31st Annual INCOSE international symposium
Honolulu, HI, USA
July 17 - 22, 2021

A value-driven, integrated approach to Model-Based Product Line Engineering

Juan Navas
Thales Corporate Engineering
juan.navas@thalesgroup.com

Stéphane Bonnet
Thales Avionics Technical Directorate
stephane.bonnet@thalesgroup.com

Jean-Luc Voirin
Thales Airborne Systems
Thales Technical Directorate
jean-luc.voirin@fr.thalesgroup.com

Hugo Guillermo Chalé Gongora
Thales Corporate Engineering
hugo-guillermo.chalegongora@thalesgroup.com

Abstract. Pushed to the edge of their capabilities in a highly competitive world market, organizations everywhere look for efficient means to innovate and develop their products and services. This paper proposes and illustrates a holistic integration of Product Line Engineering (PLE) and Model-Based Systems Engineering (MBSE) to connect and align market and business analysis, architecting, design, and engineering. This value-driven, integrated approach capitalizes on MBSE best practices to tackle the concrete challenges of product line engineering.

Introduction

Organizations in every business sector are striving to achieve greater profitability and competitive advantage. Many organizations struggle to propose attractive and innovative product and service offers, while facing growing pressures to decrease costs and times to market. As product-services become smarter and highly interconnected through new technologies, their complexity continues to grow. Adding to this complexity is the inherent complexity of the business contexts, competitors and industrial landscapes of today's world market. Pushed to the edge of their capabilities, organizations everywhere look for efficient means to architect, design and produce their product-services.

Two of the most popular approaches organizations have turned to for the past several years are Product Line Engineering (PLE) and Model-Based Systems Engineering (MBSE). The reason for this is that virtually every organization performs some form of reuse (no system is created from scratch) and hardly manufactures one single instance of a product. Furthermore, several attempts to combine these two approaches into Model-Based Product Line Engineering (MBPLE) exist literature [Oster 2016][Chalé 2014][AFIS 2013][Young 2017][Li 2019]. Whilst most of these studies succeed in demonstrating that the combination of the two approaches can yield greater benefits than when applying them individually, practically none of them explores the problem in a holistic way nor tackles the challenges that arise when trying to implement MBPLE in real life, everyday practice in large organizations.

Applying PLE in an efficient way to meet strategic business objectives through the definition of the appropriate product line, while ensuring that all product instances will satisfy the needs of customers, raises indeed many questions related to organization and methodology. Examples of such questions include:

- What methods can be applied to architect, design and build the contents of a product line?
- How to align the architecture and design of a product to market and business analyses?
- How to verify the consistency of alternatives and options at different system levels?
- How to master variability and secure its consistency with architecture descriptions?
- Is a feature model enough to understand what each product option or alternative consists of, or should tacit knowledge be made explicit to make informed design choices?
- How to guarantee that each defined configuration results in a feasible and valid architecture?

Lessons learnt from years of product line management and deployment of architecture-centric, model-based engineering methods in our organization have lead the authors to consider that the answers to the above questions largely rely on an integrated product line approach that connects and aligns market and business analysis, architecting, design, engineering and manufacturing. This collaborative work between marketing, sales, architects, product managers and engineers can greatly benefit from model-based engineering techniques:

- To undertake the *architecting* of a future product line (i.e. to explore the problem and solution spaces in order to orient and frame a solution in an effective way);
- And to *design* the actual product line (i.e. to make the solution explicit).

This paper promotes a Model-Based Product Line Engineering (MBPLE) approach providing the foundational concepts and tools to perform this collaborative work. It proposes engineering practices to implement it in an industrial organization, and addresses a subset of the numerous challenges of this implementation.

The remaining of this paper is organized as follows. We start with a brief reminder of the main principles of Product Line Engineering (PLE) and Model-Based Systems Engineering (MBSE). Then we describe the overall MBPLE approach by presenting the major perspectives that define a solution, from the exploration and synthesis of the problem and solution spaces, to detailed needs analysis and solution design. For each perspective, we describe the associated methodology and illustrate it with a fictional case study.

Building blocks of the approach: PLE and MBSE

Product Line Engineering

Product Line Engineering is the engineering of a family of similar products (as opposed to the engineering of a single product) that exhibit variations in their characteristics. In its most advanced form, Feature-Based Product Line Engineering [INCOSE 2019] allows building a product line portfolio as a single production system rather than a multitude of individual products. This is achieved through the use and management of a shared set of configurable engineering assets that leverage the commonality within the family, through a systematic and rigorous management of variation amongst the products of the family and through automation mechanisms that allow generating asset instances adapted to a given customer need.

The variability of the product line is formalized by a feature model, which provides the common language to describe and share the variability domain of the product line across an organization. Features represent distinguishing characteristics that describe how the products within a product line differ from each other. Products are configured by selecting a set of features in the feature model.

The shared configurable assets are supersets of artifacts (requirements, models, documents, plans, etc.) that can be represented digitally and contain variation points, which are pieces of content that should be either transformed, included in or omitted from a product based on the features selected for that product. When creating a product instance, automation software applies the set of selected features to evaluate the variation points in the shared assets and creates instances of the transformed assets accordingly (Figure 1).

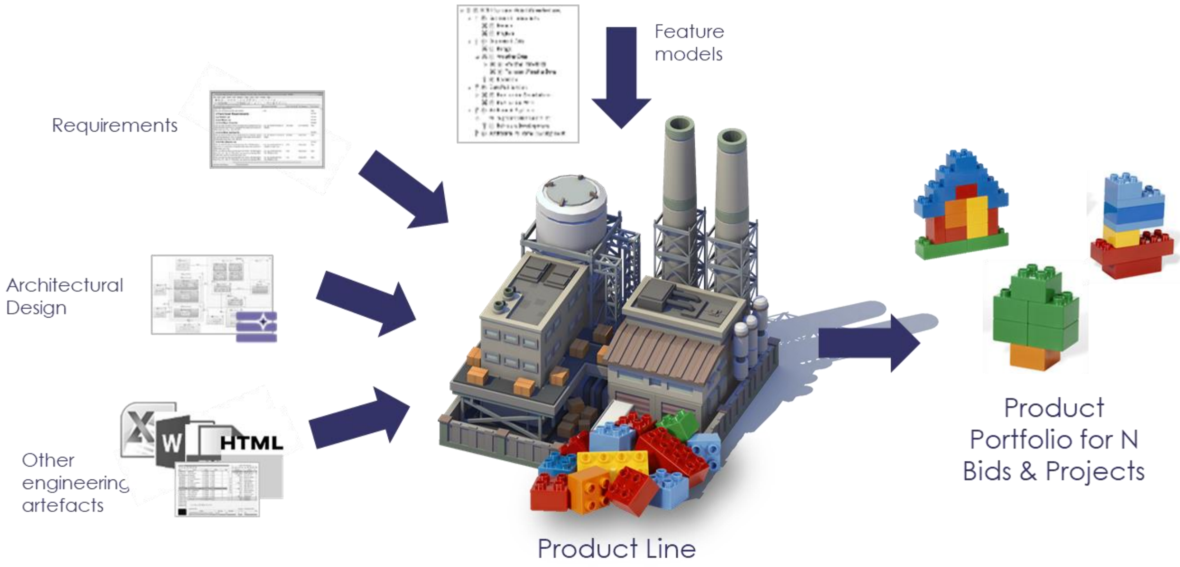


Figure 1: Product Line Engineering process

Once this “production facility” is implemented, it can deliver significant improvements in time to delivery, quality and cost of products. The main difficulties related to the implementation of PLE, however, are not related to the operation of this facility. The main issues relate to the definition of the path to properly setup the production facility (not to mention the necessary organizational and cultural changes that must be conducted, as explained in [Chalé 2014]):

- Aligning business strategy and product architecture;
- Producing the shareable assets and the feature model;
- And maintaining the consistency amongst all these elements.

These issues only becomes harder to tackle as the complexity of products and product lines increases. The approach presented in this paper provides a solution to these issues by capitalizing on MBSE practices.

Model-Based Systems Engineering

Model-based systems engineering is the formalized application of modelling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases. [INCOSE 2014]. In addition to increasing the level of rigor in these engineering activities, one essential objective of a model-centric approach is to provide a single source of truth that can be shared with all stakeholders.

In this paper we use the Arcadia model-based method devoted to systems, software and hardware architectural design [Voirin 2017]. It describes the detailed reasoning to understand and capture the needs of the stakeholder, define and share the product architecture across engineering teams, early validate its design, and justify it. The Arcadia method has been applied over the last ten years in a large variety of contexts, ranging from complex systems of systems to equipment, software or even hardware architecture definition. It is particularly relevant when strong constraints such as cost, performance, safety, security, reuse, consumption, or weigh have to be reconciled.

Arcadia intensively relies on functional analysis, which is a very popular technique among systems engineers. It also enforces an approach structured on different engineering perspectives that establish a clear separation between the need understanding (Operational Analysis and System Need Analysis) and the solution architectural design, in accordance with the [IEEE 1220] standard and covering parts of [ISO/IEC/IEEE 15288].

Each Arcadia perspective is described according to three dimensions:

- **Purpose**: why the system exists, described by operational capabilities and delivered system/technical capabilities (shown in orange in Figure 2).
- **Behavior**: what the system does. This mostly relies on functional analysis and modes/states. The behavior concepts can be used at different levels, either to describe the need –what the system does– or to describe solution –how the system works– (shown in green in Figure 2).
- **Structure**: what the system is. This covers the organization of the system in terms of constituents and interfaces. It also covers the humans and other systems in the environment of the system/product of interest (shown in blue in Figure 2).

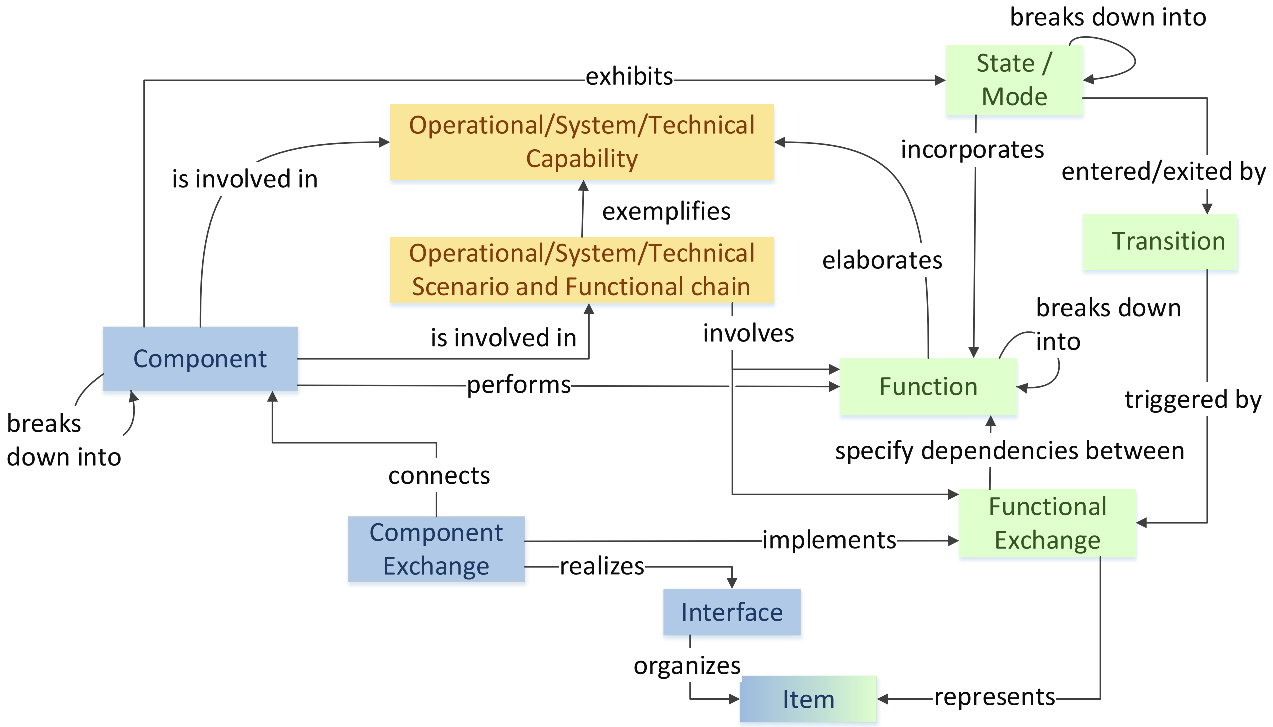


Figure 2. ARCADIA simplified ontology

As the lack of properly tailored tools has proven to be a major obstacle to the implementation of MBSE in industrial organizations [Bonnet 2015], Arcadia is recommended to be implemented using the open-source workbench Capella [Voirin 2015][Capella 2020]. Their combined use has proven well-suited for large-scale deployments of MBSE intended for engineers having diverse backgrounds [Bonnet 2020].

Overview of the Model-Based Product Line Engineering approach

The MBPLE approach presented in this paper consists of three *perspectives*, i.e. three ways of thinking about the product [ISO 42010]: Stakeholders Value, Product Value and Architectural Design. Figure 3 presents the perspectives and the systems engineering activities considered in each of them.

Our approach states a clear distinction between the problem and solution spaces. The problem space perspectives aim at achieving a proper understanding of the business and operational context of the product, the value the product can provide to stakeholders and the customer-oriented variability and configurations the product shall comprise. The solution-related perspective aims at defining the product architecture that will frame the stakeholders value and solve the problem expressed in the product value with the required degree of variability.

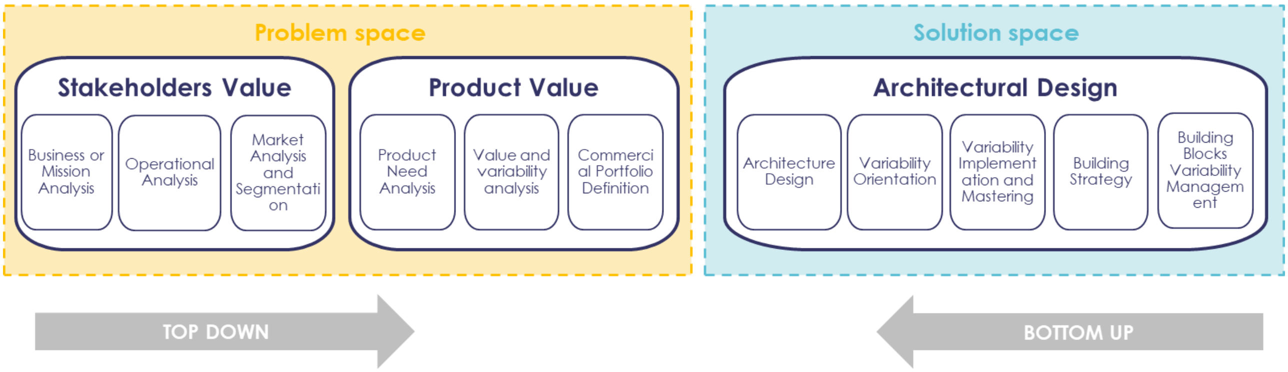


Figure 3. Overview of the MBPLE approach

The Arcadia perspectives presented before are tailored and embedded into these perspectives. Also note that this approach does not impose a strict order of execution of activities, although activities are illustrated in a top-down basis for clarity purposes in this paper.

Problem Space Analysis: Stakeholders Value Perspective

Activities

Business or Mission Analysis. This architecting activity initiates the systems engineering life cycle of a product line. This exploration typically defines the business or mission problem or opportunity, characterizes the solution space including the identification of measures of effectiveness, and determines potential solution classes to address the problem or take advantage of an opportunity. When applied to a product line, this activity yields a business strategy (business models, target markets and segmentation, product roadmaps, value propositions and differentiators), high-level operational needs and functionalities, and architectural concepts or orientations for the products of the family (for instance, a solution based on a drone system vs a solution based on a fleet of drones).

Many of these elements are high-level and conceptual, so out of the scope of the modelling concepts and perspectives of Arcadia. However, they feed Operational Analysis, as explained below, and may also drive and initialize further architecture perspectives, in a continuous and often iterative approach.

Operational Analysis. Operational Analysis formalizes the outputs coming from the Business and Mission Analysis. The stakeholders potentially involved in the different life-cycle stages of the product (e.g. use, operation and logistics) are identified, classified and characterized. They are captured as operational entities (organizations, systems, etc.) or operational actors (human users for example). The expectations of stakeholders are expressed in terms of operational capabilities, missions, scenarios, operational processes and related operational activities.

Market Analysis and Segmentation. In parallel, customer profiles are established collaboratively with marketing and sales teams to describe each customer segment. For each market segment and each relevant stakeholder, *jobs*, *pains* and *gains* are described: Jobs are what customers are trying to get done in their work and in their lives; Pains are anything that annoys the customers before, during and after trying to get a job done; Gains are the outcomes and benefits customers intends to achieve.

The concepts of the Operational Analysis constitute a great support: jobs naturally relate to operational activities while pains/gains can be considered as characterizations of these activities. The concurrent execution of Operational Analysis and Market Analysis favors thorough analyses and understanding of the activities and expectations of the stakeholders. Indeed, by characterizing the value for each market segment, we can formalize the value and criticality of the expected quality of service and of non-functional aspects for customers, as well as the possible combinations of operational activities and of quality of service commitments.

Throughout these analyses, stakeholder-level alternatives and options emerge. This variability is captured in a feature model. In an analog way, variability analysis may in turn shape the content and structuring of the operational capabilities and activities. Checks for global consistency of options and alternatives and user profiles must be performed here (e.g. seeking for activities needed to ensure a capability but not associated to the required feature).

Case study

The fictive Pythagoras company sells drone-based solutions. As the drone market grows exponentially, Pythagoras aims at capitalizing on its medium- and short-range drone platforms to sell both end-user solutions and simple services. Examples of targeted markets are monitoring of farms, inspection of aircraft exterior, surveillance of events, and inspection of remote structures such as catenary systems or electric power lines.

For each of these possible markets, the Pythagoras company needs to gain understanding of its future customers: their activities, their pains and the possible gains a drone-based solution could bring. An Operational Analysis is performed for each segment. In the case of farm monitoring, this analysis describes the various activities performed by a farmer, including the monitoring of crops health, the verification of the infrastructure, the surveillance of the livestock or the planning of interventions in the field. The identification of pains and gains can easily be associated to these activities. The added-value of a drone-based solution is confirmed:

- A drone would allow to analyze more accurately and extensively the health of the crops, and would help shorten the time between detection of issues and interventions.
- Offering automated analysis based on the collected data and proposing recommendations could be a differentiating factor.

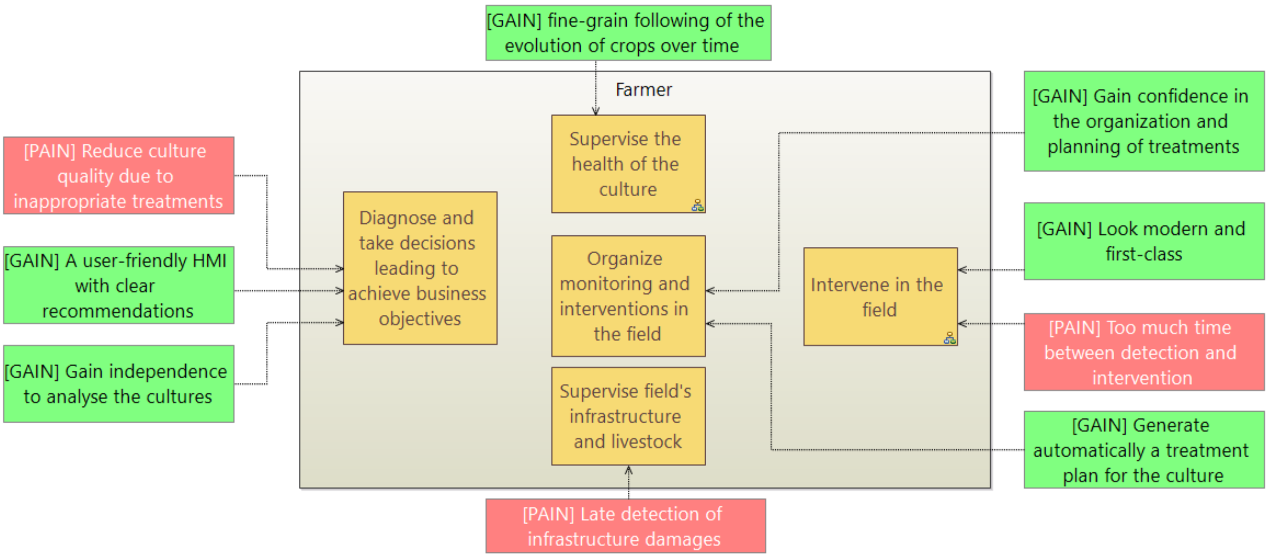


Figure 4: Pains and gains related to operational activities – Farms monitoring segment

Operational Analysis can also reveal unforeseen business opportunities. For instance, rather than selling a drone-based solution to a farmer, another option is to sell the solution to a service provider who would contract with the farmer, freeing the farmers from the hassle of installing and operating the drone. The Pythagoras company could even operate this service itself on behalf of the farmer.

At this stage of the analysis, the emerging variability can be captured in a dedicated feature model, as illustrated by Figure 5. The market segments (monitoring of farms, inspection of aircraft exterior, surveillance of events, inspection of remote infrastructures) are also captured in this feature model.

Figure 5. Emerging operational variability captured in a feature model (extract)

Problem Space Analysis: Product Value Perspective

Activities

Product Need Analysis. The objective is to identify and characterize the expected capabilities. Product capabilities rigorously capture the ability of the product to render services contributing to the realization of the operational capabilities of the previously identified market segments. Product capabilities are characterized by functional chains, scenarios and functions that not only describe how the product is expected to behave, but also help specify non-functional expectations (e.g. acceptable latencies). Operational Analysis and features of the market segments identified in the previous perspective constitute valuable inputs for the definition of product capabilities.

Value and variability analysis. In response to the pains and gains of the stakeholders, identified in the previous perspective, services, gain creators and pain relievers are identified. Services (typically captured as capabilities) describe what the product will offer and what will help stakeholders perform their jobs or reach their objectives. Gain creators emphasize how a given product capability will help users and customer be more efficient (in terms of time, quality or effort, for example). Pain Relievers will emphasize how certain product capability will contribute to help users and customer address the difficulties they face.

Value analysis strengthens the Product Need Analysis. It grounds the product definition on solid foundations and, when combined to the stakeholder profiles, it justifies the creation of a new set of product-level features that are captured in a refined version of the feature model. This integrated, model-supported approach benefits and simplifies variability analysis in at least three ways:

- The alignment between product capabilities and the structure of the feature model brings both organizational (structuring of engineering responsibilities and activities) and technical benefits (easier consistency checking, easier impact analysis).
- The modeling effort favors the identification of commonalities, with elements of the Product Need Analysis (capabilities, functional chains, scenarios, and functions) that are transverse to all segments and markets
- The dependencies between elements of the Product Need Analysis influence and even simplify features. For example, it is much simpler to define one single feature or option on a functional chain instead of on the numerous functions that are involved in this functional

chain. In addition, dependencies between features can be deduced from dependencies between elements of the Product Need Analysis (e.g. an optional function requiring inputs from another optional function).

Commercial Portfolio Definition. Based on the outcomes of these activities, the commercial offer portfolio can be defined by means of product standard configurations, which are selections of consistent feature options or alternatives. Standard configurations are key in order to simplify the definition of the solution for a given customer. They guide the customer choices according to market segments and towards proven product definitions, capabilities and assets, hence maximizing the reuse of assets. Standard configurations reduce the number of different product architectures to evaluate and validate both at definition time and at Integration Verification and Validation (IVV) time.

Consistency checks between Product Need Analysis, Value and Variability analysis and Commercial Portfolio Definition have to be performed. Consistency with the outcomes of the previous perspective must also be verified. Both checks are eased by the built-in semantic links between model elements.

Case study

While the Operational Analysis was performed for each market segment, Product Need Analysis intends to maximize the commonalities between all segments. The drone-based product has seven main capabilities: Fly, Acquire/Analyze/Visualize/Archive data, Sprinkle substance, Plan missions. Functional chains and scenarios provide specific implementation examples of these capabilities. For example, the "Visualize data" product capability covers the following functional chains:

- Display acquired HD video
- Display multi-spectral images
- Visualize all collected mission data
- Visualize substance levels

The degree of automation of operations and the timing of the data exploitation (live or post-navigation) are two structuring variability aspects of the product. Each high-level product capability can be refined in smaller capabilities directly linked to corresponding sections of the feature model (Figure 6).

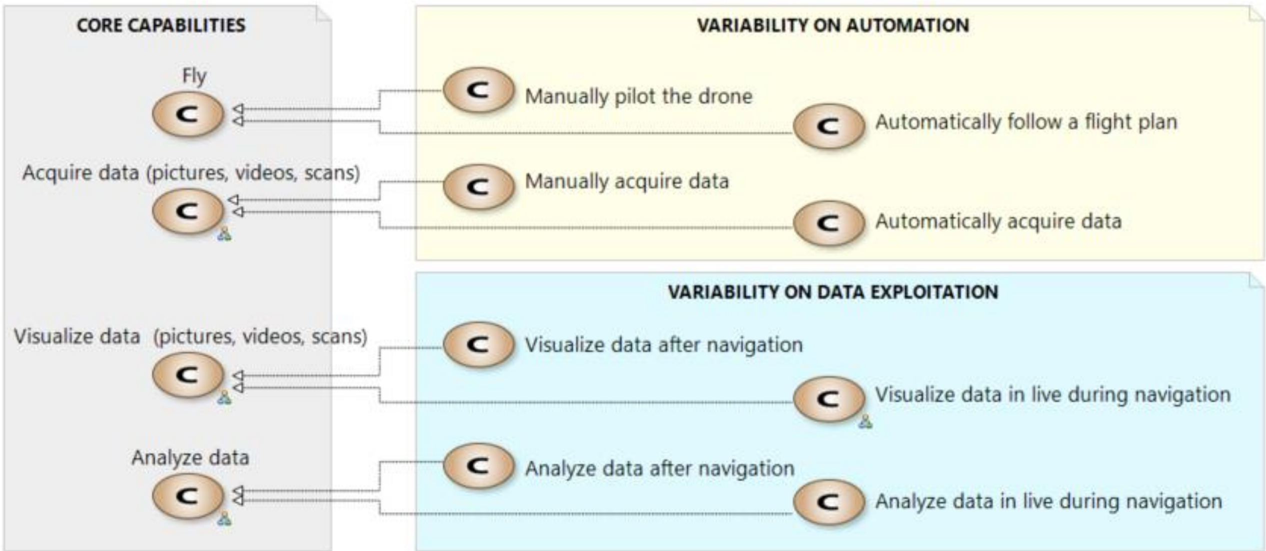
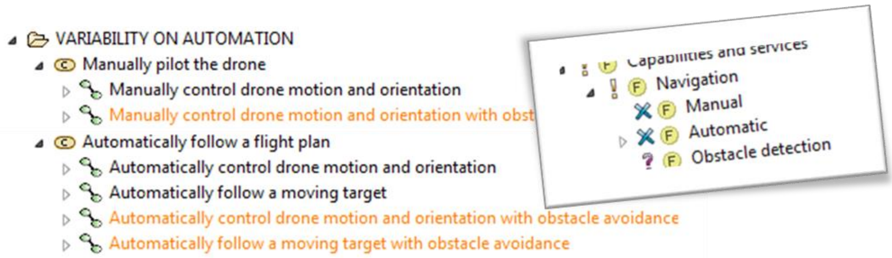
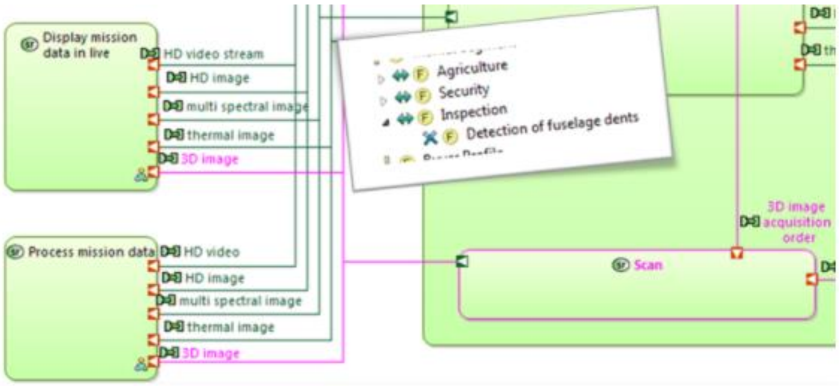


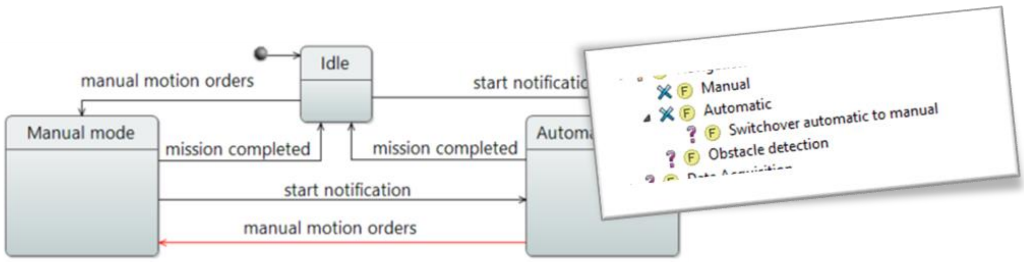
Figure 6: Product capabilities and their variabilities (extract)

Capabilities help structure engineering activities by providing powerful means to assign responsibilities and cadence the synchronization of all teams (architects, developers, V&V engineers) by defining the functional expectations of each development iteration [Navas 2020]. Therefore, capabilities constitute an excellent support to reflect the most structuring features captured in the feature model. It is however often necessary to capture variability at a finer grain. In these cases, other Arcadia concepts can be exploited, such as functional chains, scenarios, functions, functional exchanges, modes and states. The following table illustrates three of these cases.

<table>
<tr><th>Arcadia ontology concept</th><th>Example of variability</th></tr>
<tr><td>Functional chains and scenarios</td><td>Obstacle detection is an option that is available whatever the navigation mode. It is captured as specific functional chain in the capabilities dedicated to manual and automated piloting.


</td></tr>
<tr><td>Functions</td><td>The detection of fuselage dents on the exterior of an aircraft requires a specific kind of visual acquisition. This is captured in the model with a dedicated "scan" function.


</td></tr>
<tr><td>Modes and States Machines</td><td>The feature model proposes an option to automatically switch from automated to manual piloting mode as soon as the operator touches the joystick. In the model, this translates into a specific additional transition between modes.


</td></tr>
</table>

The result of the product needs analysis modeling is twofold:

- A feature model that is enriched with concrete alternatives and options, ready to be exploited by marketing (Figure 7).
- An architecture model that formalizes this variability and precisely describes what is expected from the product, in each of its forms.

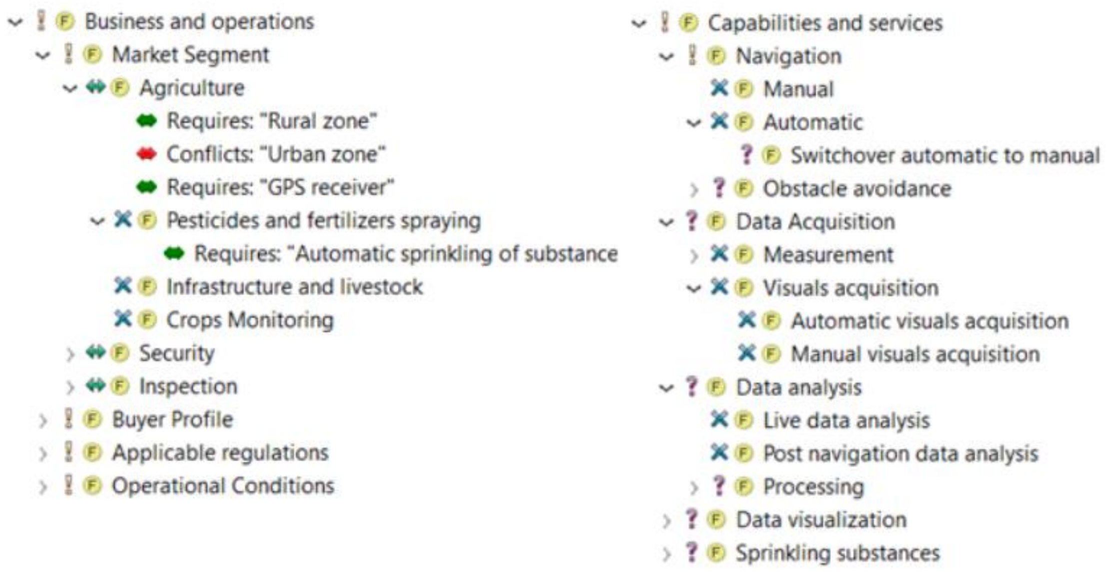


Figure 7: Feature model including capability and service-related features

Several standard configurations are established. For example:

Agriculture low cost	*Agriculture high end*	*Aircraft inspection*
Crops monitoring only, no automated flight	Crop monitoring and livestock/infrastructure monitoring, night and stormy conditions, automated flight and obstacle detection	Indoor, obstacle avoidance with minimal vertical distance

Solution Space Analysis: Architectural Design and Technical Concerns

Conceptual and finalized architectures address the design of the product architecture with different levels of abstraction, including behavioral aspects and structural decomposition into components, under technical and product line constraints.

Activities

Solution Architecture Design – conceptual and finalized architectures. The model-based design of the solution architecture follows the same patterns than the ones of Operational and Product Needs Analyses. Capabilities are exemplified with scenarios and functional chains that describe how the system works, thus specifying the exact contribution of each system constituent. The functional behavior of the solution must not only realize the functional analyses defined in Operational and Product Analyses, it must also reflect and implement their variability: for example, if a functional chain of the Product Need Analysis is associated to a feature, its corresponding functional chains in solution architecture must also be associated to it.

From a structural point of view, product variability may significantly constrain architecture design: for example, functions that have different variability conditions could be implemented by separate component. Similarly, alternative behaviors could preserve similar common Interfaces, and common core behaviors could be implemented by dedicated core components. Variability is also likely to emerge from limitations or opportunities in solution, technology or context, such as environment or regulations, leading to a solution-level feature model. Links between architecture and feature models are built accordingly.

Solution Variabilities Orientation. An analysis of the solution architecture of the product can simplify and reduce the number of features, notably due to architectural consistency or dependencies. For example, architecture constraints may lead to group different features that can be then considered as a whole: there is no need to treat them separately if the same components and functions are required for all of them.

Product architecture definitions and product standard configurations must be aligned. Product standard configurations are expected to cover the user needs in given market segments, but they also have to deal with solution-specific variability, architecture constraints and simplification opportunities. This may lead to revisiting product standard configurations during design. In the end, the standard configurations should notably:

- Satisfy a set of stakeholders belonging to a segment identified in the Stakeholders Value perspective
- Comply with the commercial offer portfolio as defined in Product Value perspective
- Be consistent with the designed architecture, for feasibility and efficiency reasons.

Here again, checking the consistency between architecture, feature model and configurations contents is key. This concerns notably:

- The consistency of components breakdown and of function-to-component allocation with variability
- The identification of dependencies or incompatibilities between features due to architectural concerns
- The validity of features and configurations in preserving components dependencies, scenarios and functional chains
- The respect of non-functional properties, quality of service, etc.

Variability Implementation and Mastering. Once architecture and feature models are consistent with each other, the features and configurations definitions should be applied to most engineering assets. This includes requirements, architecture definitions and models (done above), simulations models and scenarios, specialties and disciplines specific assets and models, test means and enabling systems, test campaigns, test cases, and also product breakdown structure, development and production means, support means and tooling.

Building Strategy. Standard product configurations should flow down to the building block level, so that building blocks can in turn be configured in a consistent way. The result of this collaborative work should be part of the development "contract" for every building block that contributes to the product line. Note that a component C1 may also have to adapt to product-level variability on other component C2 or C3 (e.g. if the component usually interacts with optional components absent from certain configurations). Integration, verification, validation procedures should be put in place to ensure the consistency of the composed architecture (i.e. of the complete configured product).

Building Blocks Variability Management. The product line architecture may rely on certain building blocks that can have their own variability and business strategy. Variability at product line and building block levels must be compatible and harmonized accordingly. This means that features related to building blocks are also captured in the feature model to guarantee the consistency of the overall model.

Case study

The conceptual and physical architectures in Arcadia capture how the product works, with different levels of detail. They cover both functional and structural aspects. Functions are refined until elementary functions can be entirely allocated to one product constituent. This allocation effort unambiguously specifies the expected contribution of each constituent.

While some of the features originating from the Product Need Analysis directly relate to the architecture model (e.g. the type of data acquisition directly dictates the choice of the type of camera), the feature model is likely to be enriched with solution-related features:

- New variability can emerge when designing the solution. For example, the need for a monitoring capability on the drone can be required in certain operating conditions.
- Design alternatives and trade-offs can be captured in the feature model.
- The choice of certain components or component versions (based on supply chain strategy, local regulations or optimization of stocks) might also be integrated in the feature model.

For example, depending on where the drone operates and on the kind of mission it performs, different sensing and navigation performance is required. If the drone is used in a hangar to inspect the fuselage of an aircraft, the navigation precision is critical. If the drone is used to monitor crop fields in a rural area, its endurance will most likely be more critical than the precision of its navigation system. The corresponding architecture alternatives can be captured as features in a dedicated branch of the feature model (Figure 8).

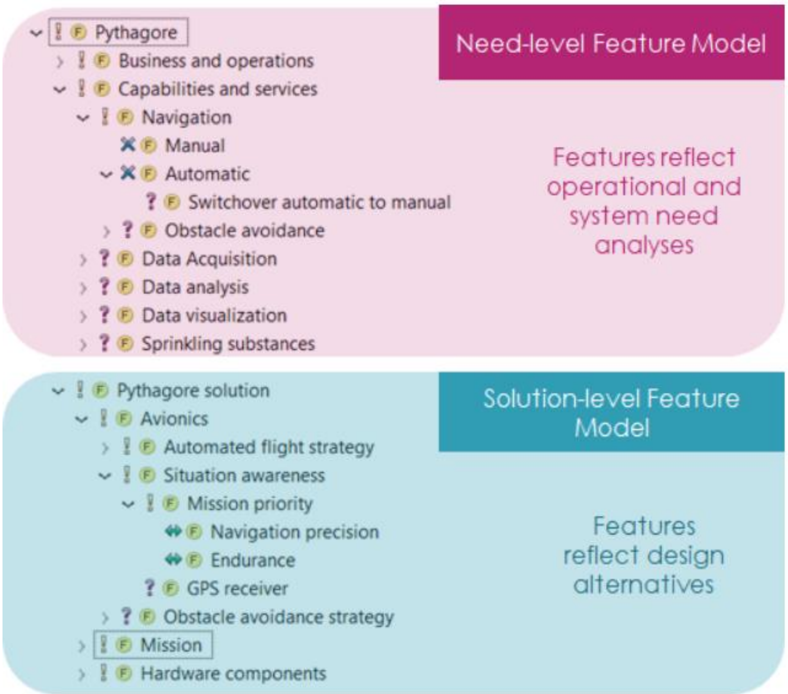


Figure 8: Enrichment of the feature model with architecture alternatives

The "Mission priority" feature has a direct consequence on the drone sensors, as illustrated in Figure 9 and Figure 10, where greyed elements are excluded from the current previewed variant.

Choosing "Endurance" means the duration of operation is critical. Of course, this option has a straightforward impact on the battery choice. But this option also means the global weight of the drone is a primary concern. In this setting, the amount of sensors is kept minimal and a compromise is made on the expected precision of the navigation, as a single inertial measurement unit is used.

When choosing "Precision" over "Endurance", dedicated sensors are added s the inertial measurement unit. This activates multiple sources on the computation functions, and generates the need for data correlation, which is expressed with textual requirements allocated to the functions in our example.

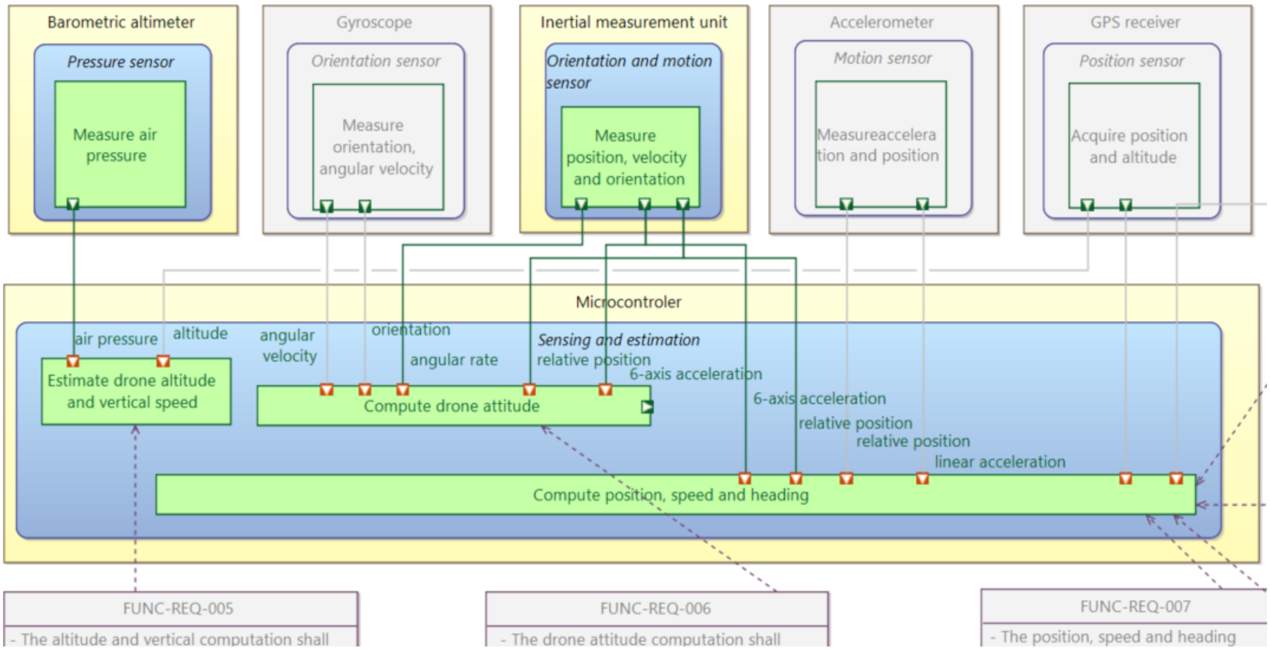


Figure 9: « Endurance » variant

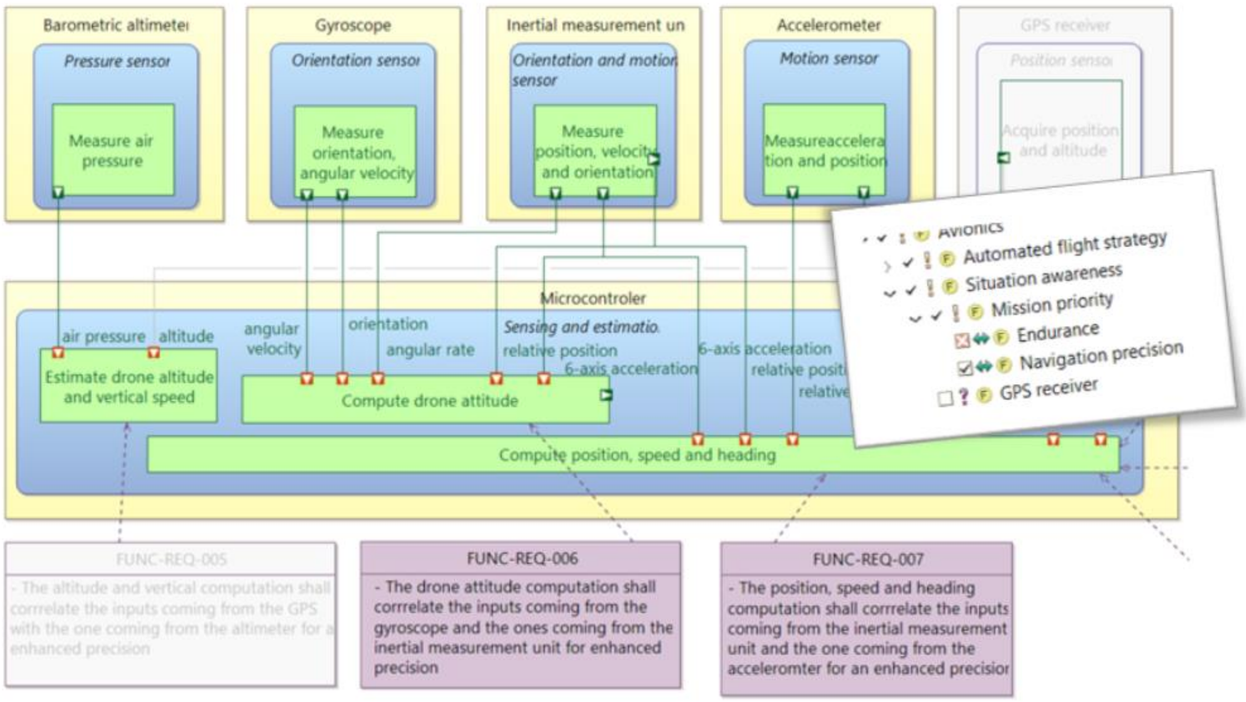


Figure 10: "Navigation Precision" variant

As illustrated by Figure 9 and Figure 10, each single feature has consequences not only on numerous connected model elements including interfaces, but also on textual requirements specifying the excepted behavior of the functions. Architecture models prove to be extremely useful to master the numerous and interdependent impacts of each choice. To master this complexity, we rely on a standard pattern to separate concerns: Features and individual models elements are not necessarily directly linked. Instead, the concept of variation point helps optimizing this mapping by connecting coherent sets of model elements to the feature model. This intermediate element helps reducing both the required amount of features and the feature mapping maintenance effort.

In its simplest form, a variation point simply refers to the selection of one given feature. But a variation point can also be activated by an expression containing several features and conditions. For example, avionics monitoring is not a feature. However choosing an urban operating zone in certain countries requires the implementation of an avionics monitoring function on the drone, in order to conform to local regulations. The monitoring variability point references a set of additional sensors and computers, and is activated when a certain combinations of “operating zone” and “country” features are selected.

Conclusion

The integrated approach described in this paper provides answers to the questions raised within several operational projects in Thales.

What methods can be applied build the contents of a product line? An integrated approach between market and business analysis, architecture definition and variability engineering, which can rely on methods that provide continuity and justification from users needs up to solution definition and IVV, such as Arcadia.

How to articulate market and business analyses with architecture and design? Operational Analysis is the place to start collaboration. A strong methodological focus on the articulation between need and solution secures consistency and applicability of product variability to product architecture and design.

Which verification process can be used for variability? Value analysis helps to understand the needs and profiles of stakeholders, as well as business opportunities. Formalizing this value analysis in need and solution architecture models not only secures a deep understanding of the product content and utilization, it also strengthens the variability definition by formalizing the scope and impact of each option or alternative.

How to master the complexity of variability, how to simplify and secure it, in coherence with architecture description? Architecture analysis simplifies some variability sources, helps define and reduce dependencies between variants, and secures adequacy between variability and architecture.

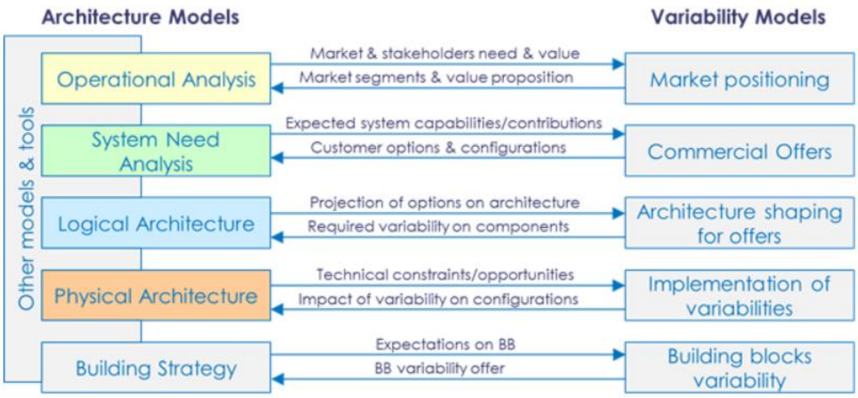


Figure 11: Summary of the mutual influences between architecture and feature models

Is reading the feature model enough to understand what each option consists in and to make informed choices? Usually not, except in very simple cases. Looking in the architecture model is necessary to understand what each option represents in terms of functional content, performance, global consequences (e.g. excess weight or price by adding options...), and make the link with the need (what consequences for the customer if they do not select it).

How to guarantee that each defined configuration results in a feasible and valid architecture? By deriving an architecture model per configuration, and analyzing it for consistency, completeness and adequacy to needs and constraints.

An effective and high-performance implementation of this integrated approach requires companies to implement changes in organization, product development processes, practices, and tools, which were not included in the scope of this paper. Among the most relevant ones:

- The organization structure of the company shall reflect the key role Product teams play in different functions: management, marketing and engineering to name some of them .
- Processes shall state and support the interactions between the company functions that are involved in Product development; architecture models are good candidates to support these interactions, as they provide a language that can be common to all the stakeholders and they reflect a large set of the key decisions related to product development.
- Regarding engineering, processes and tools shall be tailored so as they embrace MBPLE approach. Among those processes that are critical, configuration management and IVV execution definitely stand out. The integration of tools and, generally speaking, the implementation of Digital Engineering approaches are well-suited vehicles to ensure that processes are properly integrated.

References

Oster, C., Kaiser, M., Kruse, J., Wade, J. and Cloutier, R. 2016. “Applying Composable Architectures to the Design and Development of a Product Line of Complex Systems”, Systems Engineering Journal, Vol. 19, No. 6, Wiley: NY.

Chalé Góngora, H.G., Ferrogalini, M. and Moreau C., 2014. “How to boost PLE by using MBSE - A case study of a Rolling Stock product line”, Proceedings 5th Int. Conference on Complex Systems Design and Management CSDM (Paris, FR). Springer.

AFIS, 2013. *Systems Product Line Engineering*, Cépaduès Editions, Toulouse (FR).

Young, B. and Clements, P. 2017. “Model Based Engineering and Product Line Engineering: Combining Two Powerful Approaches at Raytheon”, INCOSE 2017 International Symposium (Adelaide, Australia). INCOSE: San Diego.

Li, M., Grigg, a., Dickerson, C., Guan, L and Ji, S. 2019. “A product line systems engineering process for variability identification and reduction”, IEEE Systems Journal, doi:10.1109/JSYST.2019.2897628. IEEE.

INCOSE, 2019. *Feature-based Systems and Software Product Line Engineering: A Primer*, Prepared by the Product Line Engineering International Working Group, INCOSE, San Diego (CA).

INCOSE, 2014, ‘A World in Motion – Systems Engineering Vision 2025’.

ISO/IEC/IEEE 15288, 2015, ‘Systems and software engineering – System life cycle processes’.

IEEE 1220, 2005, IEEE Standard for Application and Management of the Systems Engineering Process

Voirin J-L, 2017, ‘Model-based System and Architecture Engineering with the Arcadia Method’, ISTE Press, London & Elsevier, Oxford, 2017

Bonnet S, Voirin J-L, Normand V., Exertier D, 2015, ‘Implementing the MBSE Cultural Change: Organization, Coaching and Lessons Learned’, 25th INCOSE International Symposium

Voirin J-L, Bonnet S, Normand V., Exertier D, 2015, ‘From initial investigations up to large-scale rollout of an MBSE method and its supporting workbench: the Thales experience’, 25th Annual INCOSE International Symposium, 2015

Capella, 2019, Capella Polarsys Website: https://www.eclipse.org/capella/

Bonnet S, 2020. ‘Fostering MBSE in engineering culture’. INCOSE International Workshop 2020. https://www.youtube.com/watch?v=nxIB65aa4JE

ISO/IEC/IEEE 42010, 2011, ‘Systems and software engineering — Architecture description’

Navas, J, Bonnet S., 2019, 'How MBSE practices support the effective implementation of a Product Line Engineering approach', 29th Annual INCOSE International Symposium, 2019.
Navas J., Bonnet S, Voirin J-L, Journaux G., 2020, 'Models as enablers of agility in complex systems engineering', 30th Annual INCOSE International Symposium, 2020.

Biography

Juan Navas is a Systems Architect with 10-years' experience on performing and implementing Systems Engineering practices in multiple organizations. He is in charge of the Thales Corporate MBSE coaching team and dedicates most of his time to training and other consulting activities worldwide, for Thales and other organizations. He accompanies systems engineering managers and systems architects implement MBSE approaches on agile operational projects, helping them define their engineering schemes, objectives, and guidelines. He holds a PhD on embedded software engineering (Brest, France), a MSc Degree on control and computer science from MINES ParisTech (Paris, France) and Electronics and Electrical Engineering Degrees from Universidad de Los Andes (Bogota, Colombia).

Stéphane Bonnet is Systems Design Authority for the Avionics Global Business Unit of Thales. He oversees system architectures for the five business lines and leads the system-level R&T. He holds a PhD in software engineering and has led for numerous years the development of Capella, an open source modeling workbench for systems, hardware and software architectural design. He has been an active contributor to the Arcadia method. For several years, he has helped engineering managers and systems architects implement the MBSE cultural change, with a range of activities spanning from strategic engineering transformation planning to project-dedicated assistance to modeling objectives definition and monitoring.

Jean-Luc Voirin is Director, Engineering and Modeling, in Thales Defense Missions Systems business unit and Technical Directorate. He holds a MSc & Engineering Degree from ENST Bretagne, France. His fields of interests include architecture, computing and hardware design, algorithmic and software design on real-time image synthesis systems. He has been an architect of real-time and near real-time computing and mission systems on civil and mission aircraft and fighters. He is the principal author of the Arcadia method and an active contributor to the definition of methods and tools. He is involved in coaching activities across all Thales business units, in particular on flagship and critical projects.

Hugo Guillermo Chalé Gongora, PhD, CPRE, is is Product Line & Systems Engineering Director at Thales Corporate Engineering. He has over twenty years of experience in Systems Engineering and Product Line Engineering in the Energy, Infrastructure, Automotive, Railway and Aerospace & Defence sectors, where he has worked on the tailoring and application of SE, MBSE and PLE to the development of complex systems and products. Over the years, he has been particularly interested in safety-critical systems, formal methods, architecture description languages, autonomous systems, baseball and the Beatles. He holds a PhD on Energy & Thermal Systems, a Master on Energy Conversion and Internal Combustion Engines, and an Engineering Degree on Mechanical-Electrical Engineering. Guillermo is Co-chair and founder of the PLE International WG, former founder and chair of the Automotive WG and member of the Transportation WG and the MBSE Initiative of INCOSE.