cedex 16
olivier.renault.pro@gmail.com

Abstract: this paper aims to share industry experience in managing the reuse and variability in the industry and to analyze the linkage between this topic and system engineering. Reuse and variability management are two faces of the same coin and strongly influence business performance. Hard products industries (where mechanical and structural engineering were historically dominant) and soft systems industries (where electronic and software engineering are dominant) addressed the questions from different perspectives. After describing the observed practices for managing the reuse and variability from the physical product standpoint, and taking in account concepts and approaches used in “Soft” industries, we analyze how systemic approach should help in better mastering the variability. In conclusion, we identify some principles and rules which would need to be investigated through research to better link PLM and systemic approach in variability management

Keywords
Variety, Variability, Reuse, Options, Variants, Configuration, Product Families, Product Lines, Requirements, Features, Systems, Platforms, Modules, Architecture, Interfaces, Product Life Cycle, PLM Systems

1 Introduction

As we will see across this paper, variability and reuse management is a topic which crosses the full life cycle of complex products and systems and address all the engineering disciplines. One of the identified difficulties is that variability management have often been addressed separately by different methods and IT systems according to their positioning on life cycle and engineering disciplines. My own experience presented in the first sections of this paper, is mostly relevant to the management of the variability implemented in PLM and ERP systems. It focuses on variability management relying on product structure (from the classical perspective of BOM management supported by both classes of systems, but also from the perspective of the 3D Digital Mockups used by a large bench of engineering disciplines for concurrent engineering. But this perspective is too limited. I intuitively thought that system engineering should be analyzed in the variability management context for two main reasons:

- First, electronics and software is massively invading all traditional products whose design and development previously relied on structural and mechanical design. Even if these engineering disciplines are often treated with separated methods and dedicated IT application, mutual dependencies are growing. Complex systems/product definition need to be managed from a consistent point of view for configuration, including variability management and changes across time.
- Secondly, my experience focused on physical product variability management and the problems encountered convinced me that a systemic approach is required to better manage the variability. This relies on the fact that variability is strongly linked to the way we structure and manage the requirements in parallel with the definition of the system/product architecture.

This also pushed me to do a brief research review where I found interesting papers. A part of them were centered on the domains of the physical product variability. But I also discovered that a lot of the research papers address the question of variability from the software engineering perspective. These papers helped me to clarify and confirm some intuitions I got from my own experiences. They also help me to formulate the reasons why I think that system engineering and system modeling approaches may provide a strong foundation to design and model the variability. They provide a way to consistently articulate variability and architecture definition during the development process of complex systems /products.

2 Business approaches and drivers for variability management in industry.

We will mention here two industries which showed a strong concern on variability management, just to underline the business impact it had.

In the Information Technology industry, IBM introduced early in the 60s an ambitious modular and configurable architecture of business computers with the IBM 360 systems. This program was very successful and constituted one of the main reason of the further dominance of the market by IBM [Manet Hamm O.Brien 2011]). IT industry is now an industry where standards (OS, telecoms, DB, Internet….) and layered architecture took an essential part in its uninterrupted growth.

The automotive industry is probably one, in hard products domain, which faces among the most complex challenge to manage variability. A car is a technical complex object (several thousands of parts) and has to sustain a very large variability [Jiao, Simpson, & Siddique 2007]. [Volkswagen 2011] & [Renault Nissan 2013] summarize the respective approaches of Volkswagen and Renault-Nissan groups to manage their variability based on approaches to platforms and modules. We may summarize them by the following principles.

- Vehicles of these brands are organized by vehicle families. Often a vehicle family covers one market segment for a brand and includes all types of bodies for the brand and the segment, with different possible engines. One vehicle family generally reuse one single platform family (sometimes two for market–cost optimization).
- A platform family is generally common to several vehicle families of one or several brands and may cover one or more vehicle segments. The platform integrates the chassis and all equipment generally hidden to the customer while the complementary part of the vehicle includes all elements of style directly perceived by the customer.
- Common modules are designed to equip the whole range of platform families in order to maximize the scale economy. A similar function (e.g. a seat), may be implemented through one, or a very limited number of module families. Each module generally include the variability required to cover common market needs for all platform and vehicle families.

Multiple sources of research papers and articles develop the business drivers for variability and reuse management.

- [Jiao, Simpson, Siddique 2007] provides a comprehensive review of state of arts research on product family design and platform-based products As part of the work the economic justification section references most important papers on this topic..
- For the software industry, an economic model is proposed by [Rokunuzzaman & Choudhury 2006]. It estimates benefits to reuse software components for building a customized software solution. [Lim 1994] gives metrics collected during two reuse programs of Hewlett Packard.
- [Pil, Holweg 2004] analyzes the variability and its economic drivers focusing the study mainly on the automotive sector. They gives order of magnitude of the variability and focus on how the products variability has to be linked to the order fulfillments strategy. Their paper well illustrates the strong focus that hard products industries had to manage variability from the physical product and supply chain perspective.

Table 1 hereafter proposes a summary the main business drivers for the variability and reuse management for hard products industries.

Table 1

Drivers for Variability	**Drivers for Reuse**
(B2C) Customer diversity of demand (ie: Car bodies, painting, engine, equipment's …)	Supply chain costs and delays reductions • Production (scale effects) – Internal / External (supply chain) • Standardization / flexibility of production process and facilities • Capacity planning • Order to delivery delay • Quality improvement (repeatability) • Development reduction cost (initial and change management)
(B2B) Ordering company differentiation (ie: Aircraft companies specific cabin layout – engine – electronic systems variants, length, capacity, mission…)	Development delays / costs / risks • Development reduction cost (initial and change management) • Innovation value focus • Risk minimization • Tests validation reduction
(B2C – B2B) Country constraints • Regulations • Climate • Customer Usages • Infrastructures …	Innovation • More focus on value innovation • Faster introduction in existing products (standardized modular architectures)
(B2C – B2B) New Technologies introduction	Flexibility – Speediness to change and adaptation. (same changes to apply on a wider spectrum)

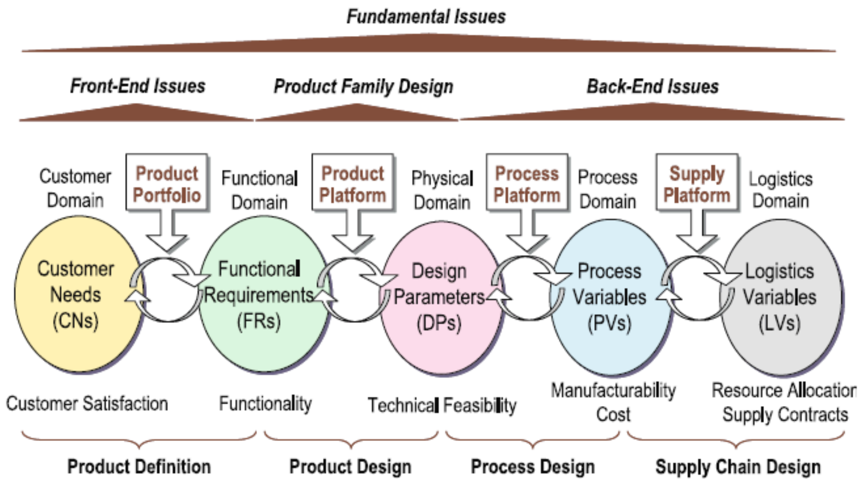
[Main business drivers for variability and reuse for hard products industries]

3 Approaches and gaps for managing variability in IT Systems.

3.1 Management of variability in CRM, MRP/ERP and PLM

As explained by [Pil, Holweg 2004], the variability management encompasses the full life cycle of the products. This is illustrated by Figure 1 hereunder reproduced from their paper.

Figure 1: [Pil, Holweg 2004] – Holistic view of Product Family Design and development.



CRM (Customer Relationship Management), MRP/ERP (Manufacturing Requirements Planning/Enterprise Resource Planning) and MRO (Maintenance, Repair and Operations), each of these systems manage the product during one of its "physical" life cycle stages. So they are impacted by reuse and variability management. Variability management requires capabilities for each of these domain systems. They may be standard capabilities provided by market software packages, but they also often rely on specific development extending these software package or working as stand-alone systems/applications.

The industry experience I present in this paper is mainly focused on practices used in PLM (Product Life Cycle Management). PLM is the system used to support the design and definition of products and of the life cycle processes and resources related to these products. The PLM supports and coordinates all engineering disciplines and manages all the technical information attached to the product and its product life cycle. So, PLM is placed at the critical stage where variability and reuse are designed. PLM architecture is built around a PDM (Product Data Management) system which offers central services to store, retrieve, classify, and configure all technical data (models and documents). The different applications or systems used to sustain each of the engineering discipline activities are commonly articulated and integrated with the PDM central system to build the overall PLM system architecture. Software engineering and configuration management remains relatively autonomous. Nevertheless there is need to better integrate them within the PDM systems to enable a consistent multi-level configuration management.

PLM variability definition has vocation to be the reference basis for the configuration models used by CRM and MRP/ERP systems or applications. The configuration models of the product families used by each system need to be maintained and synchronized across the change management process.

3.2 Main practices used in PLM for managing the variability

We present here PLM observed practices for managing the reuse and variability. We compare practices mainly applied for managing the reuse and variability in product structures representing the physical product. As explained in section 4, for hard product industries, requirements and functional variability were historically managed through documentation in a traditional development process. Impact of system and software massive intrusion in these industries changed the game. But, from our experience, we still see system and software and PDM managed very separately.

In the following, we will use the words

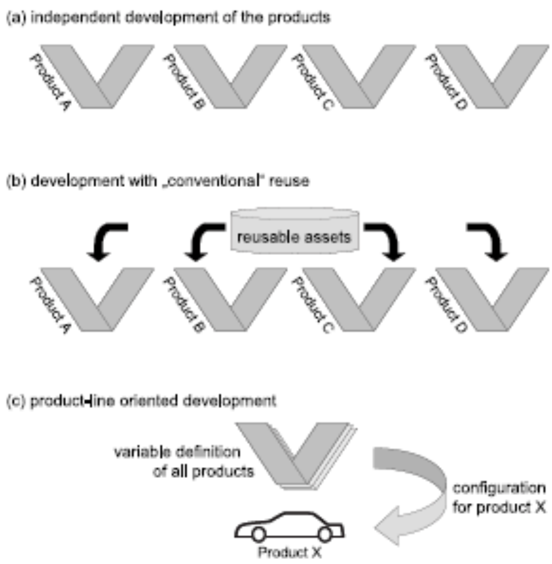
- system/product to specify the high level "final" system/product which has to be designed and developed and which represents the higher level of integration. If, in physical product structure, we should speak only on product, we extend the concept to system/product in the perspective of system integration in our analysis as presented in section 4.
- Sub-systems/modules to specify the element of a product/system definition which could be reused between different higher level systems/products. As the high level product is represented by a product structure defining its composition, a module may be itself defined by a product structure representing its own product composition.

The 3 dominant practices are summarized in Figure 2 extracted from [Reiser 2009]

<u>Practice a: Duplicate and specialize systems/products structures (independent development of products)</u>

The principle is to create a specific system/product structure for each (top level) product. The reuse is done by initial copy or several partial copies from structures of similar system/product. The inconvenience of this approach is that the duplication of common elements encourage the specialization of the definition, even if there is a significant business advantage to maintain a common definition.

Figure 2: Product portfolio development approaches [Reiser 2009]



Practice b: Separated products top structure sharing common configured product components (development with conventional reuse)

In this approach, common sub-systems/modules internal structure are instantiated in each top system/product structure, but remain unique. If the common sub-systems/modules holds variability, it is only the resolved configurations which are instantiated where needed. The main advantage is that this approach forces to maintain a better communality of sub-systems/modules across the different systems/products where they are used. The constraints and limits are:

- sub-systems/modules need to be designed to address requirements of future product-systems (at least at the architectural design level).
- changes to sub-systems/modules need to be controlled with all different upper levels systems/products using them.
- sub-systems/modules variants need to be explicitly configured (specific references) in the upper-level systems/products structure.
- on multi-level architecture, rules for managing and updating the configuration of the different systems/products (number of Configuration Items (CI) levels –revision number absorption levels and rules), and process for propagating changes on upper levels need to be carefully designed to minimize the number of revision updates.
- when there is a large number of combinations of options-variants, impacted by a change, the process to update and maintain all these combinations may be complex. Revision numbers have to be updated on

all parent structure representing these combinations and may need some PLM automation. A good approach is to group several changes and to apply revision number changes on the upper structure only when the group of lower level changes has been fully defined and validated.

- Another condition is to really have a unique structure to maintain the common part of different configured variants structures. On the contrary a change in the common part implies to update each configured variants where it is duplicated. This could be painful and leads to error if this is not automated in some way.

Practice c: Define a common system/product family structure.

This practice consists in creating a unique structure for product family holding the whole variability description for the family. This rely on options – variants mechanisms. This practice is detailed in the following section.

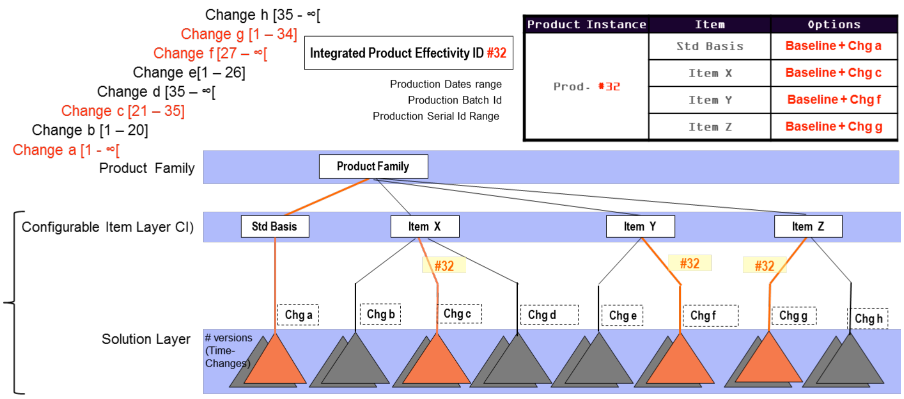
3.3 Practices used in PLM for managing system/product family (product lines oriented development)

Practice a: Product family unique structure carrying the variability description for the whole family by production effectivity

The principle is to have a unique structure for a family of products and to associate a production effectivity to the proper elements of the structure. A production effectivity is generally a set of serial numbers, a range of date delimiting a batch of production for the same products, a batch ID. The configuration of one specific system/product instance (physical product produced or planned to be produced) may be retrieve by selecting all structure items with a product effectivity matching its own production ID. This mechanism may also be implemented through a change management process as summarized in Figure 3 here under.

This approach seems well suited to industries where the variability is driven by a custom to order process where specificities of each variant/option cannot be anticipated and may be very specifically linked to the order requirements for customization.

Figure 3: Product family with variability managed by production effectivity



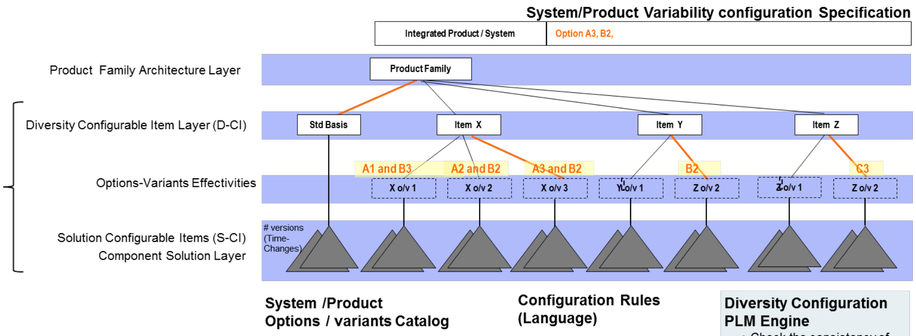
Practice b: Product family unique structure carrying the variability description for the whole family by variability effectivities and logical rules

This principles are the following:

1. Describe the possible option/variants by a language based on:
    a. variability criteria objects holding each one a "dimension of variability"
    b. variability criterion values, each value corresponding to a variability possibility inside the same variability dimension
2. Associate to item nodes in the structure a logical expression corresponding to the combination of option/variant values for which the structure under the item has to be retained. We name it variability effectivity or effectivity. This logical expression may contain logical operator such as NOT, AND, OR…
3. A set of rules may be added to define compatibilities or dependencies of different options/variants.
4. Finally, to define a particular product in the product family, we must select directly or indirectly one value for each option/variant criterion proposed at the family level. The selection request to configure one system/product instance may be explicit. In this case we must express each criterion object with its possible options/variants value (any number). Or it may be implicit (all values of criteria not specified are, by default, retained). The selection is then modified or rejected against the set rules for dependencies and compatibilities.

The Figure 6 summarizes the main principles for describing the variability of the product family in a unique product family structure.

Figure 6: Product family unique structure using based on variability effectivity and rules



Options/Variants Criterions	Options Variants Values,
Option A	A1,A2,A3
Option B	B1,B2
Option C	C1,C2,C3,C4
Option D	D1,D2

When both the number of potential combinations and the volume of production are high, this approach enables us to directly and dynamically "solve" the configuration of the configurable structure by selecting the options/variants value for each criterion. But there are some constraints and limits:

1. The system/product family configurable structure need to be properly defined (system/product items with, for each of them, the proper variability effectivity defining the options/variants for which this item substructure and definition is applicable). This implies the two following rules to be verified.
   - Any selection of options/variants values should not lead to an incomplete configured structure (no function/parts "holes" in the configured structure).
   - Any selection of options/variants values should not lead to a number of system/product items selected beyond the number expected (no function/parts "bump" in the configured structure).
2. If this approach is basically used, the variability of a sub-structure has to be configured by criteria defined for the overall system/product family structure. This could lead to the impossibility of reusing this configurable sub-structure in another system/product family. In the automotive industry, platforms and modules variability management cannot be done in a single system/product family without strongly complicating reuse.

Practice c: Multi-level Product families

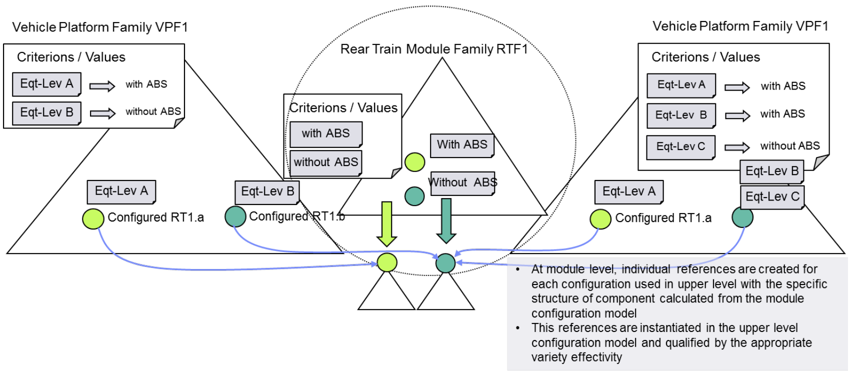
It may be necessary to manage variability at different systems/product structures levels. This is the case in automotive for vehicles, platforms and modules.

In this example, each level (vehicle, platform, sub-systems/modules) holding variability must be considered as a distinct system/product family (product line). The vehicle family structure will instantiate the platform family structure it reuses. Each of them (vehicle or platforms) will also instantiate the sub-system/module families structures they reuse. Applying variability effectivities principles, each family is supposed to have its own variability definition relying on a specific set of variability criteria and criterion values.

When configuring a specific configuration at the higher level (ie: vehicle) by selecting value for each criterion specified in this family, three approaches may be used to select the proper variability of the lower level families.

- Approach 1: Instantiate in the upper-level family structure only configured options variants of lower level family (see Figure 4 hereunder). Each configured option/variant of the lower-level family will be characterized in the upper level structure by a use-case (logical expression of options/variants of the upper family in which this structure has to be configured). The inconvenience of this choice is that the maintenance of the configured lower family structures may be heavy in case of changes. This is especially true when changes are located in their common parts and when the number of configured structures for the lower family is important.

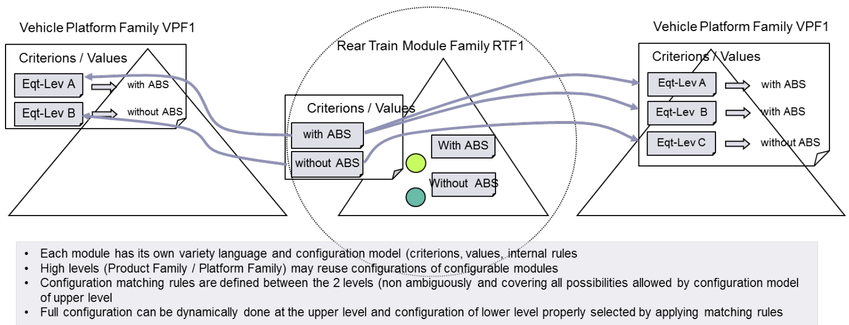
Figure 4: Instantiation of configured structure of the lower level family.



- Approach 2: Reflect all criteria and criterions values of the lower level families in the upper level. For example, all platforms and sub-systems/modules criteria and criterions values will be directly included and visible in the set of criteria used at the vehicle level. We may easily understand that this approach will strongly increase the number of criteria used at the higher level and make the configuration process complex. In other words, with this approach, we expose all the internal variability of the lower family levels to the higher levels even when this does not create value. (e.g.: expose the internal variability of the seat with its options motor and heating when, at the vehicle level we want to use only a couple of criterion such as level of equipment and country/geography of sale to drive the internal configuration of the seat). This approach may be impracticable.

  Approach 3: Define the selected options/variants of the lower level family by rules enabling to convert variability effectivity of the higher level family to variability effectivity of the lower level family. This approach enables us to simplify the number of criterions used at the higher level, while enabling the use of a larger number of options/variants at the lower level. The complexity of the work is to define the mapping rules and to be sure that this mapping enables the effective respect of the two basic rules ("no holes", "no bump") in any resolved configuration. It is summarized in Figure 5 hereunder.

  Figure 5:Multi-level families with variability effectivity mapping



3.3 Variability management in engineering – The question of 3D configurable Digital Mockups

This is a complete topic which needs to be developed more. We will only summarize the main outcomes of our experience here without explaining them in detail.

The principle main goal of the 3D Digital Mockup (DMU) is to integrate all the 3D definitions of the different components in a common model making a complex product. In this way, it sustains concurrent engineering and allows 3D Design in context.

The first concern of variability management in DMU is that each engineering team needs to design the whole variability corresponding to the product items and potential families it has in charge. But reversely, it must take care only of the surrounding variability which could affect it. The approach of 3D Design in Context based on DMU pushes some companies to fully configure these mockups with variability and to do it at part level. Even when achieved, the observed fact is that it is very difficult for engineering teams to select among the multitude of surrounding variability combinations those which are the most constraining for their design. That is why I recommend to model physical architecture of the product in the DMU. For one product family, we may have variability in spatial architecture, but it would be considerably reduced compared to the whole variability of parts. Thus, we will privilege 3D Design in context with a context specified by architecture models greatly simplifying the variability selection of the context.

Another important difficulty observed for configurable DMUs is the management the variability of positions of assemblies and parts. For the supply chain, variability management at the BOM level does not need to consider the positions. But DMU has to do it. It is extremely important to use an architecture relative positioning in DMU to minimize the variability. Otherwise, absolute positioning will introduce additional variability even where sub-assemblies are strictly identical, just because they have different absolute positions.

So we recommend modeling spatial architecture of the product with the required variability and based on the following types of models:

- Reference geometry models. They specify dimensions and provide the architecture of positioning through the set of triaxial geometric frame of reference needed to support relative positioning of lower level assemblies.
- Geometric Interface Models. They specify the geometric interface between physical assemblies.
- Space allocation geometry to define the overall space allocation for each of the physical assemblies.

4 System Engineering and variability management

4.1 Impact of systems massive intrusion in traditional products

Systems and software are invading traditional hard products to make them smarter and to allow them to operate as pieces of larger systems. This trend increases the complexity of products. For example, the automotive industry anticipates now an order 10 million of lines of code for the embedded systems of one car. Management of variability needs to cover this systemic dimension. Said in another way, in engineering, the variability of industrial products cannot be managed only under the angle of the physical products structures anymore, as it is often done, but needs to address the system variability (including functional, and behavior). Moreover, these two dimensions of variability need to be managed consistently in configuration (including the management of changes across times).

The PLM/PDM systems role is mainly focused on sustaining the engineering activities for the definition of product families carrying internal variability (design, development and change), as well as the definition of the technical processes of their life-cycle and the definition of the technical resources involved by these processes.

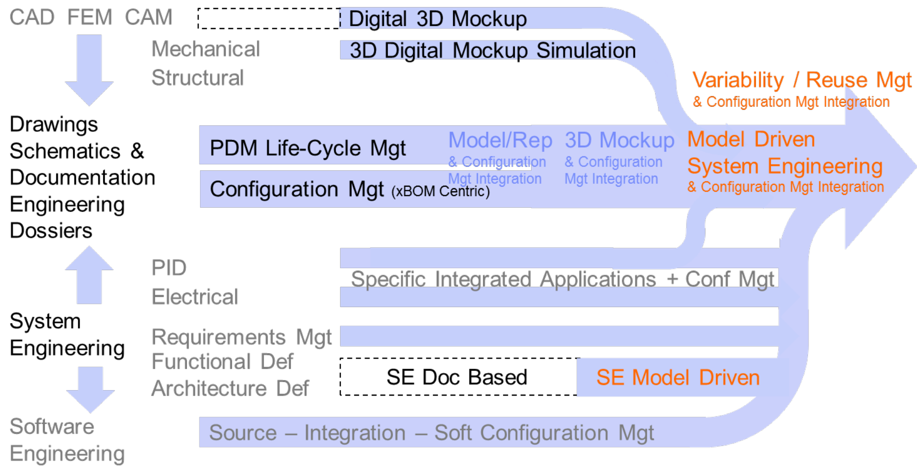
Figure 5 hereunder summarizes the different components of the PLM. It illustrates that the full coverage of all engineering activities relies on different sources of applications which were progressively integrated around PDM systems. It illustrates also that System and Software Engineering are still often being managed independently and reflects a need and a trend to make them converge under a consistent and integrated configuration and variability management

Until now, the PLM/PDM focus was mostly dedicated to managing the technical dossiers (definition, manufacturing and maintenance) and the DMU. This explains that the focus of configuration and variability management in PLM/PDM was to manage the physical product structure and parts configuration.

But there are two strong trends in the PLM/PDM landscape:

- The transformation of system engineering with the development of Model Based System Engineering (MBSE) and a better integration of software engineering.
- The very fast and strong intrusion of systems into traditional products which push PLM/PDM editors to better address and integrate system engineering under the PLM/PDM umbrella.

Figure 5: PLM progressively integrates all engineering specialized applications under the PDM umbrella for configuration, models and documentation management



Sections 4 and 5 discuss the perspectives for using the System Engineering approach for improving reuse and variability management and overcoming current limits observed in current PDM practices to address them.

4.2 Concepts and lessons from Software Variability Management

Variability management has been largely studied in numerous research papers. There are differences between variability management of software and physical products. Software is largely immaterial, easily produced and can be easily changed while physical products often require expensive manufacturing facilities and tools and changes are difficult and costly or even impossible on the already manufactured products. But for many aspects, variability management faces the same challenge in both kind of industries. Moreover, due to the very important intrusion of electronic and software in traditional industry, variability in both domains needs to be managed consistently as the "mechatronics" nature of present products induces dependencies between them. The [Chen, Babar & Ali 2009] paper reviews research studies in variability management and software products lines management (SPL) which is an equivalent concept of products families. [Capilla, Bosch & Kang 2013] made a systematic review of the main concepts and principles used for managing variability. When looking at these papers and others cited in the bibliography, we may notice several interesting concepts, questions and approaches to solve them.

Feature modeling is a key concept to specify and model SPL. SPL is defined from the variability point of view by a feature tree. It structures the configuration model of SPL with two main types of nodes: variability points and, under them, options-variants nodes. Variability points means there is a variability choice to make at this level to configure the software product. The choice must be made by selecting one of the options-variants nodes proposed as child nodes. Option means that the node can be selected or not. Variant means that one of the variant nodes must be selected. [Kang, Cohen, Hess, Novak & Peterson 1990] proposes a method for identifying and specifying features (Feature-Oriented Domain Analysis FODA method). These basic principles are enriched with more possibility of specification of the cardinality of the choices and the possibility to add attributes and constraints in the model [Capilla, Bosch & Kang 2013]

But numerous research papers point out some questions and difficulties and approaches to respond to them. We list hereunder these points and how they are linked to our observations and industry experience in hard products domain.

1. Definition and number of features

   The way to define and choose the features to build the configuration model may be difficult because variability may be seen from different points of view and the number of features to support them may become important and complex to manage. [Capilla, Bosch & Kang 2013] said that features are used by a feature based approach as container of:

   - Capability that is delivered to a customer
   - Requirements containers i.e., units of requirement specifications
   - Product configuration and configuration management
   - Development and delivery to customers
   - Parameterization of reusable assets
   - Product management for different segments

   Difficulties to define variability criteria and values often encountered in the hard products industry would benefit from an approach based on requirements and feature modeling.

2. Multiple point of views for variability:

   One of the reason for complexity of features modeling is the fact that the variability modeling must endorse different points of view. [Chen, Babar & Ali 2009] underlines first a distinction between external variability (as seen externally by the customer) and the internal variability or technical variability. It also shows that requirements are progressively defined and refined from the initial architecture definition stage to the running system across all the stages of software development.

   This is a current weakness of variability management in the hard product industry to essentially manage the physical product point of view and not the others.

3. Variability and product / systems life cycle definition artifacts:
The definition of a complex systems or products is made through artifacts organized in different hierarchical structures, often managed relatively separately. The variability model has to be declined on each of these structures to retrieve and compose them accordingly to the configuration selected. [Jiao & Tseng 1999] addresses this topic by proposing an integrated data model mixing the different views consistently (not specific to software engineering). [Asikainen, Soininen & Männistö 2003] studies and compares how applications used to model and manage software architecture may be also used to manage variability through product configuration. This is also comparable to the industry experience presented in section 3 where we see how PLM is used to manage the variability, but with a focus on physical product structure and DMU.

4. Multilevel variability – Multi-level product families
[Reiser 2009], who deeply analyzes SPL for automobile, suggests that variability must be designed at different levels. This fits with our observation and approaches developed in section 3.2 well (multi-level products families). This requirement seems to be fundamental if we want to reuse a variable module in different variable platforms and in different variable vehicles. [Reiser 2009] develops a concept of configuration link which seems close to the concept of variability-effectivity mapping that we describe in section 3.2 and whose mechanism is provided by some PLM/PDM software packages. This approach and principle should also enable "local" specification and to management of the variability by considering only those which are meaningful for the perimeter of the considered product family. Inside the low level product family reused, the variability effectivity definition is not constrained to be expressed by options/variants values of the higher level product families. Configuration links or equivalent variability-effectivity mapping rules are needed to select the proper lower level family configuration corresponding to the variability effectivity of the upper level. Another way to see it is that this multi-level approach may enable some decoupling of the specification of the variability of the upper level product family from that of the lower level reused product family. This make possible to hide internal complexity of the variability lower level product family from the upper one. The classic example for automotive is to decouple a commercial feature such as "level of equipment" (values: lux – comfort – economy) and "country" from the technical features used at a module level such as a seat. A technical feature at the seat level such as "heating seat", for example, could be linked through a configuration link/mapping rules to the equipment level "lux" in Southern Europe and "comfort" in Northern Europe.

5 Conclusions: Using the System Engineering approach to define and model Systems/Products Families variability

These are just embryonic and not yet proven ideas which came to me when confronting my experience and the research review I have made in SPL. These ideas are driven by the conviction that the system engineering and the traditional physical product engineering approaches need to be consistently integrated into a unified approach and model.

From the experiences seen and described here, we may derive some conclusions and intuitions for the future:

1. It is necessary to find a common approach for managing systems variability and product variability. Until now, the two approaches were traditionally managed separately. The hard products industries mainly focused their PDM and ERP systems on managing the physical product variability. Strong intrusion of systems in hard products industry and growing interdependency of functional and physical dimensions push for an integrated approach
2. Variability criteria (features) are strongly related to requirements. In other words, it seems us that a variability criterion value may be quite formally linked to a consistent set of requirements.
3. The system model (according to SYSML common standard) offers a way to simultaneously and consistently mix the requirements, the functional, the physical and the behavioral points of view. So, if we are able to model the variability through a consistent set of features (requirements regrouping in line with systems model components), we have a solution to the question of multiple points of view for variability.
4. System level requirements are defined at the beginning of the system design and refined and allocated to the system architecture components in parallel with the architecture design. So variability definition is naturally and strongly related to the system engineering approach. Adding a feature concept to SYSML model (functional, physical and behavioral) offers a perspective to model variability progressively with the system architecture development, and to manage consistently the different points of view provided by SYSML.
5. Variability must be multi-levelled and structured by an architectural approach. When defining a system level, it is only required to define or know the external specifications of the sub-systems it relies on, but it is not required to define them internally. This abstraction capability enables us to define the variability focusing the engineering effort for the relevant level of abstraction. The system engineering approach could strongly help to properly define the architecture of a complex system/product in a hierarchy of product families according reuse and variability strategy. Moreover, limiting the variability of the architecture itself by standardizing interfaces may enable us to fit in different module families without (or limited) side effects on neighbor modules it interfaces with.

There are still questions not addressed here, which would need to be studied, for example:

- How, with a systemic approach, would we model manufacturing resources and processes (with the linkage of process to the product and with the variability at all levels product, process and resources)?
- What does variability for behavior mean (simulation, test, validation)? What variability in behavior is driven by the system or product variability definition, and what variability is added by the methods and process for simulating, testing and validating?

Nevertheless, for the reasons exposed above, it looks to us that the system engineering approach and that system-based-modeling-engineering (SBME) relying on SYSML could be a strong foundation for supporting the definition of complex systems/products with their variability. The configuration-linking/variability-effectivity-mapping-rules needs to be articulated with the system/sub-system concept. In this perspective, classic product definition would be embedded into the system definition. This model would also need integration of proper positions management with the physical description of the system and with the system/sub-system architecture.

I would be interested in getting feed-back of researchers about these ideas and possible research done on this subject I may have missed.

References

[Asikainen, Soininen & Männistö 2003] Towards Managing Variability using Software Product Family Architecture Models and Product Configurators
Timo Asikainen, Timo Soininen, Tomi Männistö – Helsinki University of Technology – Software Business and Engineering Institute (2003 SoberIT).
http://www.soberit.hut.fi/pdmg/papers/ASIK03TOW.pdf

[Becker 2003] Towards a General Model of Variability in Product Families
Martin Becker -System Software Group, University of Kaiserslautern Kaiserslautern, Germany - mbecker@informatik.uni-kl.de

[Capilla, Bosch & Kang 2013] Systems and Software Variability Management - Concepts, Tools and Experiences -
Capilla, Rafael, Bosch, Jan, Kang, Kyo-Chul (Eds.). SPRINGER
http://www.springer.com/computer/swe/book/978-3-642-36582-9

[Chen, Babar & Ali 2009] Variability management in software product lines: a systematic review,
Proceeding- Lianping Chen (University of Limerick, Ireland) - Muhammad Ali Babar (University of Limerick, Ireland) - Nour Ali (University of Limerick, Ireland) - SPLC '09 Proceedings of the 13th International Software Product Line Conference - Pages 81-90 -Carnegie Mellon University Pittsburgh, PA, USA ©2009
http://dl.acm.org/citation.cfm?id=1753247

[Jiao, Simpson, Siddique 2007] Product family design and platform-based product development: a state-of-the-art review
Jianxin (Roger) Jiao - Timothy W. Simpson Zahed Siddique - Springer Science+Business Media, LLC 2007 - July 2007

[Jiao & Tseng 1999] An Information Modeling Framework for Product Families to Support Mass Customization Manufacturing
Jianxin Jiao, Mitchell M. Tseng)
Department of Industrial Engineering and Engineering Management, The Hong Kong University of Science and Technology, Kowloon, Hong Kong
CIRP Annals-Manufacturing Technology, 1999 – Elsevier

[Kang, Cohen, Hess, Novak, Peterson 1990] Feature-Oriented Domain Analysis (FODA) Feasibility Study
Kyo C. Kang, Sholom G. Cohen, James A. Hess, William E. Novak,
A. Spencer Peterson
Software Engineering Institute - Carnegie Mellon University - Pittsburgh, Pennsylvania 15213
November 1990
http://www.sei.cmu.edu/reports/90tr021.pdf

[Lim 1994] Effect of reuse on quality, productivity and economics
WC. Lim Hewlett Packard– IEEE Software September 1994

[Manet Hamm O.Brien 2011] Making the word work better – The Ideas that shaped a century and a company 'IBM)
Kevin Manet, Steve Hamm, Jeffrey M.O'Brien
IBM Press/ Pearson 2011

[Pil, Holweg 2004] Linking Product Variety to Order-Fulfillment Strategies
Frits K. Pil – University of Pittsburgh - Matthias Holweg - Judge Institute of Management, University of Cambridge –
Interfaces 2004 – Vol 34, N° 5 September-October pp 394-403

[Reiser 2009] Managing Complex Variability in Automotive Software Product Lines with Subscoping and Configuration Links
Mark-Oliver Reiser Fakultäat IV {Elektrotechnik und Informatik der Technischen Universitäat Berlin

[Renault Nissan 2013] Common Module Family (CMF): a new approach to engineering for the Renault-Nissan Alliance
http://media.renault.com/global/engb/alliance/Media/PressRelease.aspx?mediaid=49441

[Rokunuzzaman & Choudhury 2006] Economics of Software Reuse and Market Positioning for Customized Software Solutions
M. Rokunuzzaman+ and Kiriti Prasad Choudhury+*
*School of Engineering & Computer Science, Independent University, Bangladesh (IUB), Dhaka, Bangladesh and Beximco Pharmaceuticals Ltd, Dhaka, Bangladesh. zaman.rokon@yahoo.com, kpmoni@yahoo.com
Journal of Software, Vol 6, n°1, January 2006

[Simpson, Siddique & Jiao 2007]
Platform-Based Product Family Development
Timothy W. Simpson[1 -], Zahed Siddique[2,] and Jianxin (Roger) Jiao[3]
[1]Departments of Mechanical & Nuclear Engineering and Industrial & Manufacturing - Engineering, The Pennsylvania State University, University Park, PA 16802;
[2]School of Aerospace and Mechanical Engineering, University of Oklahoma, Norman, OK 73019;
[3]School of Mechanical and Aerospace Engineering, Nanyang Technological University,Singapore 639798 - Springer 2007

[Volkswagen 2011] Volkswagen Fact book 2011
http://www.volkswagenag.com/content/vwcorp/info_center/de/publications/2011/04/Volkswagen_Group - Factbook_2011.bin.html/binarystorageitem/file/Factbook+2011.pdf

[Zhang & Fan 2006] A Conceptual Framework for Product Lifecycle Modeling

Wenlei Zhang[1,2], Yushun Fan[3]
[1]Shenyang Institute of Automation, Chinese Academy of Science, Shenyang, P.R. China
[2]Gradute School of the Chinese Academy of Science, Beijing, P.R. China
zwl@sia.cn
[3]Dept. of Automation, Tsinghua University, Beijing, P.R. China
fanyus@tsinghua.edu.cn
Innovative Computing, Information and Control 2006 ICICIC '06 Vol 2

[Zhang & Fan 2007] A Conceptual Framework for Product Lifecycle Modeling

Wenlei Zhang[1,2], Yushun Fan[3]
[1] Shenyang Institute of Automation, Chinese Academy of Science, Shenyang, P.R. China
[2] Gradute School of the Chinese Academy of Science, Beijing, P.R. China - zwl@sia.cn
[3] Dept. of Automation, Tsinghua University, Beijing, P.R. China - fanyus@tsinghua.edu.cn
J Intell Manuf (2007) 18:5–29 - DOI 10.1007/s10845-007-0003-2