Towards a Framework for Variability Management

Incose International Symposium, Rome 2012

SUMMARY

Product Line Engineering

- **Learning from experience and anticipating change**
  - **Innovation based on previous design**
  - **Generic & adaptable solutions**
  - **Preparing artifacts for reuse (modular architectures, product platforms)**

- **Product Line Variability** describes the variation (differences) between the systems that belong to a product line in terms of properties and qualities (like features that are provided or requirements that are fulfilled).

[Metzger et al. 2007]

01

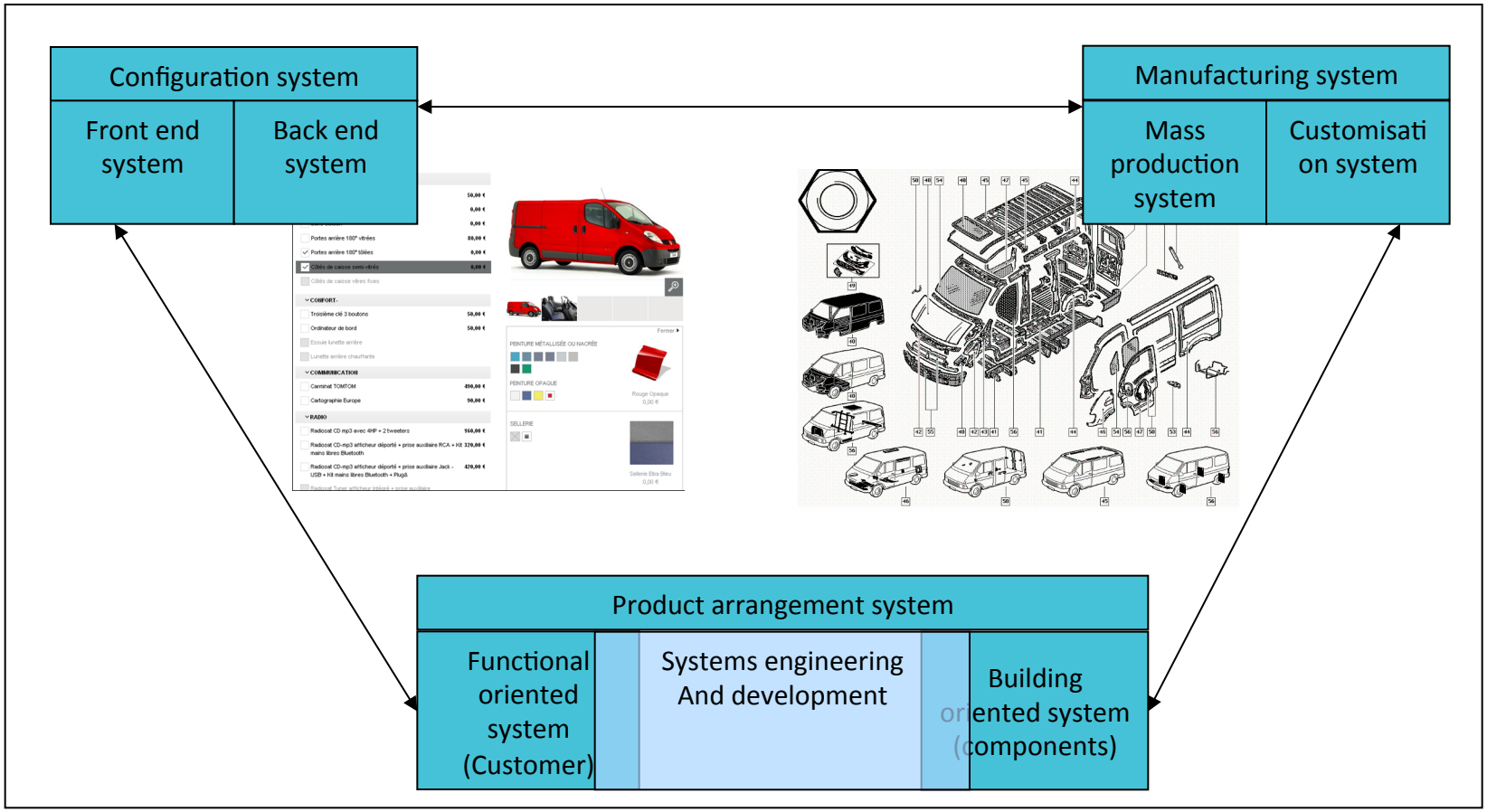
Introduction

Reuse in Automotive Systems Engineering

10^{21}

OVERVIEW OF DIVERSITY

- **Product diversity is a cross cutting concern in the organization, affecting all aspects of product design, development and manufacturing**

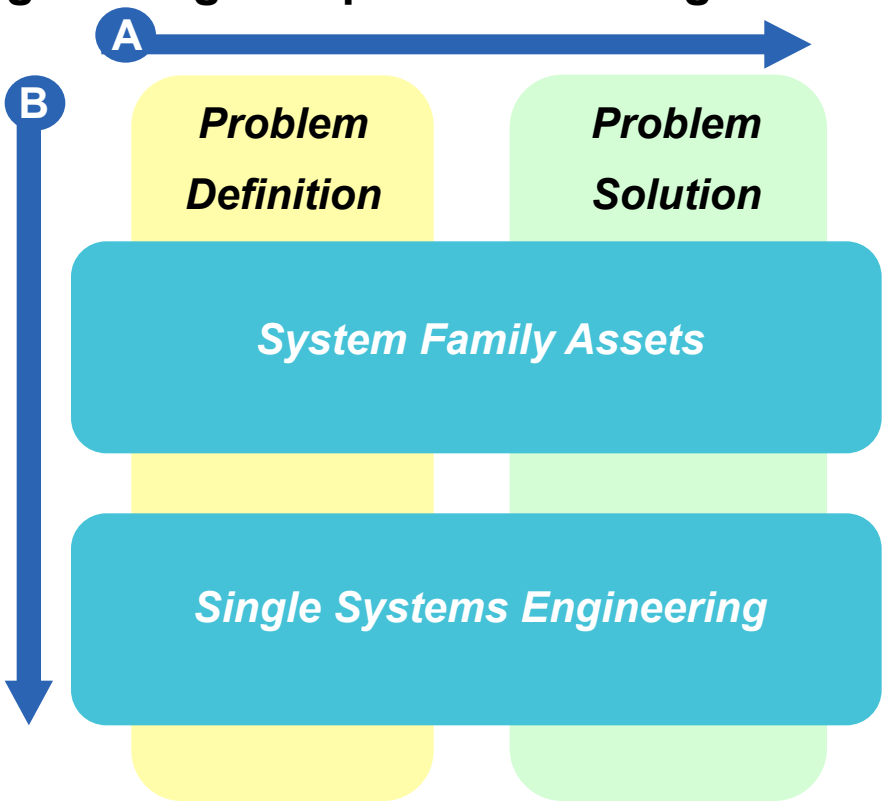


TOWARDS THE DEVELOPMENT OF FAMILIES OF SYSTEMS

- **Reuse (A) : Product Line Engineering two phase development supports reuse**
- **Problem Solving (B) : Systems Engineering as a problem solving oriented process**
- **Several problems arise such as:**
  - **Lack of methodological support to combine "product derivation" techniques" with general problem solving frameworks**
  - **Complex modeling & tool support that**

    Ensures data consistency for variability and system models

    Supports a custom configuration process (and guidance)

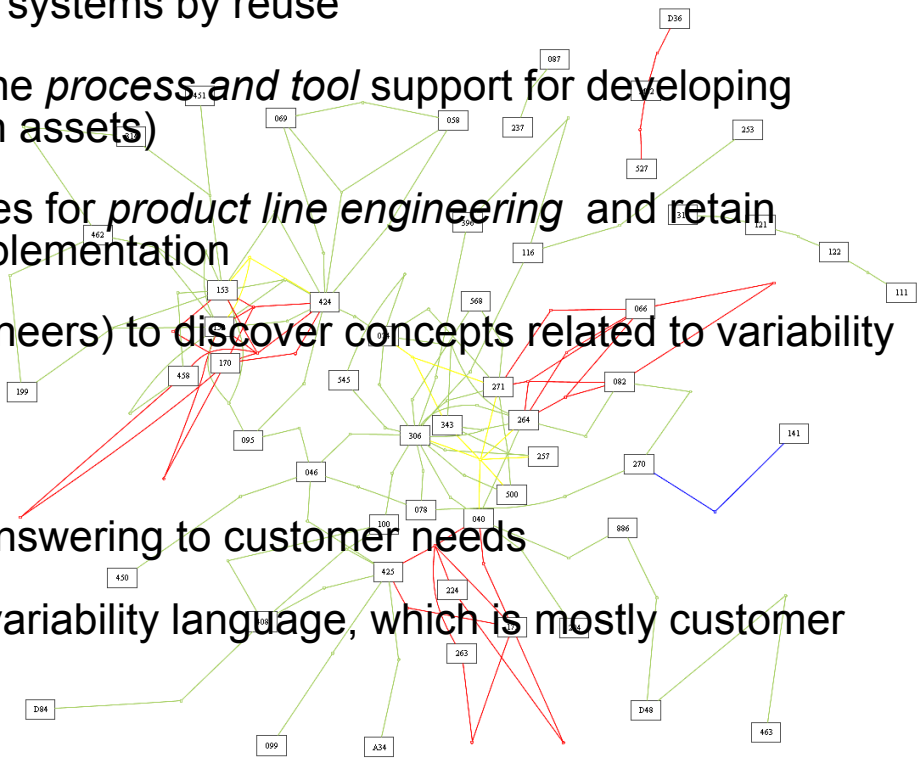


02

Scenarios for the Development of Families of Systems

RESEARCH PROTOCOL

- **Purpose**: develop an approach for managing variability for automotive systems in the context of model based systems engineering, keeping compatibility with current means of representing variability.
- **Approach:**
- Scenarios for developing automotive systems by reuse
- Define and derive requirements for the *process and tool* support for developing families of systems (sharing common assets)
- Draw on existing modelling techniques for *product line engineering* and retain appropriate concepts for our own implementation
- Interview professionals (system engineers) to discover concepts related to variability and needs.
- **Stakes**:
- Reduce the engineering cost while answering to customer needs
- And … Extend current organization variability language, which is mostly customer oriented (high level requirements)

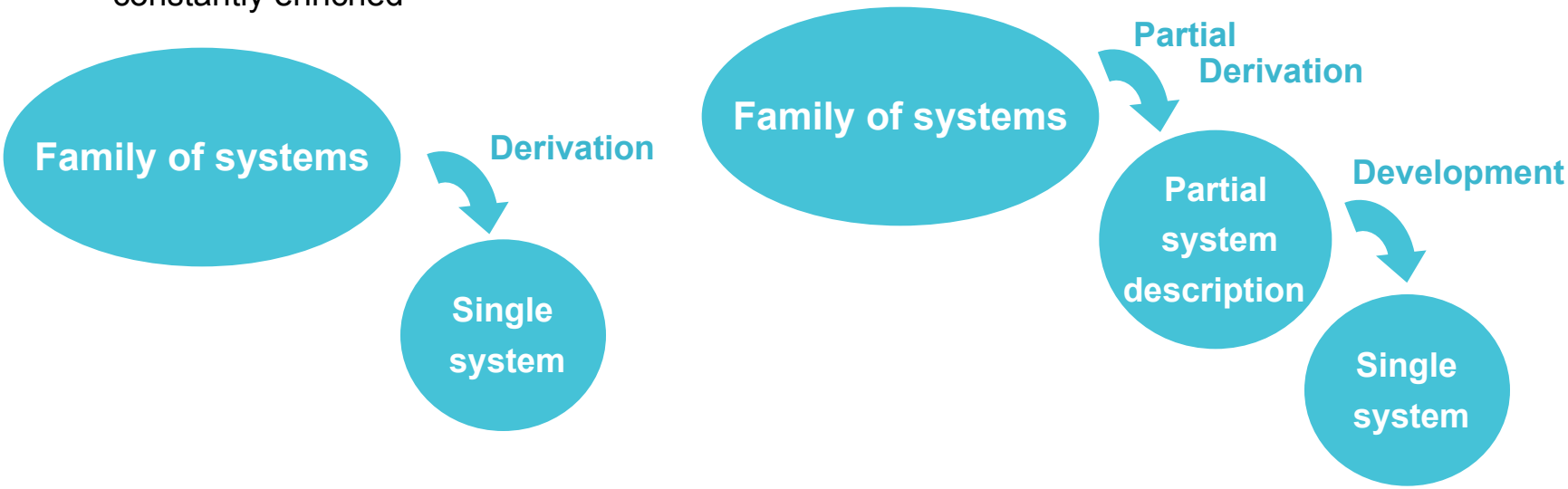


REQUIREMENTS FOR A « variability management system »

- **High Level Requirements for A framework for Variability Management**
  - **R1*[traceability]*: Enable propagation, traceability and coverage of stakeholders' requirements diversity to the solution**
  - **R2*[solution diversity]*: Enable successful realization of families of systems that cover different/variable stakeholder requirements through different variants of the realization**
  - **R3*[carry across]:* Enable reuse of development assets across projects for systems from the same family (ex: brake systems, lighting systems etc.)**
  - **R4*[reuse strategy]:* Support existing strategies of reuse by a more detailed documentation of variability, automation or assistance in the development activities.**

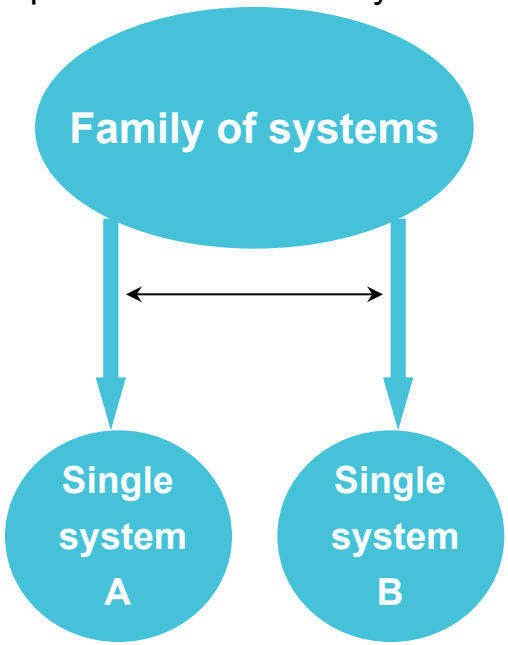
SCENARIOS FOR THE DEVELOPMENT OF FAMILIES OF SYSTEMS

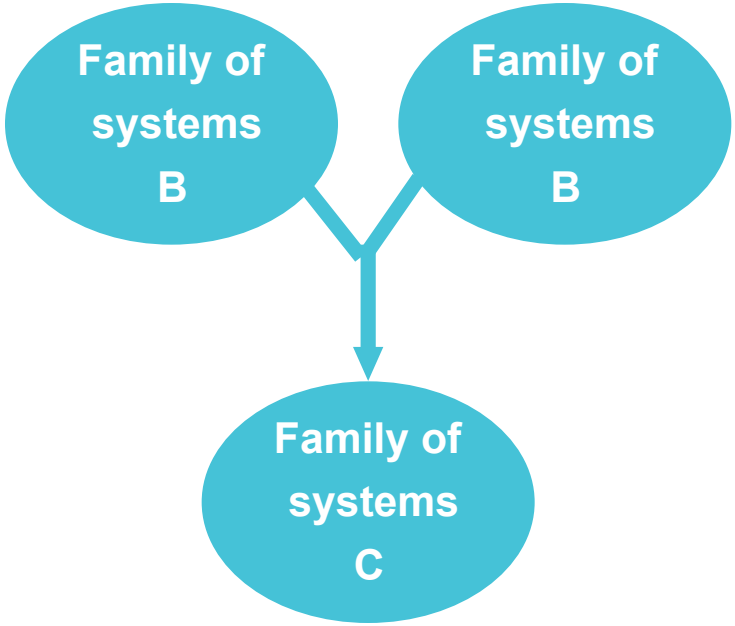
- 1. ***Deriving a single system from a system family "domain"-*** this is the nominal case where a single system can be derived from a collection of domain assets.
- 2. ***Develop a system from a partially derived system*** – the derived system does not represent a completely defined product. This scenario requires that system engineering activities continue to complete the system asset collection.
- 3.***Integrating of a single system into a product line*** – this is often the case of an innovation that enters mainstream development, a new system or option is proposed, but it is still possible to share system and lifecycle assets with existing system (e.g. improvement of a braking system by adding ABS, ESP etc.). When the innovation is based on or represents an improvement of a member of the actual product line, the scenario corresponds to an incremental development of a product line, where assets and options are constantly enriched



SCENARIOS FOR THE DEVELOPMENT OF FAMILIES OF SYSTEMS

- 1.***Synchronizing systems development*** - several systems within the same product line are developed in parallel, while all reuse related to commonalities between systems is anticipated and planned. The purpose is to maximize reuse decisions among a set of projects or contribute to the common assets.
- 2. ***Merging two product lines*** – this represents a middle case which can be useful when different systems are developed independently, initially, but common elements can be identified for future developments (e.g. development of similar products in different companies that eventually establish a partnership).





03

Modeling Approach for Variability in MBSE

Aspects of Variability

MODELING APPROACH FOR VARIABILITY IN MBSE

- **Three main concerns in capturing and representing families of systems**
  - **Variability and Commonality**

    Shared assets among several systems (multi product-systems with selection of basic elements*)

  - **Impact on the System Model**

    Basic forms of variation of the system : replacing (parts), existence/absence, configuration parameters

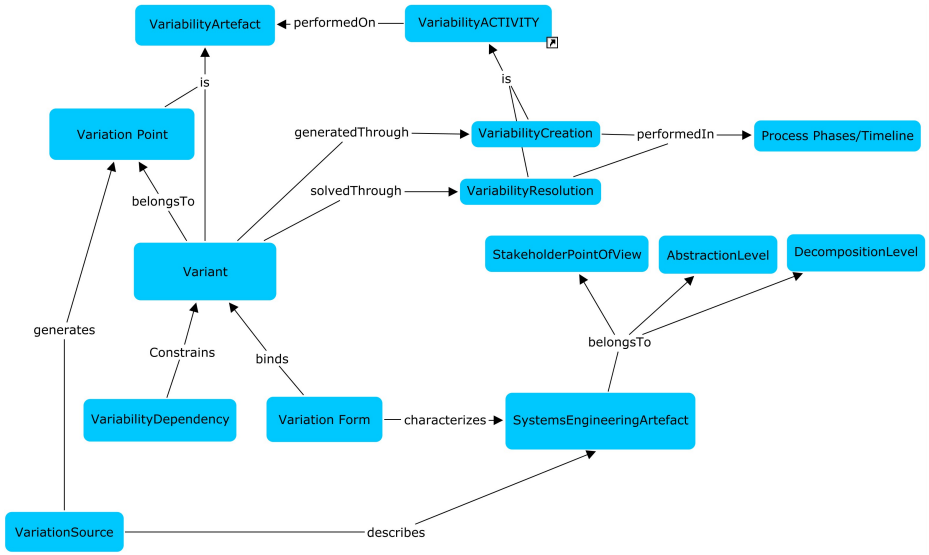
  - **Derivation information**

    Derivation *choices* as stakeholder (variable) requirements or design decisions

    Creation and resolution time of variations within the system lifecycle

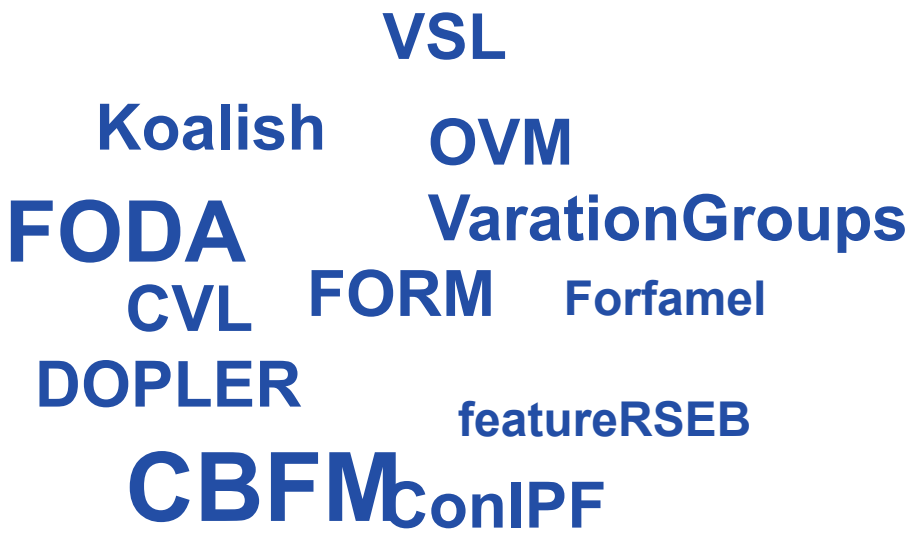
    Guidance through the design space from requirements to the detailed solution

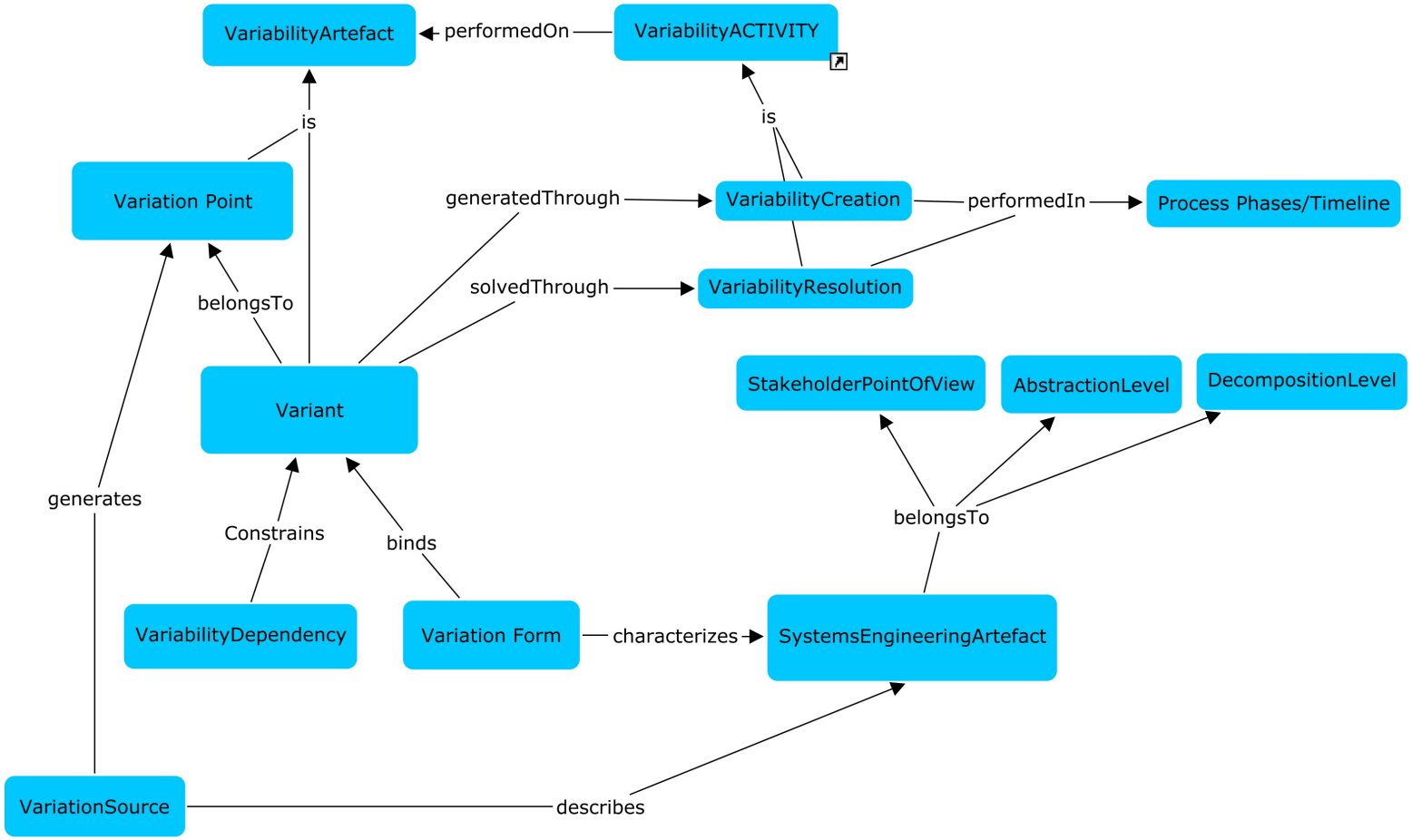
  - **Context vs. Internal System variability**



EXISTING CONCEPTS AND METHODS

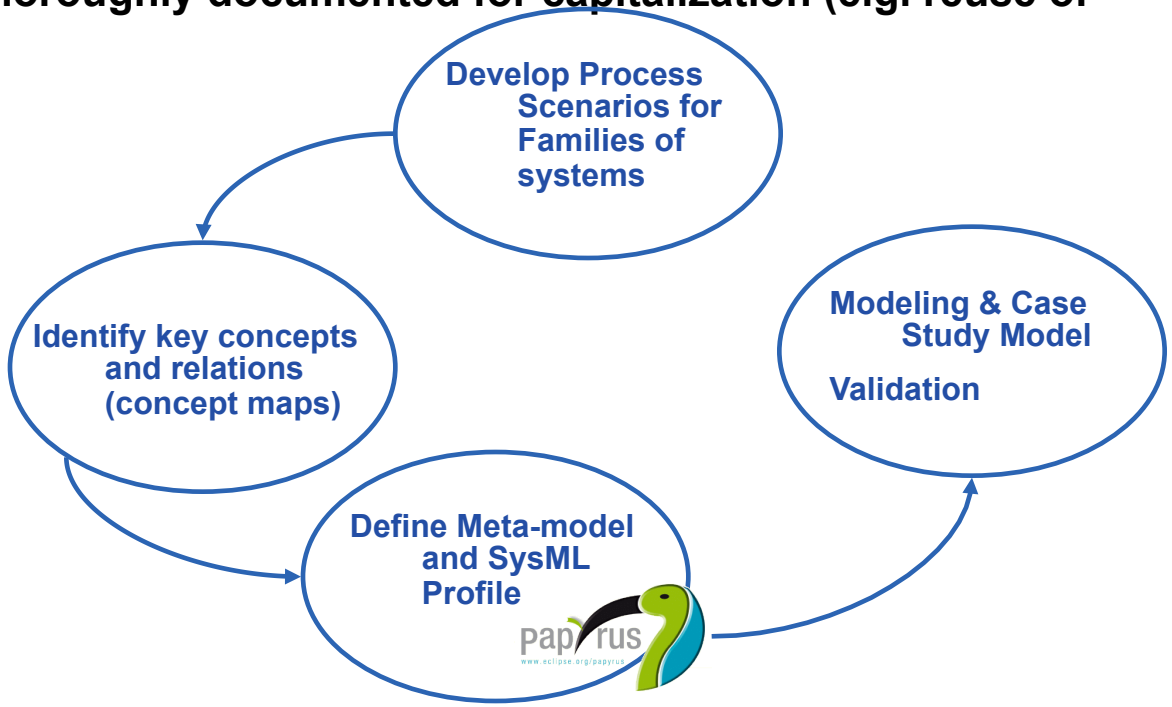
- **The existing concepts and methods for representing and managing variability**
  - **Specific context and needs, most often for software products**
  - **None covers all scenarios and requirements specific to our own context**
  - **Some contain concepts and notations redundant with the system model (like hierarchy, cardinalities)**
- **Compare existing approaches to our own requirements**
- **Draw on these approaches to extract useful concepts, consolidated in concept maps**




VariabilityArtefact
performedOn
VariabilityACTIVITY
is
is
Variation Point
generatedThrough
VariabilityCreation
performedIn
Process Phases/Timeline
belongsTo
solvedThrough
VariabilityResolution
Variant
StakeholderPointOfView
AbstractionLevel
DecompositionLevel
generates
Constrains
binds
belongsTo
VariabilityDependency
Variation Form
characterizes
SystemsEngineeringArtefact
VariationSource
describes

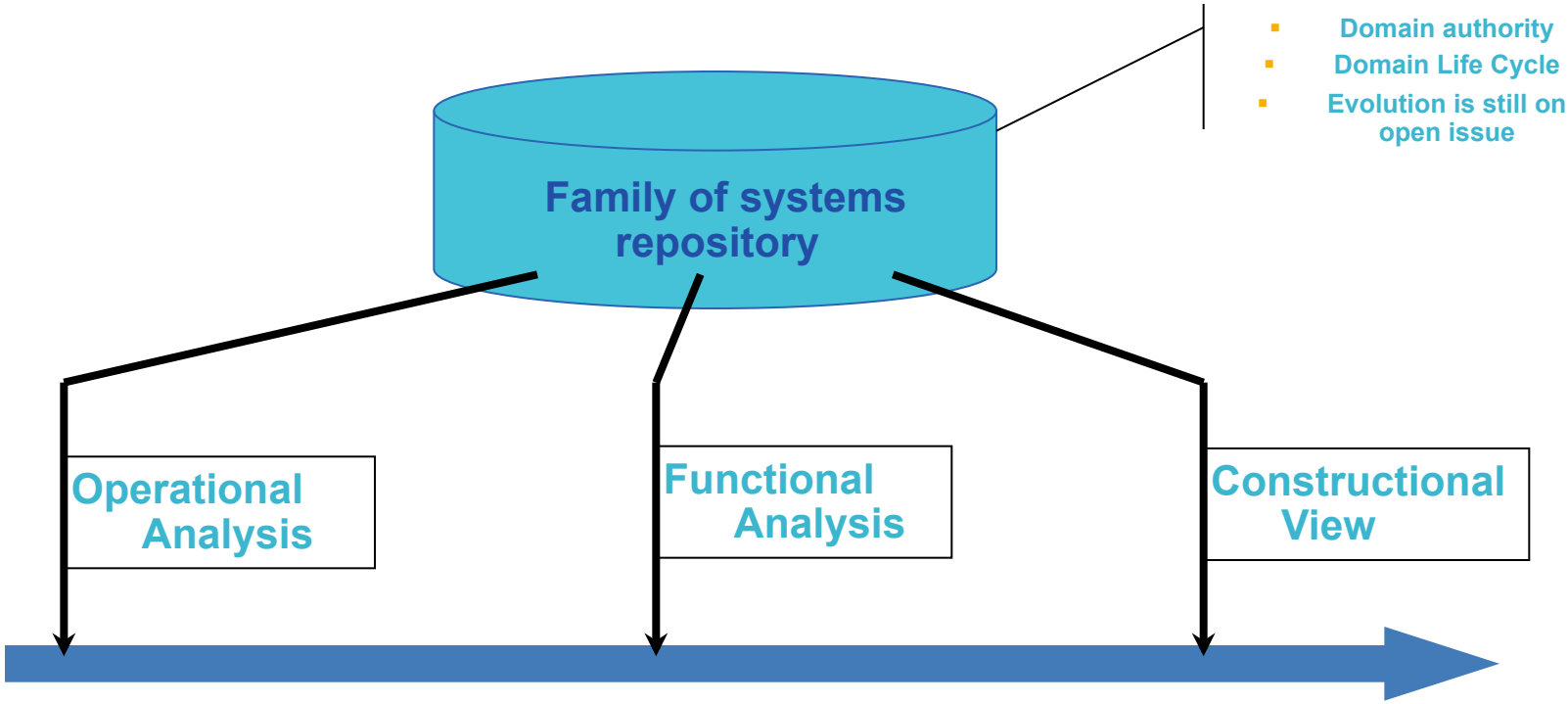
WORKFLOW

- Create A SysML profile based on the identified concepts for variability
- Concepts maps provide a mean for collaborative work
- Variation points represent stakeholder requirement alternatives as well as detailed design decisions
- Derived systems are thoroughly documented for capitalization (e.g. reuse of validation plans)



SYSTEM STRUCTURE

- **Derivation process: Between layers of abstraction or decomposition perform partial configurations**
- **Off the shelf SAT solver to support configuration**



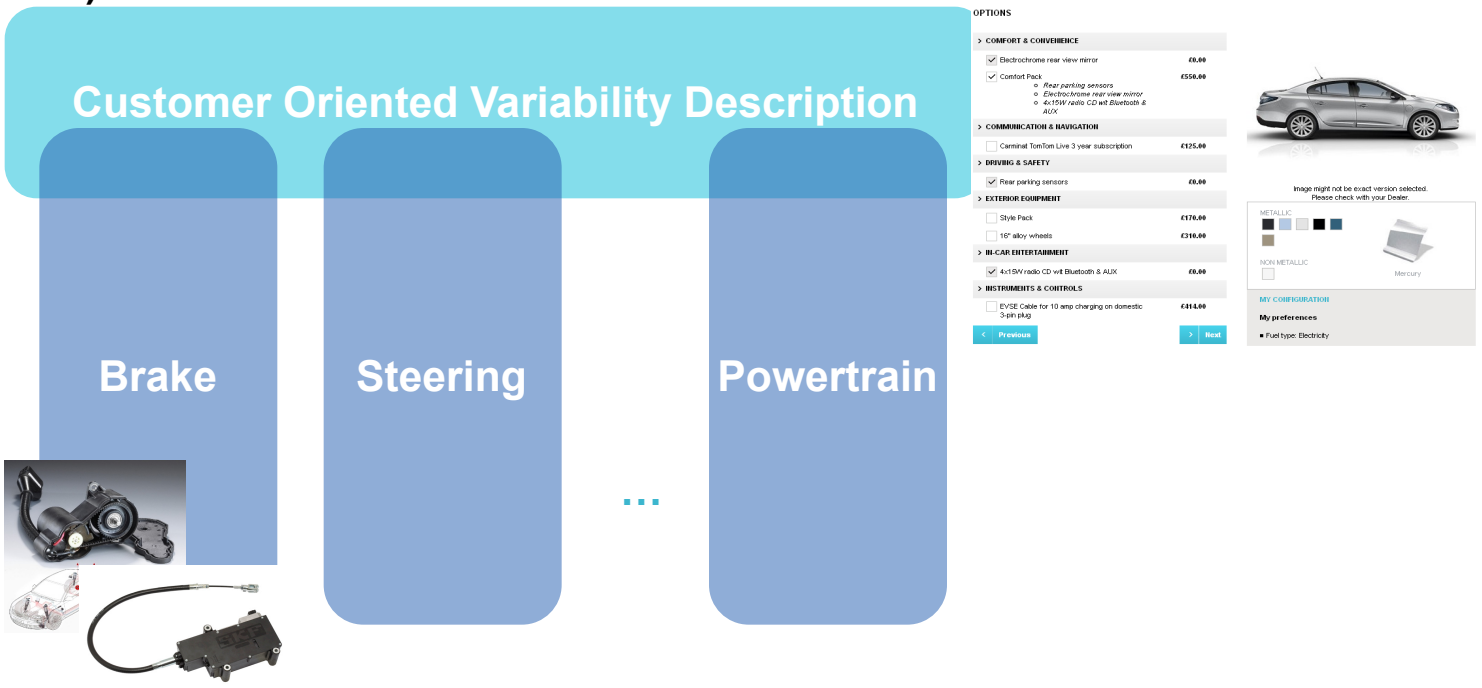
04

Perspectives

Towards integration with Legacy Enterprise Systems

PERSPECTIVES

- **Use model available information to compute variability related metrics (commonality, reuse of a component across the product line)**
- **Coupling with optimization to improve context base decision making ("reuse or develop alternative solution" decision)**
- **Problems related to product line evolution (family of systems lifecycle, discard information)**



Conclusions

- **Modeling may not be the complete solution but is important because…**
  - **A problem well understood is a problem half solved**

END

REFERENCES

- Clements, Paul, and Linda Northrop. Software Product Lines: Practices and Patterns. Addison-Wesley, 2002.
- Jones, Lawrence G., and Albert L. Soule. *Software Process Improvement and Product Line Practice: CMMI and the Framework for Software Product Line Practice*. Carnegie Mellon University, Software Engineering Institute, 2002.
- A. Metzger, P. Heymans, K. Pohl, P.Y. Schobbens, G. Saval: "Disambiguating the documentation of variability in software product lines: A separation of concerns, formalization and automated analysis", In: Sutcliffe, A., ed.: Proceedings 15th IEEE International Conference on Requirements Engineering (RE2007), 15-19 October 2007, New Delhi, India, IEEE Computer Society, 2007.
- Pohl, Klaus, Günter Böckle, and Frank van der Linden. *Software Product Line Engineering Foundations, Principles, and Techniques ; with 10 Tables*. Berlin [u.a.: Springer, 2005. http://dx.doi.org/10.1007/3-540-28901-1.
- Kang, K.C., V. Sugumaran, and S. Park. *Applied Software Product-Line Engineering*. Auerbach Publishers, Incorporated, 2009. http://books.google.fr/books?id=7XO_ghvkpt4C.
- Dhungana, Deepak, Paul Grünbacher, and Rick Rabiser. "The DOPLER Meta-tool for Decision-oriented Variability Modeling: a Multiple Case Study." *Automated Software Engineering* 18, no. 1 (November 17, 2010): 77–114.
- Tessier, P., S. Gérard, F. Terrier, and J.M. Geib. "Using Variation Propagation for Model-driven Management of a System Family." *Software Product Lines* (2005): 222–233.
- Blecker, Thorsten, Abdelkafi, Nizar, Kaluza, Bernd and Kreutler, Gerold University of Klagenfurt - Mass Customization vs. Complexity: A Gordian Knot?
- Astesana, J.M., L. Cosserat, and H. Fargier. "Constraint-based Modeling and Exploitation of a Vehicle Range at Renault's: Requirement Analysis and Complexity Study." In *Workshop on Configuration*, 33, 2010.
- Haberfellener Reinhard and Olivier de Weck 2005, 15th Annual International Symposium of the International Council of Systems Engineering, "Agile SYSTEMS ENGINEERING versus AGILE SYSTEMS engineering"