A Commonsense-Driven Architecture Framework – Part 1
A Car Manufacturer's (naïve) take on MBSE

Hugo G. Chalé Góngora, Thierry Gaudré, Alain Dauron

SUMMARY

01

Introduction

A Look at the Automotive Industry

The Automotive Industry: Usual Perception form the Outside

- **Historical point of view**
  - **Mass-production of products with increasing technological content**
  - **Offered to "non professional" users**
  - **Worldwide highly competitive market (costs, times to market, product attractiveness, Innovation)**
  - **A strong, passionate relationship between the customer and the product**

- **Systems point of view**
  - **A complex system of companies and organizations (OEMs, suppliers, regulation authorities, distribution networks and recycling networks)**
  - **Interacting in many different ways during all the phases of the vehicle lifecycle**

Evolution and Recent Histroy of the Automotive Industry

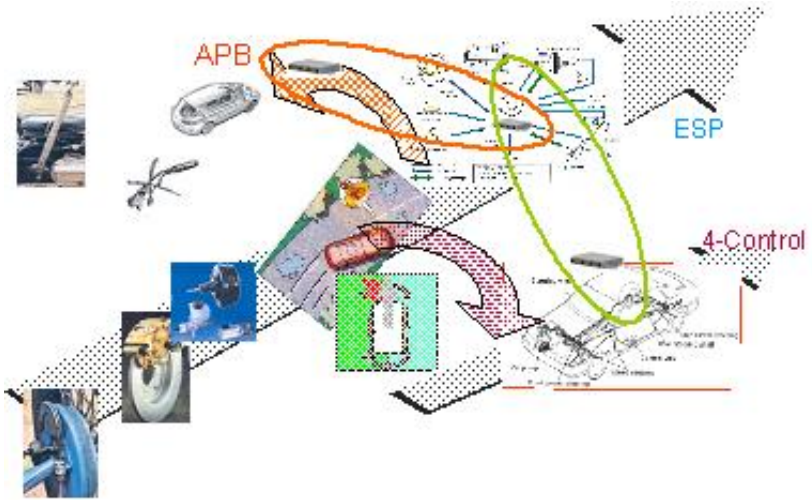
- **One Change-Inducing Factor: the market growth in emerging countries**
  - **Accentuates some of the characteristics previously described**
  - **Impose carmakers to propose attractive, "appropriate" products to this new rapidly-changing customers**

- **Three Complexity-Inducing factors**
  - **The spread of electronics into every vehicle function & the functional safety associated to them (a new ISO 26262 standard)**
  - **The constraint of drastically reducing emissions of CO_2 → electrification of the vehicle**

  - **A change of scope of the traditional automobile brought along by Electric Vehicles, advanced driver assistance systems and new mobility services**

Consequences of Complexity Inducing Factors: Spread of SW and Electronics

- **Heterogeneous involvement of carmakers in the development of SW and Electronics**
  - **Depends on the vehicle domain, goes anywhere from black box to white-box**
- **Organizational changes**
  - **From mechanical and manufacturing to the intricate organizations of today**
  - **Translates into development processes, probably not as flexible and agile as they could be**



- **… And Cultural changes**
  - **Modern vehicle systems calls for the participation of many different professional fields**
  - **Systems are mostly developed in a concurrent engineering approach, instead of a linear or cascade process**

Consequences of Complexity Inducing Factors: Functional Safety

- **The arrival of ISO 26262 standard: functional safety of road vehicles**
  - **A common standard built by the automotive community, regarding the functional safety of electrical electronic (EE) embedded systems**
  - **Describes activities and work-products covering a full system "safety lifecycle"**
  - **Allows a systematic and traceable mastery of safety risks**

➔ **The efficiency of the engineering teams involved in these activities will deeply depend on adequate processes, methods and tools**

Consequences of Complexity Inducing Factors: Vehicle Electrification 1

- The ultimate levels of CO_2 reduction can only be reached through the electrification of the vehicle powertrain
  - Full Electric Vehicles (EV) and/or Plug-in Hybrid Vehicles
- Which means…
  - Mastering new technologies, like electric motors, batteries, high voltage…
  - New use-cases, new measures of effectiveness and new trade-offs (e.g. EV charging, Vehicle autonomy in a Zero-Emission mode)
- Which implies…
  - Potential obsolescence of previous vehicle architectures
  - A need to re-shuffle established choices and trade-offs associated to Internal Combustion Engines

➔ Architecture management, based on a new architectural approach, becomes of outmost importance

Consequences of Complexity Inducing Factors: Vehicle Electrification 2

- **For Electric Vehicles and ADAS the system-of-interest is no longer the vehicle alone**
  - **Many off-board functions and parts: connected EV's, vehicle-to-vehicle / vehicle-to-infrastructure communications, automobile as part of seamless inter-modality transportation…**
- **Which means unfamiliar challenges**
  - **On the technical side: the vehicle evolving in a different ecosystem and becoming a *Constituent System* of a larger *System of Systems***
  - **At enterprise level: new interests, concerns and business models**

➔ **This is an important source of complexity for carmakers, for which methods and tools are expected**

To Summarize

- **Methodological breakthroughs are necessary**
- **But they shall combine with "classical" automotive particularities (huge product lines and diversity at vehicle, system and part levels)**
  - **High stakes associated to the efficiently mangage this product lines**
  - **Maximize the reuse of any engineering artifacts**

02

MBSE & Architecture Frameworks:

A Brief Take

Attempts to master complexity, improve quality of products or adapt / improve the efficiency of production cycles

- **The most typical reactions**
  - **Process improvement initiatives (standardization of processes, adoption of maturity models)**
  - **Implementation of SE**
  - **Model-based design processes**
  - **Definition and adoption of architecture frameworks**

A Brief Take on MBSE 1

- **Move from prose forms to explicit (and theoretically unambiguous) data structures and representations of the system**
  - **Expected benefits are: improvement in quality and communication, costless traceability, increase of productivity**
  - **A model and, thus, a modeling language, should only be used to cover those aspects of a system that are relevant for a particular purpose**
- **But the use (and abuse) of models leads to the undesired effect of too complex representations**
  - **Multiple formalisms used sometimes to describe the same aspects of a system or to solve one single problem**
  - **Model-based approaches usually rely on model transformations to cope with different formalisms, with inherent limitations: number of supported formalisms, applicability to a single domain or lack of flexibility**
  - **System architects are confronted to a rather chaotic collection of views, which correspond to specialty domains or to somebody else's concerns**

A Brief Take on MBSE 2

- On the methodological side, there are multiple processes, methods and languages available to systems engineers
- But almost all of the methodologies propose solutions stemming both from the software, real-time systems domains and from the SE domain with a scope that is not always clear-cut
- Few propose a design method capable of producing *in fine* a consistent holistic view of the system
- Best-case scenario: a lot of tailoring has to be done to meet the specific needs of a particular industrial domain

Architecture Frameworks

- **Have proven particularly beneficial in some industries for the management of large complex system projects**
- **They are standard sets of elements that enable and facilitate the description of architectures and provide guidance on how to create such description**
- **The best known examples (DoDAF / MoDAF and TOGAF) are tailored for the needs of the specific domains in which they were defined and are currently used**
  - **While useful as references for other domains, they reflect the way in which very specific industrial domains develop products or services**
  - **They correspond to their own particular needs**

03

MBSE in the Automotive Context &

State of Systems Engineering at Renault

How do I “Model-base” thee? Let me count the ways…

… To the depth and breadth and height
My soul can reach, when feeling out of sight

- **In 2007, an internal survey on the term model-based powertrain control showed that**
  - **The Scope ranged from the entire powertrain down to the elementary algorithm, with domains going from materials science up to embedded SW**
  - **Activities ranged from upstream 3D design up to final in-vehicle validation**
- **Every *“metier”* defines its own modeling activities**
  - **Well fitted for local efficiency improvements, albeit “sub-optimal” in terms of overall global efficiency**
  - **As long as those activities do not have to fit to a seamless overall process supported by compatible models**

The State of SE at Renault

- **Recently, Renault decided to deploy SE in a more systematic way through two complementary initiatives**
- **The first one can be summarized as “filling the least populated place” with respect to already established professional fields**
  - **We are well equipped in terms of parts engineering (managed in BOMs, specified by OEM, developed by Tier 1 providers…)**
  - **On the other end, customer requirements at vehicle level have a dedicated division and management process**
  - **So the “void areas” of vehicle engineering were the functional aspects of systems and the “layer” between the whole vehicle and its parts**
- **System perimeters were identified and system engineers were "allocated" to them**
  - **Focus is on building a generic system baseline, which is instantiated on every vehicle project**

The State of SE at Renault

- **The second initiative aims at preparing innovations in the Research and Advanced Engineering (R&AE) area with progressive but early implementation of advanced SE methodologies**
  - **Face the challenges and the complexity induced by new architecture concepts (e.g. EV), a change of scope (SoS)…**
  - **Master the problem and the solution space for more classical innovations, including safety and security aspects**
- **It was decided to name system architects in charge of relevant innovation projects**
  - **Progressively build the system descriptions for their innovation**
  - **Anticipate of innovation "landing" in downstream, established systems and vehicle projects**

  → A *natural* context for experimenting and tuning advanced SE methodologies

To summarize…

- As in any change management, we face numerous transient situations
- We are slowly but surely gaining in SE maturity
- It seems logical, then, to work on advanced techniques, like MBSE and Architecture Frameworks
- We believe these initiatives will contribute to the improvement of our competitiveness and to the acceleration of our SE maturation
- Standing where we are, however, the question is:

"Which way from here?"

04

A Commonsensical Approach to MBSE

Approach, Strategy and First Results

“If you don’t know where you are going, any road will do” (Chinese proverb)

“If you don’t know where you are, a map won’t help” (W. Humphrey)

The Philosophy

- **No model-driven, or magic-tool-driven, or the-best-theory-ever-driven approaches… just start with our own needs**
- **A Commonsense-Driven approach based on one question**
  - **How do we go about our business? Or How would we like to go about our business?**
- **How did we proceed?**
  - **Established a clear picture of the way we designed systems and "simply" compared it to the SE principles we were supposed to follow**
  - **Identified drawbacks in the implementation of the SE process and redefined the things we should do (the What) and the way we should do them (the How)**
  - **Looked at the state of the art of MBSE methodologies**

The Rationale on MBSE

Three very simple, pragmatic principles

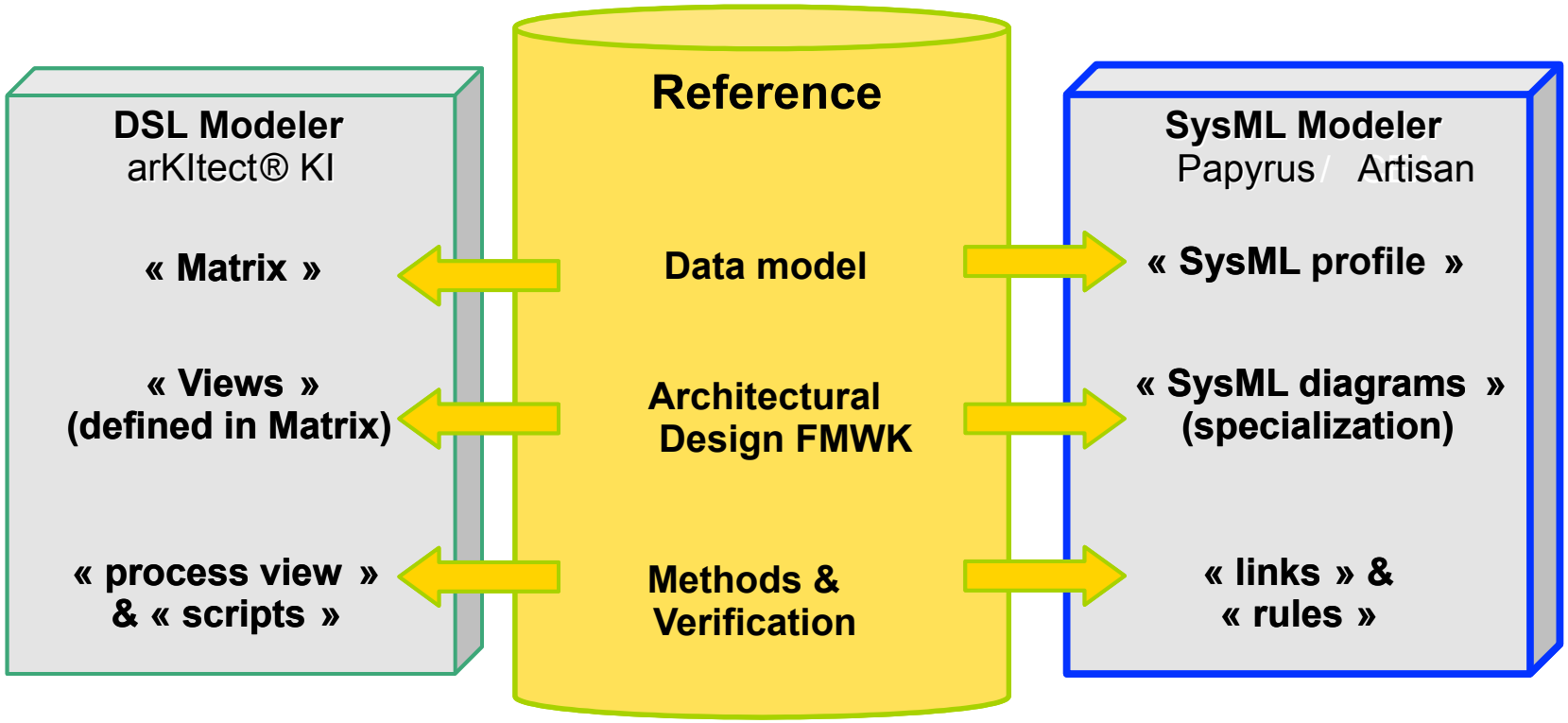
1. The activities and produced artifacts are defined and justified by the SE technical process; they add value to the SE product (the system)
2. The way to carry out these activities, plus the techniques that are strictly necessary to carry them out, must be defined by design Methods
3. Modeling and description Languages, as well as the Tools implementing them, should be used if, and only if, they support a design method

The General Procedure

- We formalized the SE process and its activities (*the What*)
- We aligned our methods (*the How*) to the methodology that best suited our needs
- We identified and defined the artifacts (or objects) and the graphical representations (or views) to use or produce in each activity of the SE process
- Finally, all of these objects, views and links amongst objects and views (or dependencies) led us to the definition of a data model (or meta-model) and of an architectural design framework

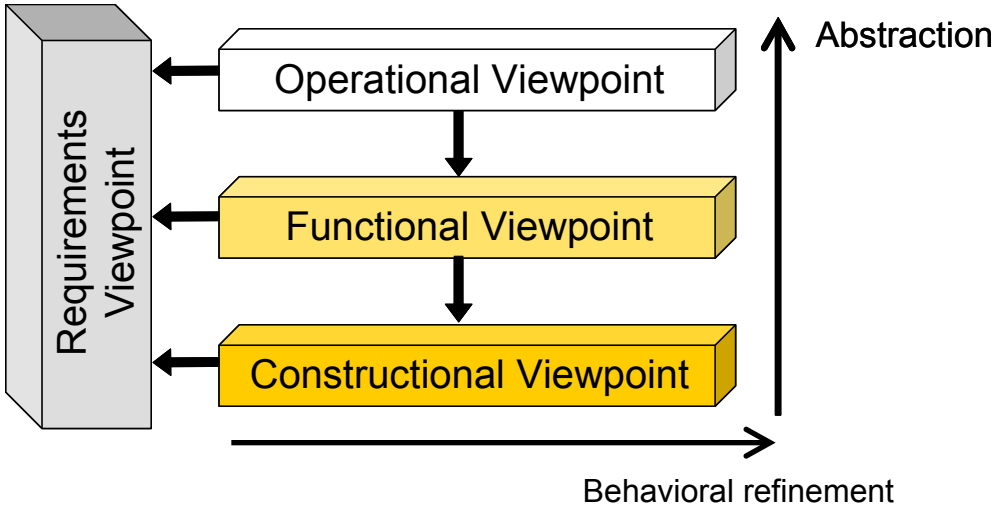
→ This is the reference to which all modeling environments have to comply

First Results



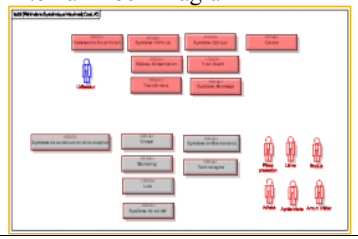
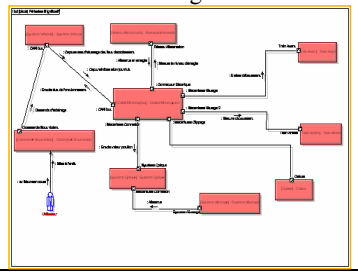
Overview of the Framework

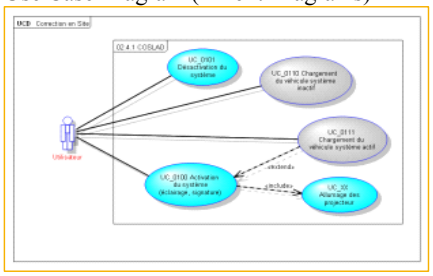
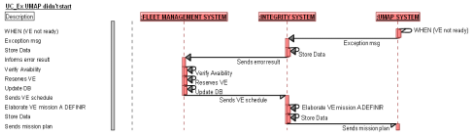
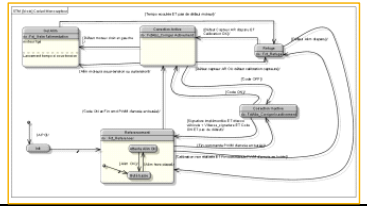
- **Analyze the system from different points of view** (abstraction levels)
- **And different refinement levels**



Viewpoints	Associated Views
Operational	Maximal System Scope
	System Environment
	Operational Context
	External Interfaces
	Use-Cases
	Operational Scenarios
	System Working Modes
Requirements	Stakeholder Requirements
	High-Level Requirements
	System Technical Requirements
Functional	Functional Breakdown Structure
	Functional Architecture
Constructional	Product Breakdown Structure
	Organic Architecture

Implementation of the Framework

Operational Viewpoint		
View	*SysML Diagram*	*Modelled Elements*
Maximal Scope	Internal Block Diagram 	*Maximal scope actors:* stakeholders, users, enabling systems and environmental factors potentially impacting the system under study. *Maximal scope categories:* actor categories to include in the maximal scope (there should at least be one actor of each category).
System Environment (Operational Scope)	Internal Block Diagram 	*System boundaries*: separation between external actors and the system of interest (or actors that "are part" of the system). *High-level interactions:* interaction or relationships between external actors and the system (including points and connections)
Operational Context	Internal Block Diagram 	System environment elements pertaining to a specific system life-cycle phase. *Life-cycle phases*: development, manufacture, assembly, delivery, use / exploitation, maintenance, recycling. (not shown on this table)
External Interfaces	Internal Block Diagrams 	Refinement of the system environment into two context diagrams: *Functional Context Diagram:* specification of flows exchanged between the system and its environment (matter, energy, information). *Physical Context Diagram:* specification of the physical means through which the flows are transported (mechanic, electric, network).

Operational Viewpoint		
View	*SysML Diagram*	*Modelled Elements*
System Use Cases	Use Case Diagram (+ Text Diagrams) 	*Use cases*: expected services and behaviors of the system from the point of view of system actors *Use case conditions:* structured descriptions of initial conditions, system reactions, successful end condition and failed end conditions for a use case.
Operational Scenarios	Sequence Diagram for operations and actor requests or Activity Diagram for internal actions and flows (depends on the type of system) 	*Interactions* between the system of interest and its enabling systems, actors or environment to fulfill the use case (or produce the expected behavior). *Actor requests* and system *Operations* in response. *Actions* to further detail each operation, control flows between actions. *System states / transitions* to synthesize the system behavior and alternative conditions of operational scenarios.
System Working States	State machines 	

05

Conclusions and Perspectives

On the Work Carried Out

- **Our approach is the result of a combination of many good ideas (inside and outside Renault) and of the analysis of different attempts and experiments, many of them unsuccessful, encountered during the deployment of SE and of Model-Based Design**
- **This kind of initiative certainly requires a wider adoption from the different actors of the automotive industry, in order to take full advantage of all the benefits made possible by such approach and to ease the utilization of supporting tools**
- **We are currently trying to establish contacts and to communicate about this work to foster the exchanges on the subject**
  - **A first meeting of the Automotive Interest Group held at IS2012, plus the work presented here, will hopefully be the object of discussions, adaptations and modifications**

On Our Future Work

- **We are beginning to work on an enriched framework that will support**
  - **Verification and Validation activities through simulation and optimization techniques**
  - **Product Line Management to fill the gap between top-down systems design and bottom-up integration of legacy elements (components, technologies)**
  - **Safety Processes, as defined by ISO 26262**
  - **A seamless transition to Software Design environments using UML**

- **BUT…**

Will this be a trip to a *Never-Never Land*?
Or will we live *happily ever after*?

END

OR IS IT… ?