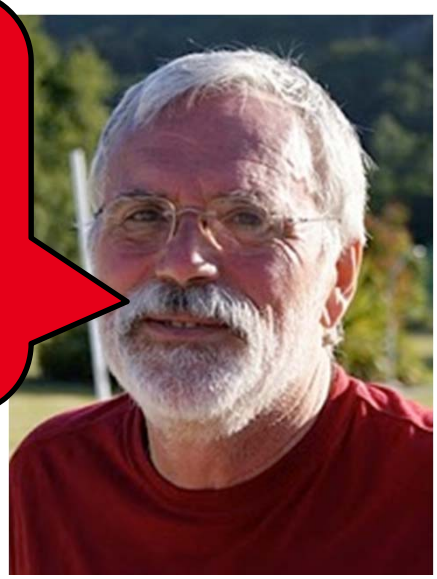
Sébastien Gérard
(sebastien.gerard@cea.fr)
CEA Tech / LIST, France

Model-based System and Software Engineering: why and how?

MANY THANKS TO BRAN SELIC FOR ALL HIS DIRECT AND INDIRECT CONTRIBUTIONS TO THIS PRESENTATION.

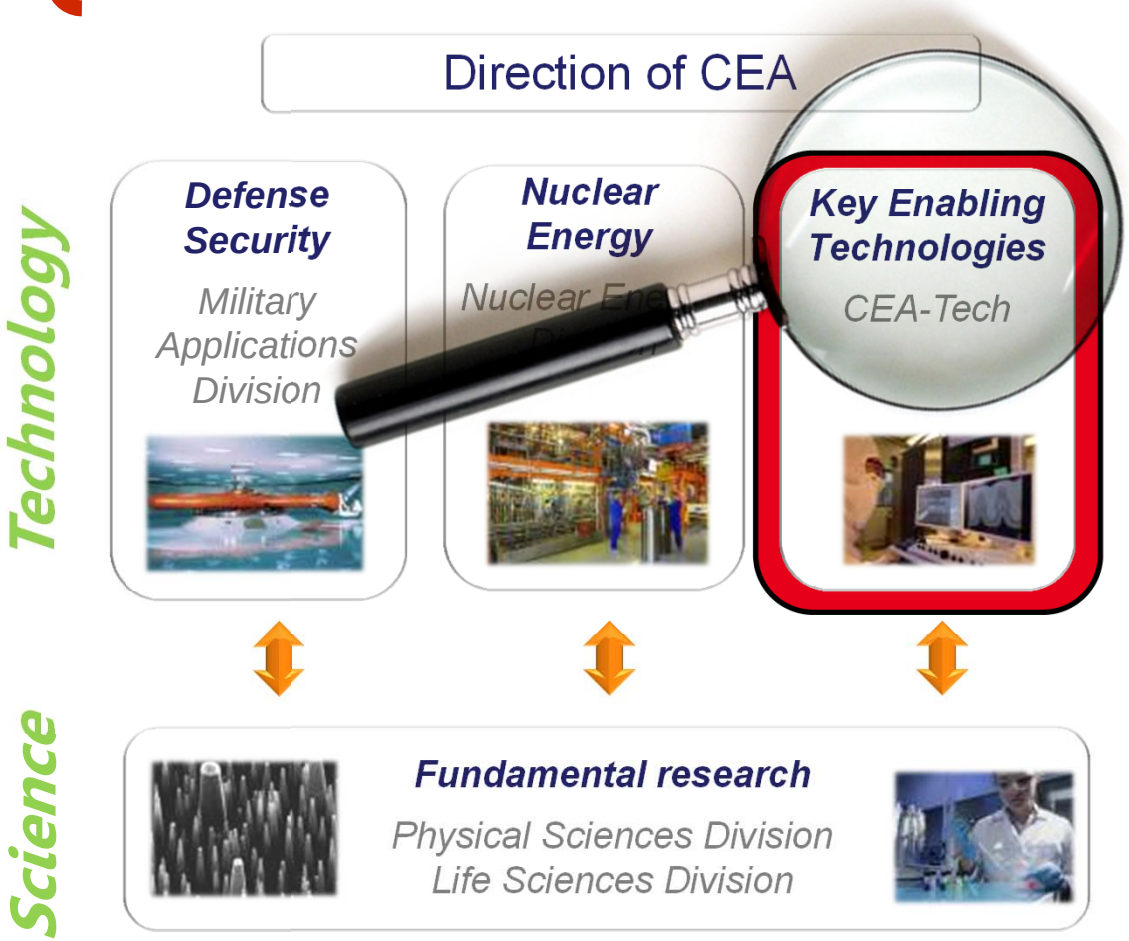
Bran Selić (selic@acm.org)

Malina Software Corp., Canada
Simula Research Laboratory, Norway
Zeligsoft (2009) Limited, Canada
University of Toronto, Canada



CEA KEY FIGURES

CEA is a major actor in research and innovation.



- **16 000 people**
- **16 centers in France**
- Budget : 4,3€ bilions
- 1 600 patents
- **4 000 publications / year**
- 150 startup created since 1984

THE LISE LABS IN A NUTSHELL

Correct-by-construction design of safe CPS

- A laboratory of 53 persons
  - 35 permanent members + 18 non-permanent members including PhD students, post-docs and CDDs

Main research concerns

- Modeling Language Engineering
- Model-based Formal Analysis (e.g., auto gen. of tests)
- Run-time Formal Verification and Monitoring
- Model-based Simulation
- Model-based Security & Safety Engineering
- Archi. Exploration, Configuration & Deployment
- Process, Requirement and Variant Engineering

So called "smart systems" are everywhere, deeply involved in our daily life.

Question: what is their common point?

➔ **Most of their innovation relies on their embedded software!**

But sadly, they also often share…

GM Recalls 50,500 2011 Cadillac SRXs Over Airbag-related Software Glitch

By Robert Charette
Posted 22 Jun 2011 | 12:48 GMT

Share | Email | Print

GM is recalling 50,500 Cadillac SRX cross-over vehicles because of a software glitch that may not allow the deployment of airbags for passengers sitting in the right rear seat during a crash, reports a blog post at Zacks Investment Research.

According to GM, the post says:

"...the front passenger

What Industry Needs from Architectur Languages: A Survey

Ivano Malavolta, Patricia Lago, Senior Member, IEEE, Henry M Patrizio Pelliccione, and Antony Tang, Member, IE

Engineering Automotive Software

For the gigabyte of costly software that will be used in intelligent cars of the future, system and software development techniques and tools are required.

Why Sof FAI

Telecom continues whirlwind of settlements with incorrect broadband meter readings

Oct. 19 (BusinessDesk) – Telecom has continued its rush to settle outstanding disputes ahead of next month's vote to split the company, reaching a deal with the Commerce Commission to repay broadband customers overcharged for incorrect meter readings.

The country's biggest phone company paid out $2.7 million to some 47,000 customers who were overcharged after a software glitch meant people hit their data limits early, the antitrust regulator said in a statement.

Telecom and the commission reached a settlement after the phone compan acknowledged the fault and sought to compensate its custom regulator would waive its right to issue legal pro Fair Trading Act.

Released by:

Technical Operations

International Council on Systems Engineering (INCOSE)

resh crash

Related Stories

Can the iPhone still scare rivals?

Microsoft services h by failure

Android 'most popular' purchase

ABOUT THE AGENDA

- **Architecture and what & why MBE**
  - Outline architecture concern, then introduces and defines MBE and explains its added value.
  - Impact studies: a selective summary of published results of industrial use of MBE.
- **How to enable model-driven engineering?**
- **And what about MBE for mission critical, realtime embedded software engineering?**

RATIONALE FOR BEING ARCHITECTURE-CENTRIC

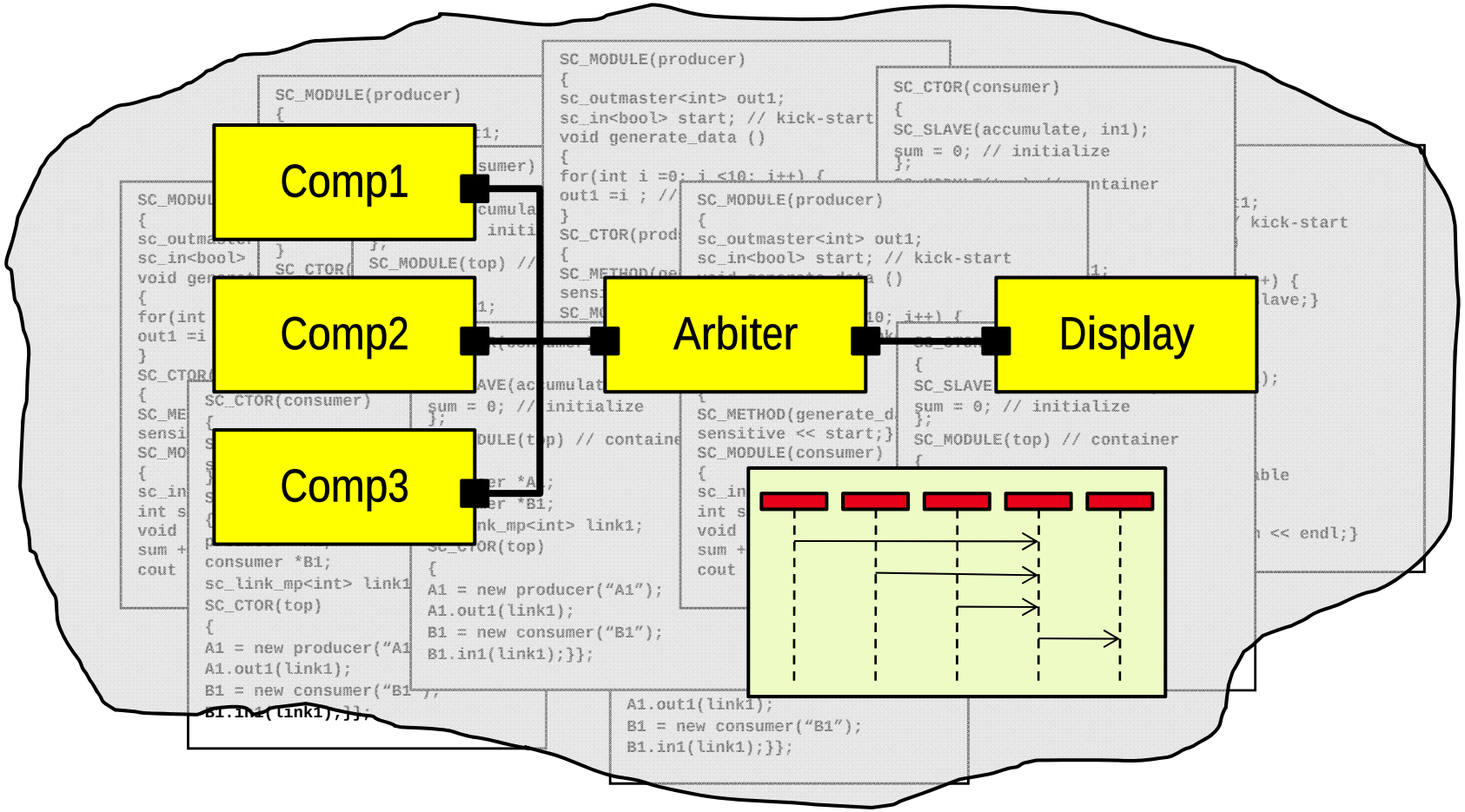
- **Separation of concerns is a good and widely applied principle for coping with complexity**
  - E.g., Design-Pattern, Aspect-Oriented Modeling, or Service-Oriented Architecture.

- **But the different concerns are seldom independent ☹**
  - E.g., performance vs. safety or cost vs. security.

➔ Requires a "big picture" approach to ensure system integrity & consistency: Architecture Description.

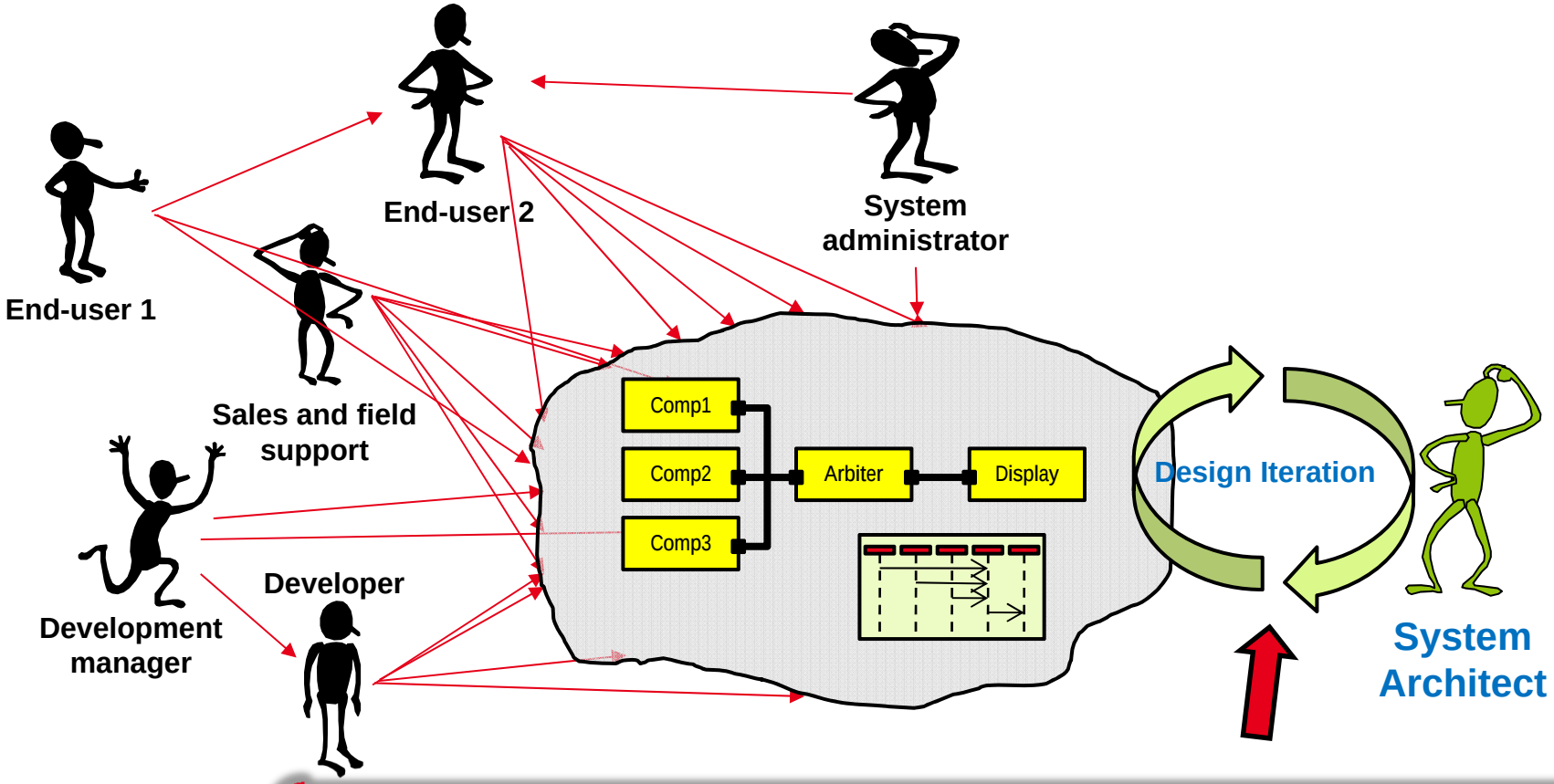
> *"Fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution"*
>
> *(definition extracted from ISO/IEC 42010)*

ARCHITECTURE DESCRIPTION LANGUAGE (ADL) ARE USUALLY GRAPHICS



(slide credit to Bran Selic)

HOW ARCHITECTURE HELPS DEFINE REQUIREMENTS

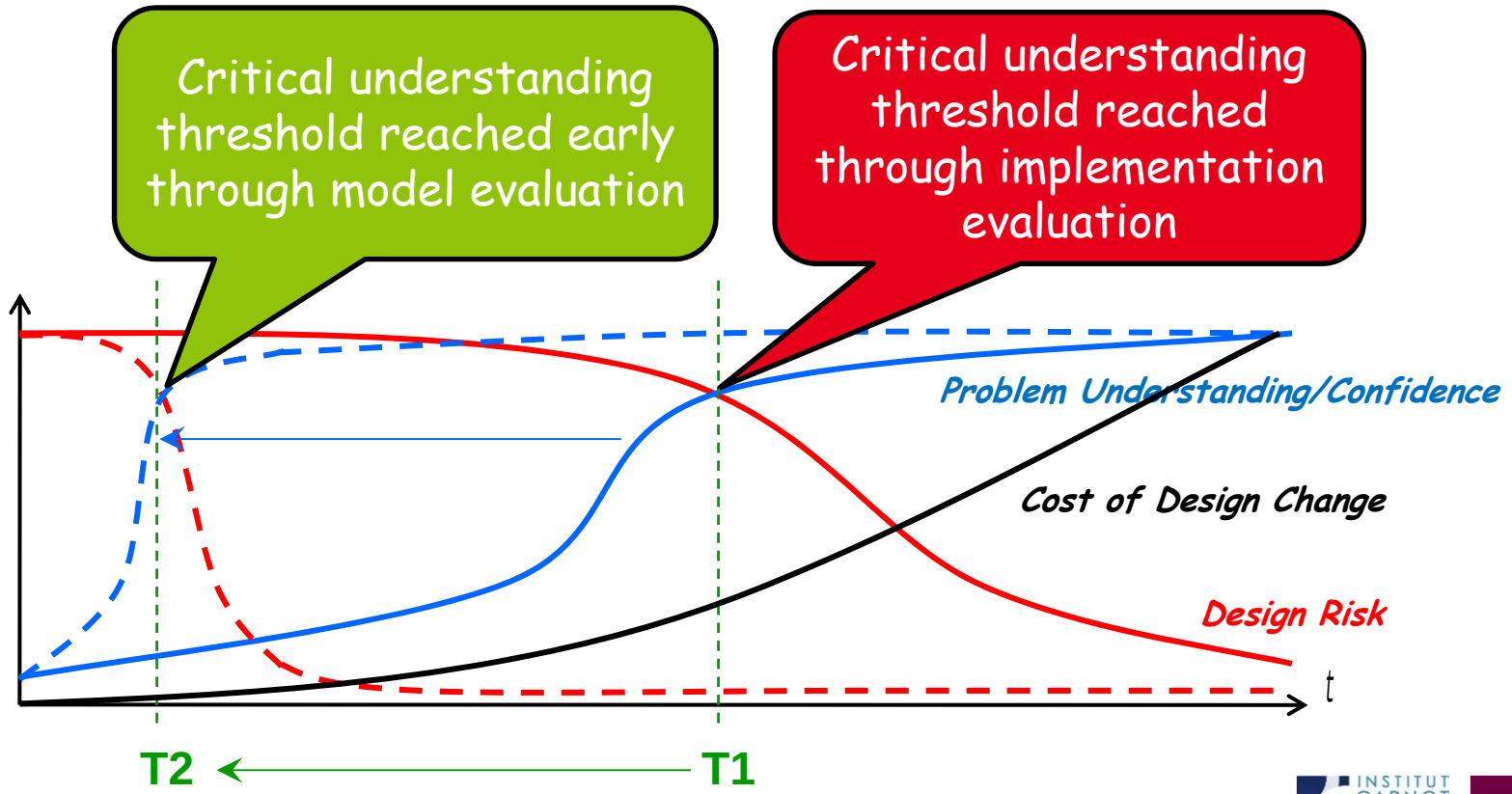


Many requirements conflicts and necessary tradeoffs are only detected through analysis of candidate architectures.

(slide credit to Bran Selic)

ARCHITECTURAL EXPLORATION REDUCES RISK

- **Repeated evaluations of architectural models (e.g., using simulation, formal and informal analyses)**
  - Early experience with the design ➔ earlier detection of potential design flaws ⇒ less expensive to fix!



(slide credit to Bran Selic)

SUMMARY: ARCHITECTURE-CENTRIC BENEFITS

Architecture description does help in designing systems, because:

- **It improves stakeholder communication**
  - Concrete/tangible representation used as a focus of discussion by stakeholders of the system development
- **It enables team working**
  - Used to distribute the tasks along working teams and Used to drive integration of its implemented subsystems
- **It reduces development risks by enabling early analysis, verification and validation**
  - Used for validation to know whether the system can meet its non-functional requirements ➔ very important result for RTE systems!

(slide credit to Bran Selic)

COMPLEXITY, COMPLEXITY, COMPLEXITY, COMPLEXITY, COMPLEXITY…

↗ **complexity of systems to design**

➔ *more functions, more concerns and more interactions.*

+

↗ **complexity of design constraints:**

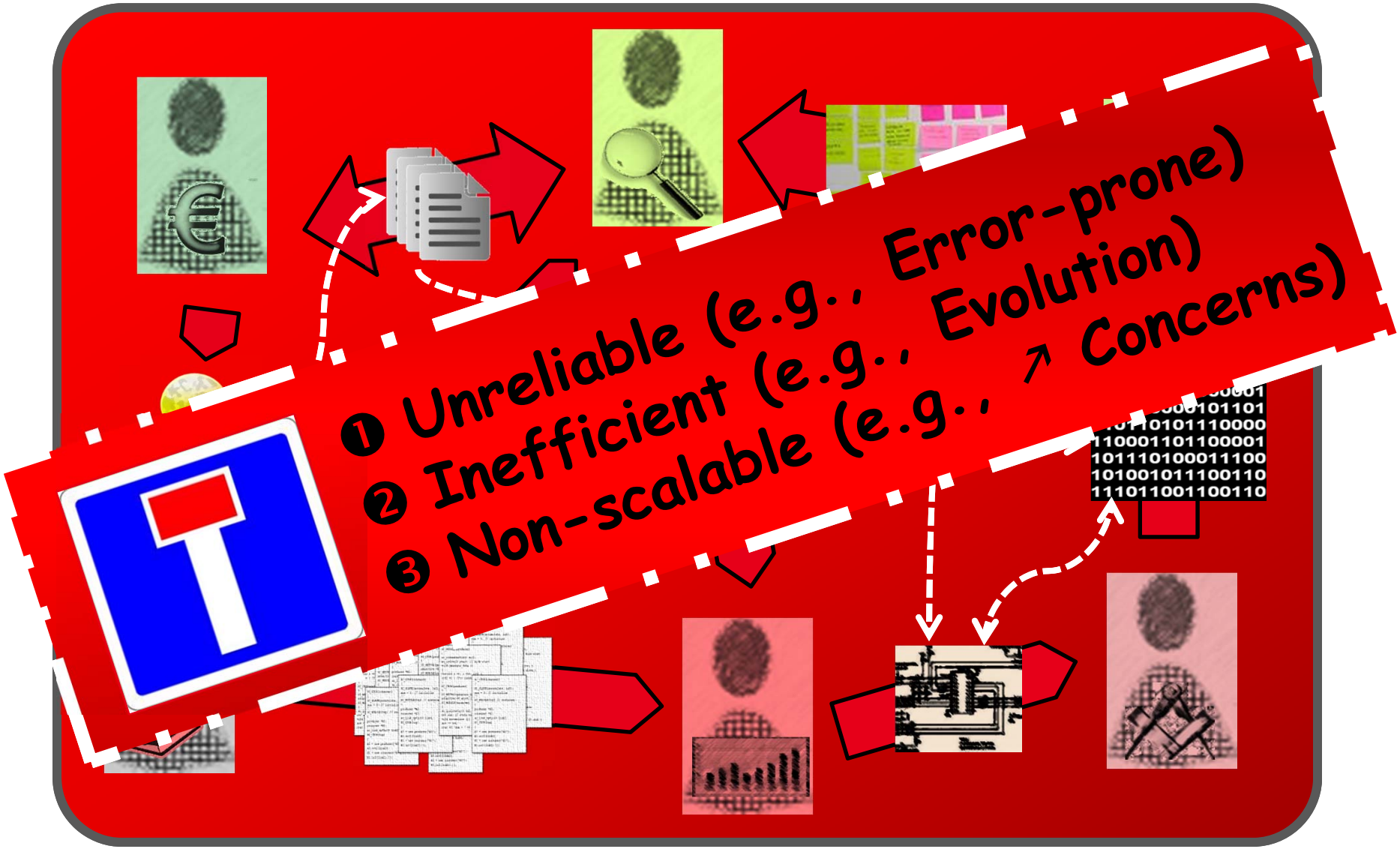
➔ *quicker, more constraining standards, better quality and cheaper!*

WHAT IS THE MOST ANCESTRAL PRINCIPLES FOR DEALING WITH COMPLEXITY?

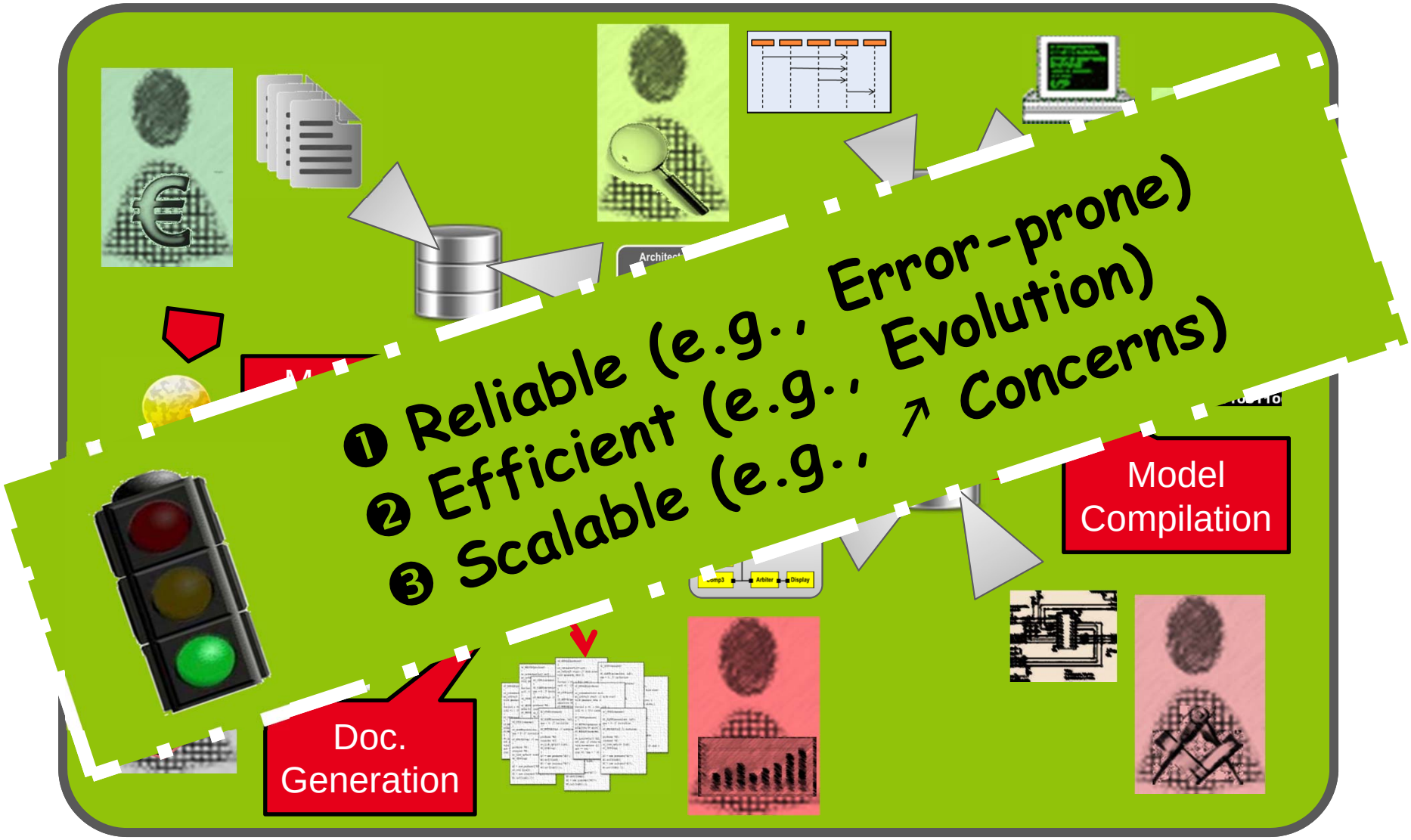
➔ Abstraction ➔

Definition: "**Conceptual process** consisting in reducing the information content of a concept or an observable phenomenon, typically to retain only information which is relevant for a particular purpose."

OUTLINES OF TRADITIONAL DEV APPROACHES



OUTLINES OF MODEL-BASED DEV APPROACHES



THE THREE PURPOSES OF ENGINEERING MODELS

To facilitate communication among stakeholders.

To support reasoning about a design.

To serve as precised specifications (blueprints) for constructing systems.

FROM ARCHITECTURE-CENTRIC DEV…

…TO MODEL-BASED DEV

System architecture is a key element of system/software development and the management of its essential related complexity.

- **Architecture-centric design has opened the door to the need/use of modeling languages:**
  - Need to express the concepts of architecture description: decomposition, abstraction and view.
  - Need to denote explicit relationships between elements at different abstraction levels and projected in different views.

Architecture-centric Dev → **Model-based Dev**

FOUNDATIONS OF MODEL DRIVEN ENGINEERING

Foster computer-aided development (including specification and design) to enable correct-by-construction of complex systems.

2 main pillars for Empowering MDE

=

Abstraction

Suitable and sound modeling language engineering

+

Automation

Efficient and scalable computer-aided engineering

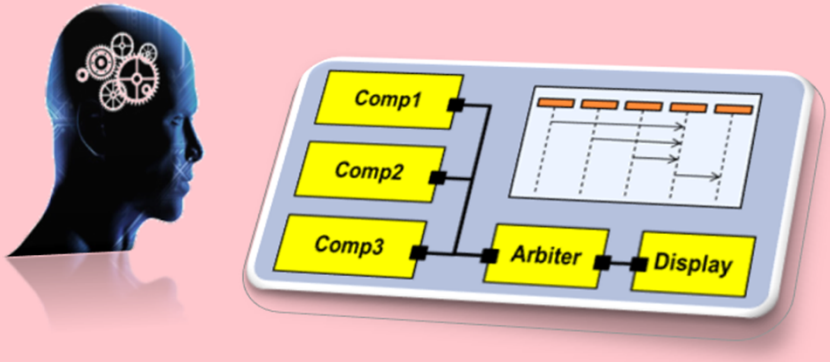
AUTOMATION (OR COMPUTER-AIDED), WHY?

- **Foster model analysis by enabling integration of complementary external tools**
  - e.g., formal mathematical analyses, model simulation or testing tools.
- **Provide concrete support for refinement-based processes**
  - Better robustness of processes (e.g., no cut & past errors),
  - More efficient to deal with evolution
    ➔ Support for tracking, verifying and propagating changes in models.
- **Enable generation of consistent documentations and implementations**
- **Empower process enactment**
  - Monitoring, driving, and synchronizing of development processes

MODEL ARE PRODUCTIVE ASSETS: FROM MENTAL TO COMPUTER-AIDED MODELING.

Contemplative Models

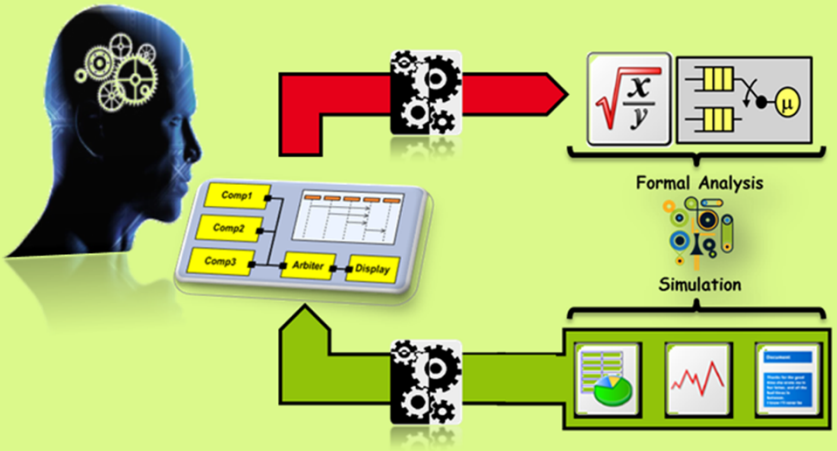
Mental-based Modeling



High-expertise
Unreliable
Not-scalable

Productive Models

Computer-aided Modeling



↘ expertise
↗ reliability
↗ scalability

MAIN KEY BENEFITS (KB) OF MODELS/MBE

KB1 - Working with higher levels of abstraction closer to problem domain

KB2 - Automatically traceable links between related model elements

KB3 - Potential for stakeholder-oriented system representation (views) of complex systems

KB4 - Ability to automate some engineering tasks (e.g., design patterns or V&V analyses)

Each of these characteristics can directly impact positively: quality, productivity and complexity management.

KB1 - WORKING WITH HIGHER LEVELS OF ABSTRACTION CLOSER TO PROBLEM DOMAIN

- **Quality impact**
  - Fosters creation of simpler, better structured, and more maintainable designs
- **Productivity impact**
  - Reduces cognitive load on developers
  - Simplifies communication between stakeholders
  - Simplifies post-release maintenance (due to more effective system documentation)
- **Complexity management impact**
  - Reduces need to perform domain to technology transformations during design and review
  - Reduces complexity by hiding implementation/technological detail

KB2 - AUTOMATICALLY TRACEABLE LINKS BETWEEN RELATED MODEL ELEMENTS

- **Quality impact**
  - Easier detection of complex system couplings and unanticipated effects of design choices and changes
  - Simplifies assessment of requirements coverage
  - Simplifies detection of extraneous design elements
  - Minimizes or eliminates information duplication
- **Productivity impact**
  - Easier detection of design issues stemming from unanticipated couplings
  - Simpler post-release maintenance
  - Minimizes or eliminates information duplication
- **Complexity management impact**
  - Fast and reliable support for finding couplings between complex system components (e.g., determining impact of proposed design change, determining requirements coverage)

KB3 - POTENTIAL FOR STAKEHOLDER-ORIENTED SYSTEM REPRESENTATION (VIEW) OF COMPLEX SYSTEMS

- **Quality impact**
  - Enables more accurate capture of both requirements and design intents.
- **Productivity impact**
  - Fosters faster and more reliable decision making due to more effective communication between stakeholders.
  - Simplifies post-release maintenance (due to more effective system documentation).
- **Complexity management impact**
  - Reduces complexity by hiding implementation/technological detail and by customizing system representation according to stakeholders concerns and ontologies.

KB4 - ABILITY TO AUTOMATE SOME ENGINEERING TASKS (E.G., DESIGN PATTERNS OR V&V ANALYSIS)

- **Quality impact**
  - Reduces or even eliminates errors caused by flawed or incomplete human reasoning (e.g., "cut and past errors"!)
  - Increases likelihood of sound design decisions due to trustworthy V&V
  - Reduced design risk (e.g., thanks to model simulation)
- **Productivity impact**
  - Accelerates execution of key steps in the design process
  - Early detection of design flaws
  - Increased confidence of design team
- **Complexity management impact**
  - Automation amplifies ability to perform complex analyses by orders of magnitude

SUMMARY: MBE, WHAT AND WHY

Model-based engineering (MBE) is a paradigm for designing and implementing complex systems in which computer-based models play a fundamental role.

- **Based on two fundamental principles:**
  - Higher levels of abstraction
  - Higher support of computer automation
- **Key potential benefits are:**
  - Increased productivity
  - Increased product quality
  - Greater ability to manage growing complexity

WHAT IS THE CURRENT STATUS OF ITS USAGE?

TWO SYSTEMATIC STUDIES OF INDUSTRIAL USE OF MODEL-BASED ENGINEERING

- **Stevens Institute of Technology (US): Analysis of SysML Usage RFI**
  - A study initiated and conducted on behalf of the OMG (2009) & INCOSE
  - Focus on SysML usage (Model-Based Systems Engineering - MBSE)
- **U. of Lancaster (UK) Project: "Empirical Assessment of the Efficacy of MDE" (EA-MDE)**
  - A general study of MDE use in industry

SURVEYS AND EXPERIENCE REPORTS RELATED MDE

- **R. Cloutier and M. Bone, "Compilation of SysML RFI – Final Report", Stevens Institute of Technology, 2010**
  - Systematic study of the use and effectiveness of model-based methods in systems engineering in industry
- **J. Hutchinson, et al., "Empirical Assessment of MDE in Industry," ICSE 2011 (*)**
  - Systematic study of the effectiveness of model-based methods in for software development in industry
- **J. Hutchinson, et al., "Model-Driven Engineering Practices in Industry," ICSE 2011 (*)**
  - Systematic study of the level of use of model-based methods in for software development in industry
- **P. Mohagheghi and V. Dehlen, "Where is the Proof? – A Review of Experiences from Applying MDE in Industry," ECMDA 2008 (*)**
  - Review of available publications on industrial application of MBE in industry
- **T. Weigert and F. Weil, "Practical Experiences in Using Model-Driven Engineering to Develop Trustworthy Computing Systems," IEEE SUTC 2006**
  - Summary of systematic use of MBE in Motorola with evaluation
- **The Middleware Co., "Model-Driven development for J2EE Utilizing a Model Driven Architecture (MDA) Approach," 2003**
  - A systematic comparative study of traditional vs. model-based development on a software project

(*) = Sources that include extensive references to other surveys and experience reports

EXECUTIVE SUMMURY OF THESE IMPACT STUDIES

- **All these diverse and widespread industrial experiences with MBE has demonstrated that it is effective in:**
  - Increasing productivity and product quality
  - Improving com between stakeholders => dealing with complexity
  - Improving maintainability
  - Faster introduction of new development staff
- **Based on:**
  - Several broadly-scoped systematic studies of industrial use of MBE in industrial environments
  - Numerous reported experiences of individual development organizations in a variety of different industrial domains

However, these studies also show that introducing MBE must be approached systematically with careful planning.

ABOUT THE AGENDA

- Architecture and what & why MBE
  - Outline architecture concern, then introduces and defines MBE and explains its added-value.
  - Impact studies: a selective summary of published results of industrial use of MBE.

- **How to enable model-driven engineering?**
  - How to introduce MBE into a development organization.
  - A general overview of MBE tools and related industry trends with special focus on open source tooling.
  - Example of Papyrus, a FOSS for MBE

- **And what about MBE for mission critical, realtime embedded software engineering?**

INTRODUCING MBE INTO AN ORGANIZATION

Unless the introduction of MBE into a legacy organization is carefully and systematically planned and executed, there is a very high likelihood that it will not be successful or that the results will be disappointing.

- There are numerous hurdles that need to be anticipated and overcome in a gradual process.

And main hurdles are indeed due to the effects of culture change rather than due to technical issues!

PRIMARY HURDLES TO SUCCESSFUL ADOPTION OF MBE

- Inadequate corporate commitment
- Inexperience of development staff
- Technology boycott by development staff
- Inadequate languages / tools
- Unrealistic expectations – overly ambitious first project
- Cost of training
- Cost of retooling

GET THE CORPORATE COMMITMENT

Instituting MBE into a legacy development environment requires a strong and highly visible commitment by upper management.

- A true “sine qua non” condition
- Cannot be achieved as “shunkworks”

Identify corporate prime for instituting MBE.

Budget resources.

Define a strategic roadmap and implementation plan.

Define success metrics and track progress continuously.

Publicize successes internally.

OVERCOMING INEXPERIENCE WITH MBE

At the start, collaborate with those who have already succeeded with MBE

- Other non-competing enterprises
- MBE experts (external hires, consultants)

Develop core competency within the development team

- Start with a small but important (production-critical!) project
- Staff with top performers
- Work iteratively
  - Continuously record and measure progress, issues, solutions (including rationale)
  - Identify potential improvements at the end of each iteration
- Seed subsequent projects with (now) experienced MBE personnel

Set up systematic enterprise-focused training programme

- Involve MBE experts in defining curriculum
- Focus more on younger developers (i.e., those with lesser attachment to legacy methods and technologies)
- Customize training to own needs and update continuously based on own production experience

DEALING WITH TECHNOLOGY BOYCOTT

Driven by:

- Genuine concern about risks of “unproven” technology
- Fear of technical obsolescence (will I be able to master the new technology?)

First ensure buy-in from respected opinion leaders

- Individuals with system-level view of product(s) and even market knowledge/concern (i.e., those who care more about the product and less about the technology used to make it)
- Involve them in key decision making on new process, tools, etc.

Identify receptive individuals but also intransigent opponents

- Do not waste time on the latter category, leave them with legacy

Demonstrate viability of new approach by publicizing any successes internally

- Requires continuous tracking and measuring of new process

Be frank about MBE capabilities: do not try to oversell or hide technical impediments from development staff

- Fortunately: successful MBE projects have demonstrated clearly that none of these are showstoppers!

DEALING WITH INADEQUATE TOOLS / LANGUAGES

Languages

- Capture own domain ontology (metamodel)
- Investigate possibility of custom profile or domain-specific language
- Collaborate with other enterprises with similar interest (even competitors! e.g., Autosar)
- Actively participate in relevant standards bodies

Tools: consider investing in open source to develop desired (custom) solution.

- E.g., Eclipse
- Contract external parties or develop in house
- Collaborate with other enterprises (even competitors!) to share R&D costs
- Institute own tools strategy group to identify and define requirements
- Seek tools with powerful customization capabilities
- May need a tool adaptation team

Use corporate leverage to influence vendors

- (NB: experience has shown that this is often very slow and unreliable)

AVOIDING OVERLY AMBITIOUS FIRST PROJECT

Set realistic expectations

- Do not forecast dramatic improvements on first pass
  - E.g., between -20% and +20% most likely
- **Do not oversell**: Identify clearly and honestly potential hurdles that must be overcome

Select relatively small but production-critical project

- To ensure proper motivation to make things work
- Project must have relatively high likelihood of overcoming potential hurdles ⇒ needs top performers

Work iteratively

- Identify promising improvements at the end of each iteration

Measure and document all facets (issues, solutions)

- Encourage candid and objective reporting (e.g., no covering up or misrepresenting of issues encountered)
  ➔ Requires a culture in which it is acceptable to report mistakes without fear

MINIMIZING COST OF TRAINING

Favour technologies and methods based on industry standards

- Easier to find staff who are familiar with technology, languages
- Easier to find available training material
- However, invariably requires some customization of training

Start initial training with small team of top performers and project and product managers

- Start with market-available language, method, and tool training
- Work with MBE experts to determine focus of initial training

Gradually evolve a custom training programme

- Identify training responsibility primes
- Work with experienced internal staff and internal/external trainers to determine syllabus
  - Seek feedback from current MBE project participants
- Do not attempt to retrain all development staff after the first successful project

MINIMIZING COST OF RETOOLING

New processes require new tools

Look for solutions that have multiple suppliers or open source tools

- Avoid vendor lock in (there a lot of examples where commercial vendors discontinue support for their products after a given period)
- Favour standards-based solutions and open solutions

Favour tools that interoperate readily with other tools

- Including existing legacy tools (compilers, version management systems, etc.)
- Tools based on a common tool framework (Eclipse, Tornado, etc.)

Favour tools with strong customization capabilities

Actively experiment with multiple alternatives (if available) before committing

- Invest in a comparative empirically-based analysis
- Identify key comparison criteria and measure against them
  - Interoperability, scalability, usability are the usual primary criteria

SUMMARY: IMPEDIMENTS TO SUCCESSFUL INTRODUCTION OF MBE

- Introducing MBE into a legacy environment requires a carefully planned long-term strategy (not a six-months project!)

- Experience has shown that attempts to introduce MBE fail primarily due to non-technical reasons:
  - Insufficient corporate-level commitment
  - Rejection by technical staff
  - Inadequate training

Although there can still be serious technological hurdles, the proven success of numerous MBE industrial projects indicates that all of them can be overcome.

FOCUS ON MBE TOOLING

- **Currently, inadequate tooling can be a major issue to the application of MBE…**

  …But, not insurmountable!

- **Automation (computer-aided) is a foundational element of MBE**
  - Greater reliance on computer-based tools than traditional development methods

- **Although improvements in MBE tooling are accelerating, some issues can still be problematic, notably:**
  - **Scalability**: tools do not scale up to large models
  - **Adaptibility**: difficult to adjust to custom needs
  - **Interoperability**: proprietary tool formats
  - **Cost**: tools licences and training
  - **Usability**: tools are often very complex to use

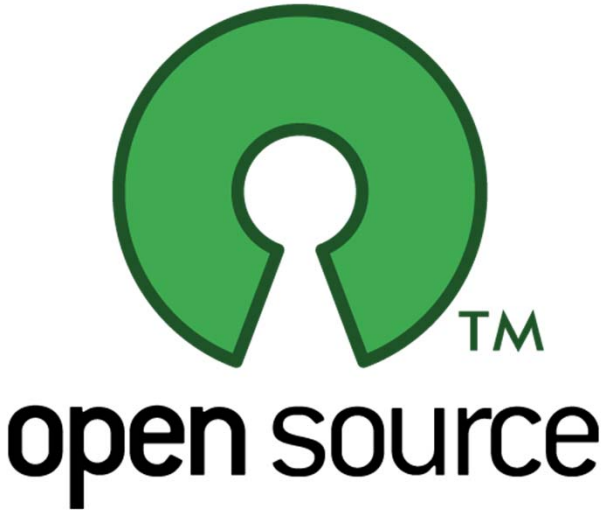
CURRENT TRENDS IN MBE TOOLING

- **Moving from vendor-driven to end-user-driven approach**
  - The domain-specific nature of MBE languages demands numerous highly specialized tools
  - Commercial vendors are reluctant and slow to respond to custom features
    - Priority given to high-volume features
    - Lack of domain expertise
  - End-user fears of vendor lock in
    - Some end users require very long term support (>50 years!)
    - No control of toolset capabilities

- **Greatly increased interest and investment in open source tools**
  - Protection against vendor lock in
  - Faster, more flexible, and easier tool customization ability

It's a good time to get involved in directing tool solutions

OPEN-SOURCE, THE DEFINITIVE CHOICE TO MAKE!



- **Main goals and values from OSS:**
  - Core technologies at the top of the state-of-the art.
  - Technology inline with industrial needs.
  - Increase standard usages by proposing an open reference implementation
  - Develop, and then benefit, a diverse ecosystem: experts, solution providers, students, etc.

STANDARDS ARE NOT AN OPTION

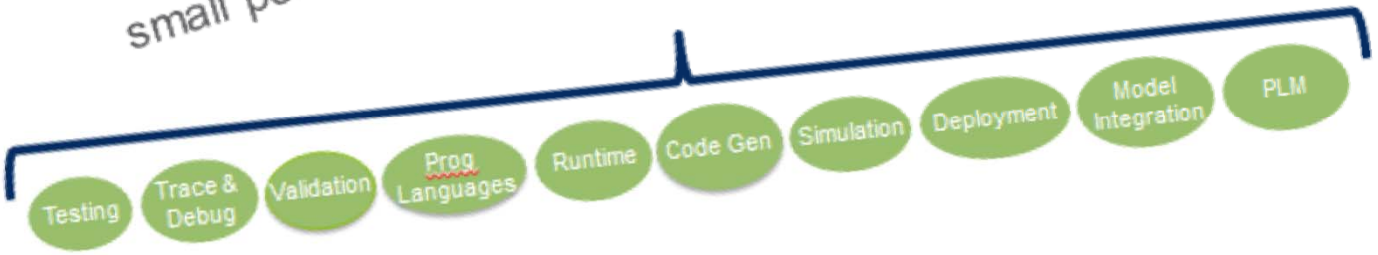
- **Usage of standards participates to cost and risk reduction:**
  - By fostering communication/exchanges between product stakeholders,
  - By improving tool interoperability,
  - By helping establish industry-wide norms for best practices,
  - By enabling availability of experienced engineers,
  - And by enabling vendor independence.

- **Standards are major boosts to technological progress**
  - By fostering vendors to compete and improve their products

UML Tool reality

Why doesn't it work in practice?

- The problem is not UML, but the UML tools!
- Commercial proprietary tools only support (very!) small portion of overall vision



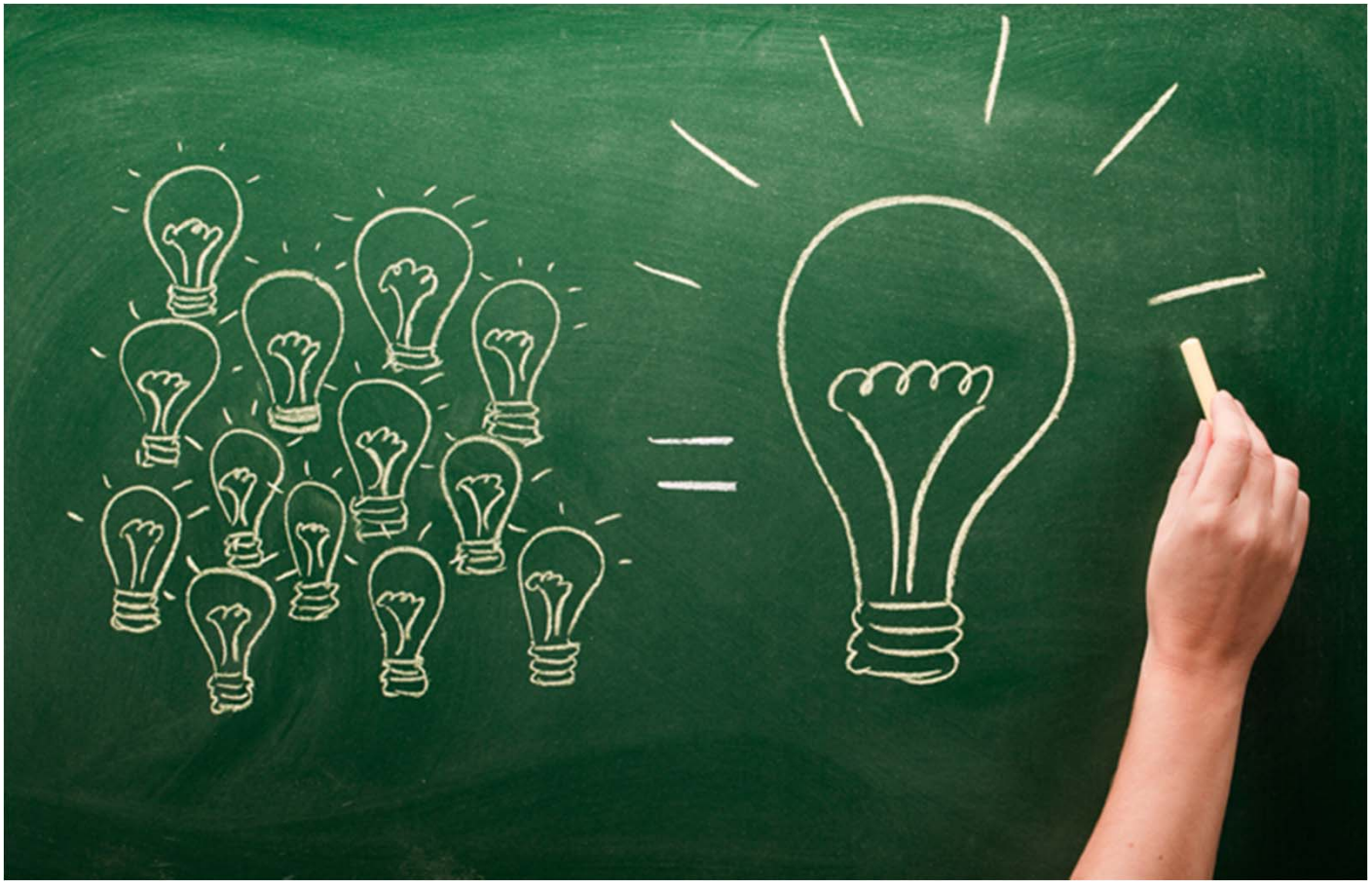
(Slide credit to F. Bordeleau, Keynote MODPROD Workshop, Linköping University, Feb. 3rd, 2015)

http://www.eclipse.org/papyrus

Licence: EPL
www.eclipse.org/papyrus

Eclipse Papyrus project delivers both a modeling tool for experienced UML/SysML modelers, and a platform for toolsmiths. As such, Papyrus enables the construction of custom modeling tools implementing specialized languages tailored for a specific application domain or company based on widespread modeling standard languages.

FOR SUCCESSFUL FOSS, COMMUNITY IS A PRIME-CONCERN



WHO IS PAPYRUS:
COMPANIES, INSTITUTIONS AND UNIVERSITIES

INSTITUT CARNOT CEA LIST

université PARIS-SACLAY

WHO IS PAPYRUS:
COMPANIES, INSTITUTIONS AND UNIVERSITIES

- User Lead members
- Supplier Lead members
- Participant members
- Academic/University members

Structrured & organized within the PIC (Papyrus Industrial Consortium) @ Eclipse Polarsys WG founded in January 2016.

WHO IS PAPYRUS? PEOPLE


A. Henriksson
Ericsson


S. Dhouib
CEA


A. Le Guennec
Esterel Tech.


P. Langer
EclipseSource


X. Zeitoun
CEA


M. Koegel


S. Dirix


M. Fleck


A. Buzila
EclipseSource


J.-M. Bruel


S. Salmons
CEA


Ed. Willink


F. Jouault
ESEO


B. Maggi
CEA


C. Rivet
Zeligsoft


S. Révol
CEA


F. Bordelau
Ericsson


C. Janssens
All4Tec


J. Tatibouët
CEA


R. Schnekenburger
CEA


F. Noyrit
CEA


A. Charfi
CEA


C. Damus
Independant


N. Fargues
All4Tec


K. Hussey
Commiterati


P. Tessier
CEA


S. Redding
Zeligsoft


S. Gérard
CEA


A. Tischenko
Montages


E. Stepper
es-computersys.


D. Lopez
CEA


O. Beaudoux
ESEO


A. Radermacher
CEA


P.-A. Gussander
Nigus


O. Gürca
CEA


C. Dumoulin
Crystal


M. Morelli
CEA


A. Cuccuru
CEA


E. Posse
Zeligsoft


Q. Le Menez
CEA

S. Li
CEA

M. Adam
All4Tec

M. Adam
All4Tec

M. Elaasar
Crossplatform Sw

S. Gabel
Esterel
Tech.

M. E. Alferez Salinas
CEA

A. Cucurru
CEA

B. Selic
Malina Software Corp.

J. Cadavid
CEA

Let's see a short live demo of Papyrus-UML.

FROM UML TO DSML…

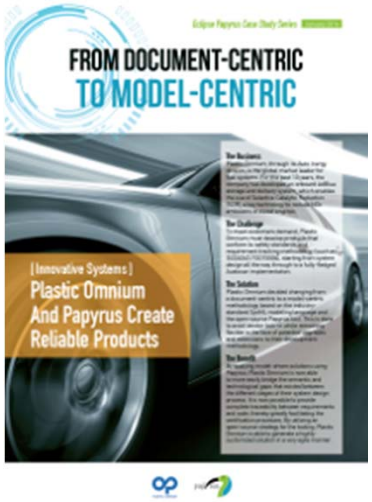
- **Originally intended for modeling software-intensive systems:**
  - UML models capture different views of a software system (e.g., data structure, run-time behavior, packaging and deployment)
  - Inspired primarily by the concepts from object-oriented languages (class, operation, object, etc.) but now supporting various development paradigms (e.g., service-oriented, component-based, functional-oriented design styles).

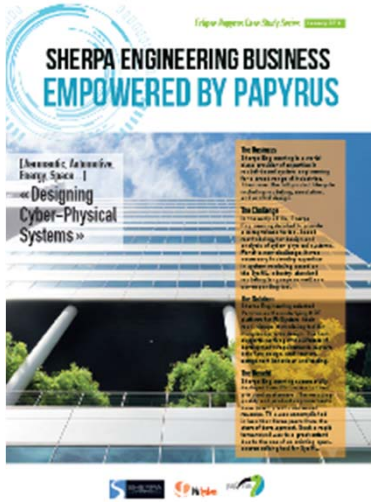
- **However, the general nature of its concepts make UML2 suitable for extensions to specific modeling domains.**
  - Domain Specific Modeling Language by profiling the UML2!
    - E.g., MARTE and SysML.
  - If too large, UML can also be pruned (via OCL Consraints)
    - "Use only what you need"
  - If not enough, UML can be extended (via UML stereotypes)
  - Enable MDE in a multidisciplinary context
    - UML profiles may be composed (e.g., system engineering and safety analysis)

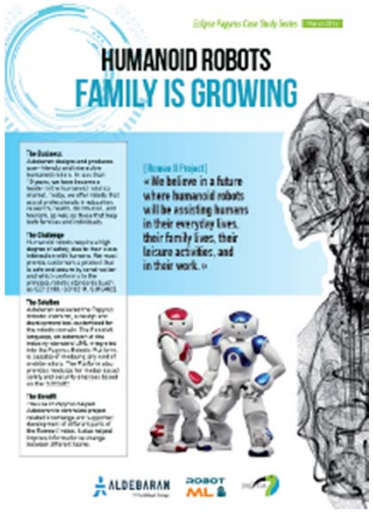
Let's see a short live demo of Papyrus-Toolsmith.

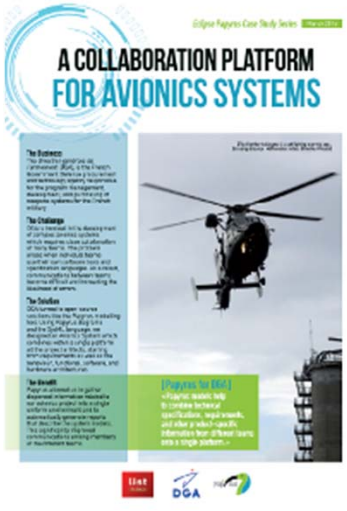
FOR KNOWING ABOUT INDUSTRIALS USE CASE STORIES AND USAGE TESTIMONIALS, VISIT:

WWW.ECLIPSE.ORG/PAPYRUS/TESTIMONIALS.HTML









EXPECTED EXTRENAL CONTRIBUTIONS

User Experience

And all the other things we never thought of…

Robustness
>> DevOps <<

Customizability
& configurability

How to contribute ?

Papyrus Wiki	Papyrus Forum	Dev Mailing List	YouTube Channel
Papyrus Git	Papyrus Gerrit	Papyrus Hudson	Papyrus Bugzilla

→ www.eclipse.org/papyrus/community.html ←

ABOUT THE AGENDA

- Architecture and what & why MBE
  - Outline architecture concern, then introduces and defines MBE and explains its value add propositions.
  - Impact studies: a selective summary of published results of industrial use of MBE.
- How to enable model-driven engineering?
  - How to introduce MBE into a development organization.
  - A general overview of MBE tools and related industry trends with special focus on open source tooling.
- **And what about MBE for mission critical, realtime embedded software engineering?**

Is MDA suitable mission critical & realtime embedded software design?

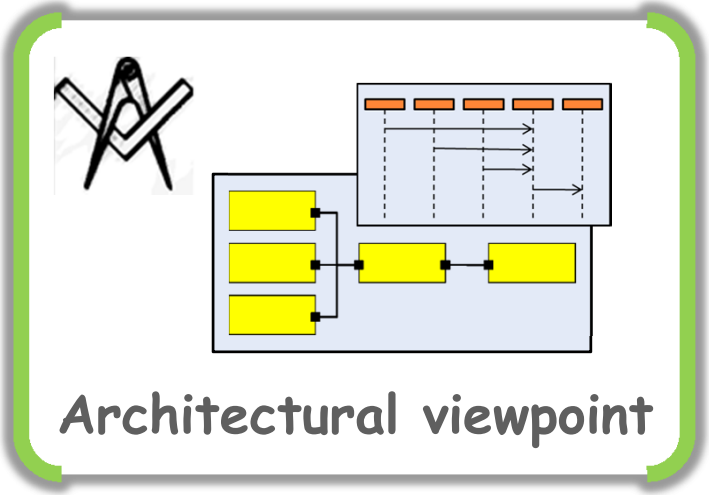
Sidebar

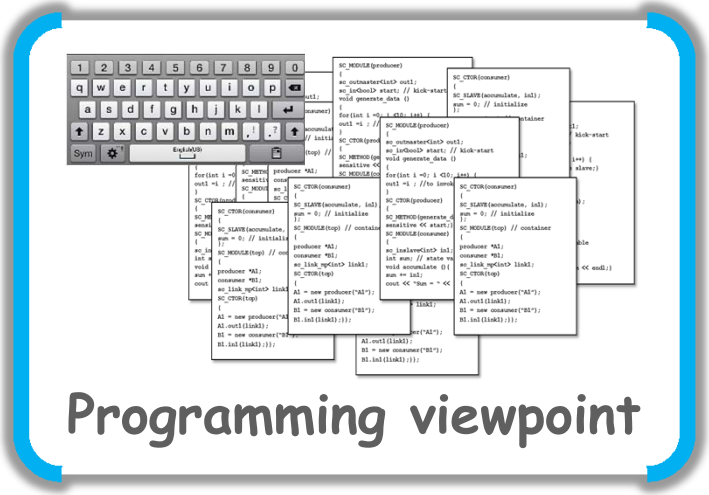
Model Driven Architecture (MDA) is a comprehensive set of OMG standards in support of MBE: UML, SysML, QvT, etc.

ABOUT A TYPICAL SOFTWARE PROJECT CONTEXT

- Software architects define the global vision and the organization of the work to be done (what has to be done and which technologies have to be used).

- Software engineers (programmers) implement the plan usually write code with their favorite language and IDE.





FOR SOFTWARE MODELING AND AS A STANDARD, UML IS A (THE?) GOOD CHOICE.

- **Mature modeling language**
  - Initially based on very experienced modeling language designers: the three amigos, Booch, Jackobson and Rumbaugh but also Coleman, Desfray, Embley, Gamma, Harel, Meyer, Odell, Selic, Shaer-Mellor and Wirfs-Brock.
  - A 20 year old modeling languages (current version:2.5) continually maintained and updated by very advanced experts coming from various origin: end users, tool providers and academics.
- **A rich modeling languages covering:**
  - All main development paradigms (e.g., OO, CBSE, SOA, or Procedural)
  - A large set of concerns (e.g., architecture description, automata, data-flow, scenario or use case).
- **Internationally popular and in-use**
  - UML is widely educated, disseminated and implemented…

…all around the world.

FOSTER SOFTWARE ARCHITECT AND PROGRAMMER COLLABORATION !

- **Reticence of MDE adoption in industry [1] because:**
  - Related controversy: diagram-based versus textual-based languages?
    - Software architects favor the use of graphical modeling languages
    - Software programmers prefer textual programming languages
- **A real need for enabling full model-code synchronization**
  - Industrial need: update model or code to deal with co-evolution [1]
    - 70 % update models (or not!)
    - 35 % update code and spend a lot of time to synchronize models and code

    ➔ Majority of people said that keeping model & code synchronized is critical to the successful use of MBE
  - Scientific research directions:
    - Need for an efficient support enabling switching in real-time from architecture description to implementation views and vice versa [2]
    - Need for dealing with model-code consistency [3]

The solution is called round-trip engineering. •••

[1] J. Hutchinson et al., "Model-driven engineering practices in industry", Sci. Comput. Program, 2014.
[2] Taylor, et al., "Software design and architecture the once and future focus of software engineering", FOSE'07, 2007.
[3] Zheng, et al., "A classification and rationalization of model-based software development." Software & Systems Modeling 12.4, 2013.

WHAT IS ROUND-TRIP ENGINEERING?

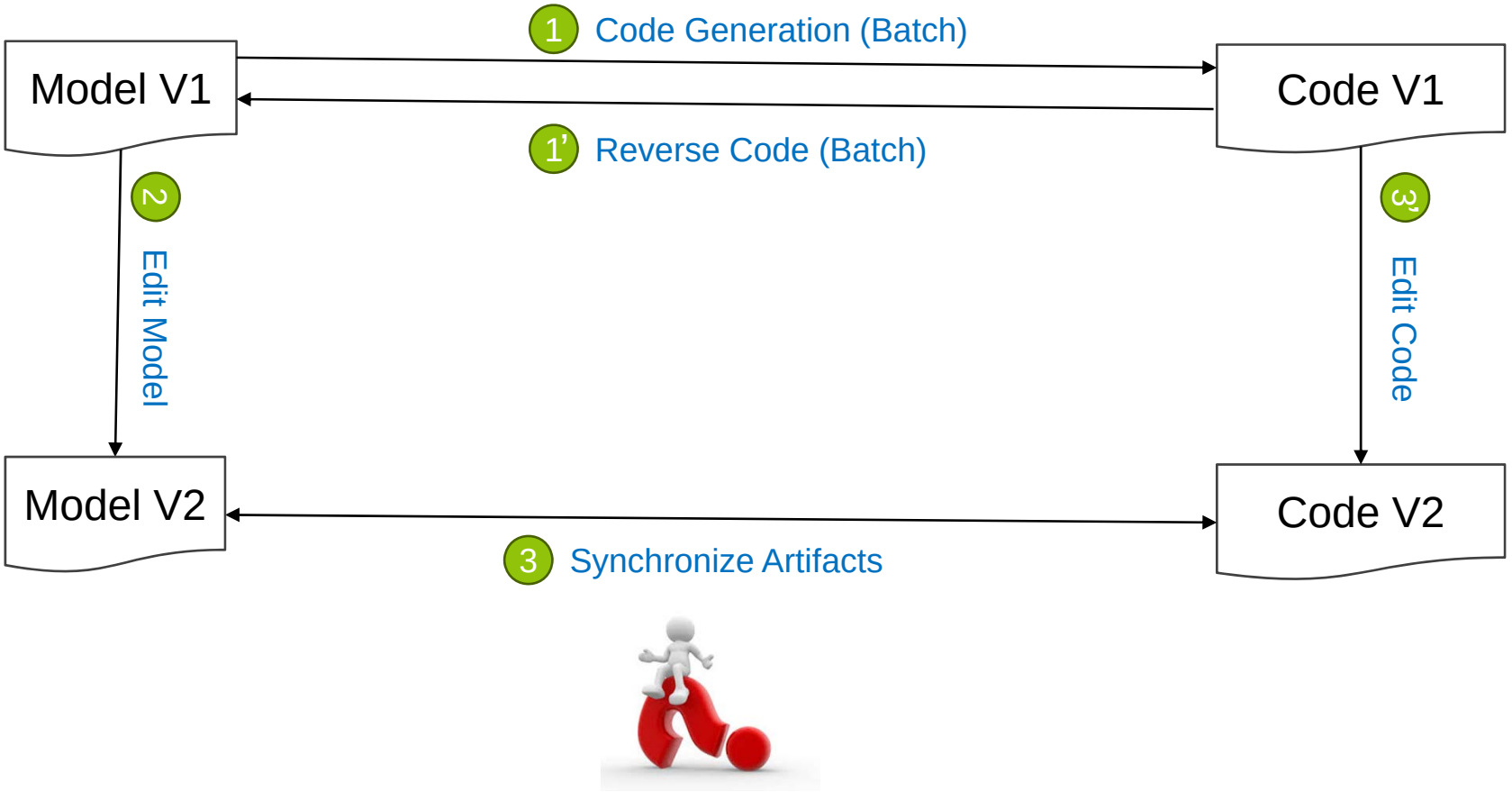
The ability to automatically maintain the consistency of multiple, changing software artifacts, in software development environments/tools [1].

- **Related to two traditional software engineering disciplines:**
  - Forward engineering: creating software from specifications
  - Reverse engineering: creating specifications from existing software
- **Round-trip engineering adds synchronization of existing artifacts that evolved concurrently by incrementally updating each artifact to propagate changes made to the other artifacts**

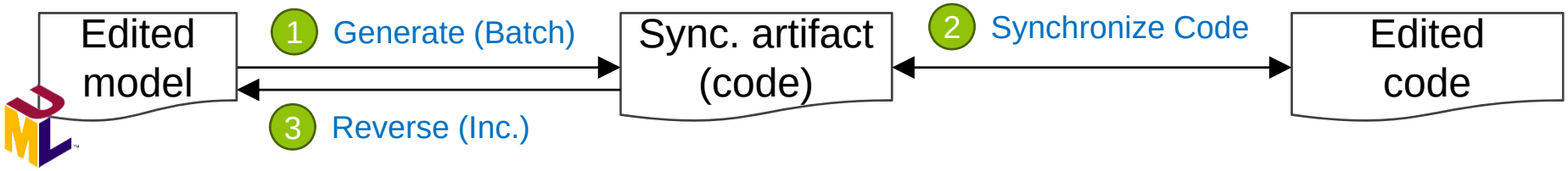
  ➔ Round-trip engineering generalizes hence both forward and reverse engineerings

[1] S. Sendall and J. Küster, "Taming model round-trip engineering", in Proceedings of Workshop on Best Practices for Model-Driven Software Development, Vancouver, Canada, 2004.

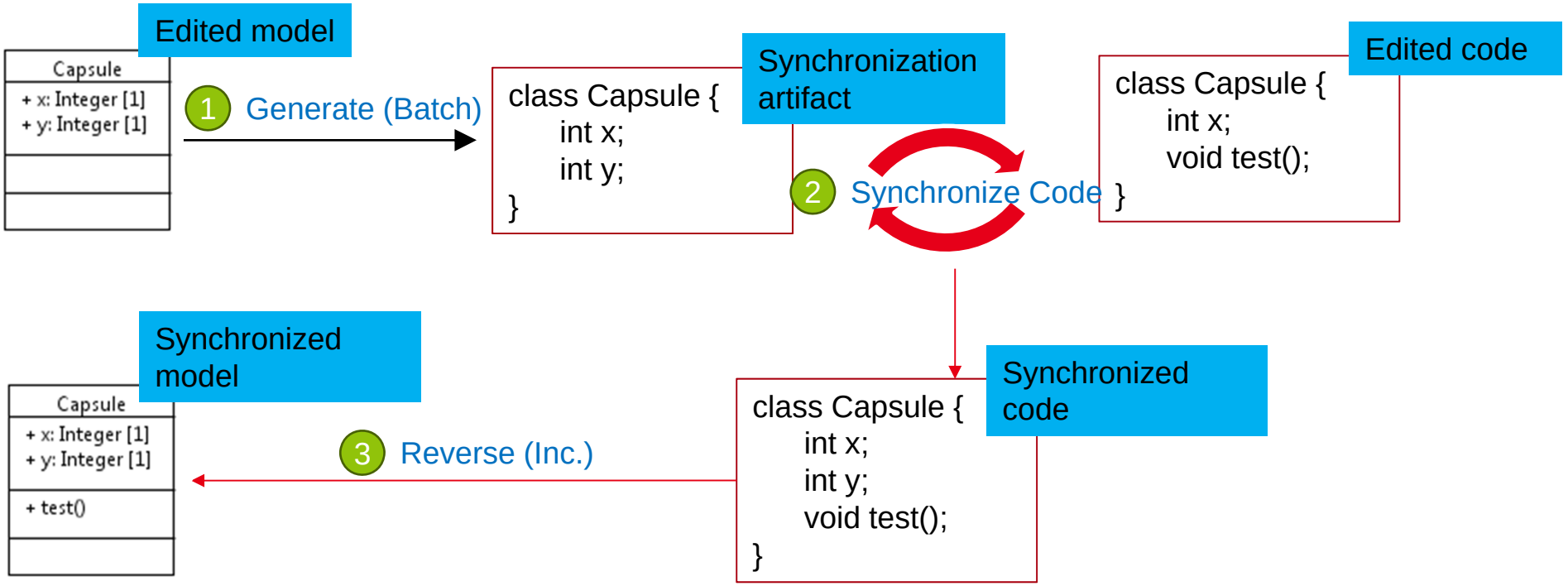
ROUND-TRIP ENGINEERING: CO-EVOLUTION ISSUES



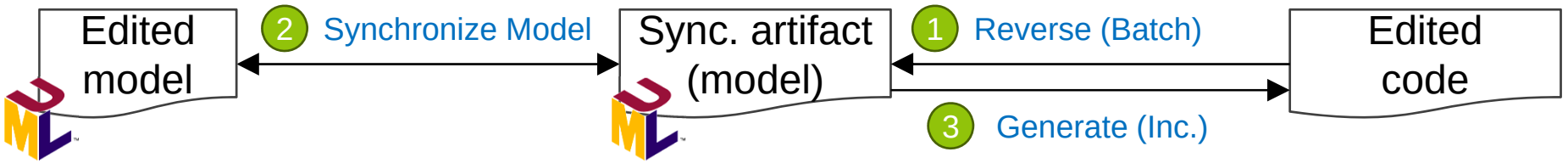
STRATEGY 1: SYNCHRONIZATION VIA CODE SYNCH-ARTIFACT



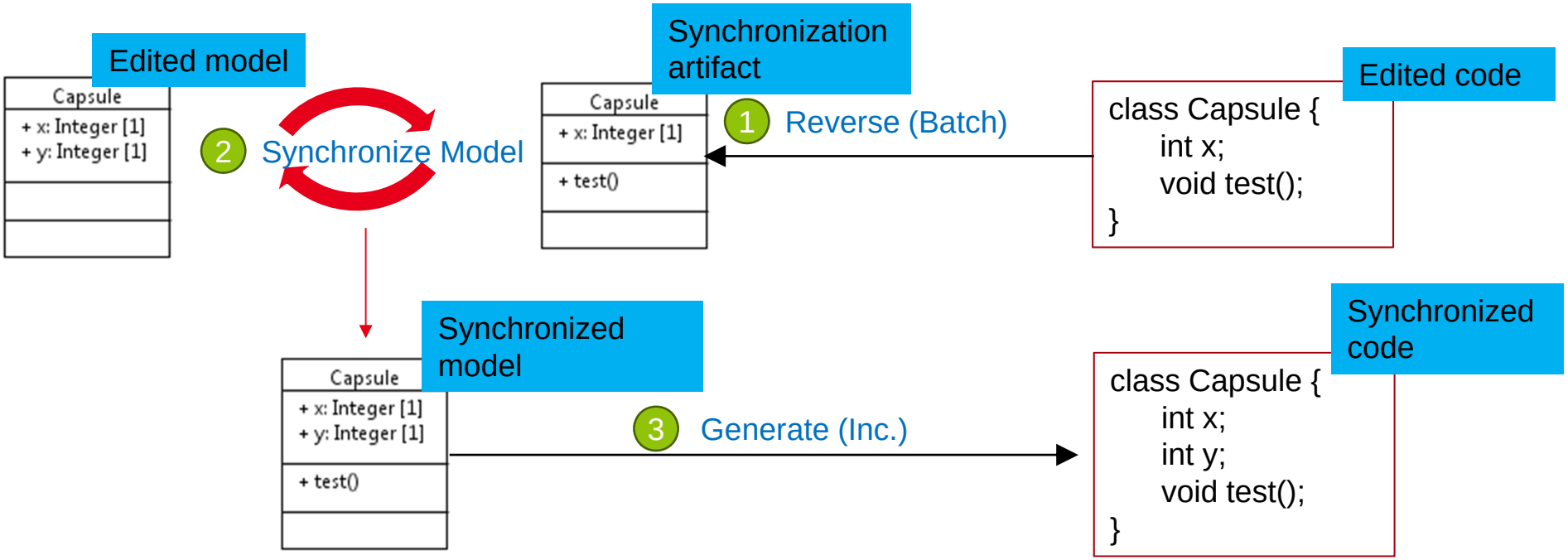
- Example



STRATEGY 2: SYNCHRONIZATION VIA A MODEL SYNCH ARTIFACT



- Example



Let's see a short demo of Papyrus-Roundtrip feature.

DE LA RECHERCHE À L'INDUSTRIE

cea

www.cea.fr

Papyrus C++ Roundtrip

December, 2015

Contributors: Van Cam Pham, Shuai Li, Ansgar Radamacher (CEA LIST)

INSTITUT CARNOT CEA LIST

université PARIS-SACLAY

Is MDA suitable for mission critical & realtime embedded software design?

A STANDARD DSML FOR RTES MODELLING

- **The rationale for UML:**
  - In 80's, too many custom approaches, languages and tools…

Need to unify modeling languages around a unique, common and shared language: UML
"not replace them, just aggregate, integrate and support them"

- **For real-time systems, a similar issue:**
  - Too many custom approaches, languages and tools…
  - Often complex access to related advanced-technologies

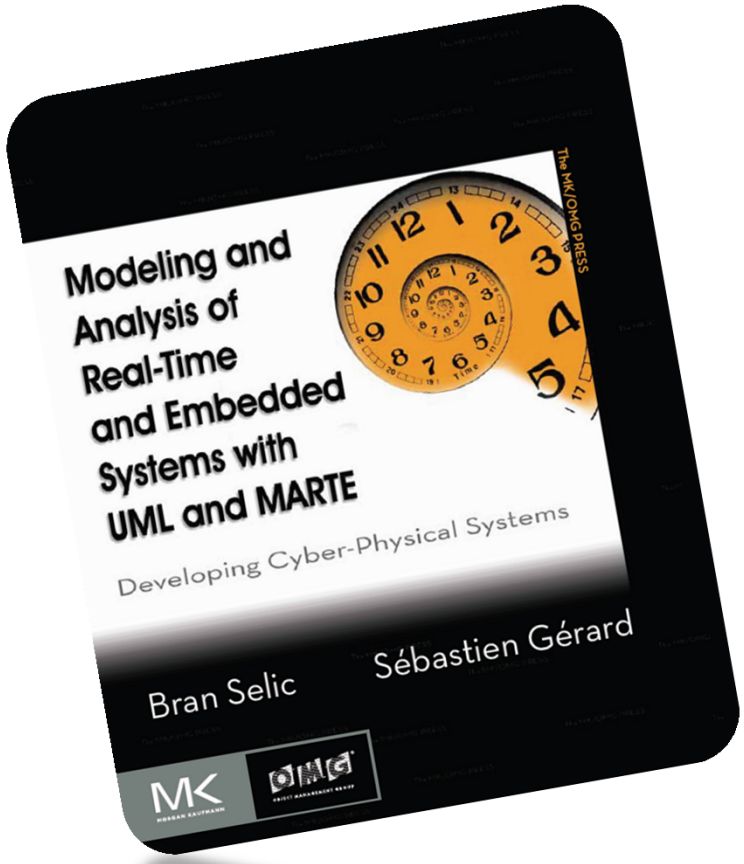
Need to unify modeling languages around a unique, common and shared language: MARTE
"not replace them, just aggregate, integrate and foster their usages"

WHAT DOES MARTE ADD TO UML?

- **A domain-specific modeling language for modeling real-time, embedded, and cyber-physical systems**
  - RTE applications, platforms, and relationships between them
- **Support for precise specifications of quality of service (QoS) characteristics**
  - Specifying physical dimensions and corresponding values
  - E.g., delays, bandwidths, memory sizes, CPU speeds, energy consumption, etc.
- **A generic framework for certain types of quantitative analyses of UML models**
  - Including two specific specializations (schedulability analysis and performance analysis)
  - Suited to computer-based automation support

FORTUNATELY, HELP IS ON HAND (A CHEEKY PRODUCT PLUG)

- Available in a web page/bookstore near you:



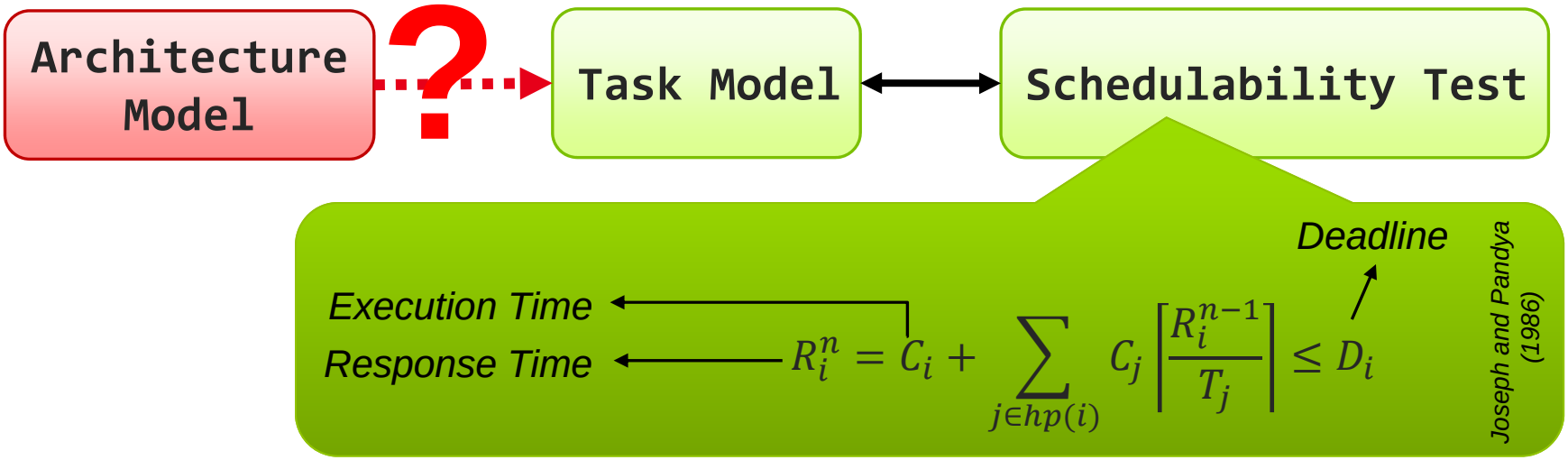
Publisher: Morgan Kaufmann
ISBN: 978-0-12-416619-6

SIDEBAR: RTES NEEDS FOR SCHEDULABILITY ANALYSIS

- **RTES usually implemented as a multi-tasking system**
  - = Concurrent tasks having deadlines and interacting

  ➔ Scheduling = method by which tasks are given access to processors, i.e. according to a scheduling policy

- **Schedulability analysis**
  - Verify that tasks meet their deadlines, when executing on limited processors, according to a scheduling policy ➔ verify schedulability



SCHEDULABILITY ANALYSIS WITH PAPYRUS SOFTWARE DESIGNER

- **Seamless analysis process:**
  - Integrated MARTE modeler with UI facilities to create a model for schedulability analysis
  - Reporting of results with charts and UI menus
  - Automatic completion of MARTE model with analysis results
- **Several implemented schedulability tests:**
  - Rate-Monotonic Analysis (RMA) for monoprocessor
  - Tindell's offset-based test for partitioned multiprocessor systems
  -  Redell's improved offset-based test (in development)
- **API to extend tool with new tests:**
  - Transformation to task models of existing schedulability analysis tools
  -  Extension mechanisms to add task models and schedulability tests to Papyrus Software Designer directly (in development)

DE LA RECHERCHE À L'INDUSTRIE

cea

www.cea.fr

Papyrus Software Designer Schedulability Analysis Example

September, 2016

Contributors: Chokri Mraidha, Florian Noyrit, Shuai Li, Sébastien Gérard (CEA LIST)

INSTITUT CARNOT CEA LIST

université PARIS-SACLAY

Is MDA suitable for mission critical & realtime embedded software design?

REMEMBER THAT…

MBE RELIES ON TWO FOUNDATIONAL PILLARS…

Abstraction

Suitable and sound modeling language engineering

Suitable and sound modeling language engineering

Automation

Efficient and scalable computer-aided engineering

Efficient and scalable computer-aided engineering

WHAT ABOUT MDE FOR CRITICAL MISSION SYSTEMS?

Suitable and ~~sound~~ **formal** modeling language engineering

Efficient and scalable computer-aided **analysis and simulation engineering**

TOWARDS MORE FORMAL MDE: OMG TRENDS
LONG-TERM VISION: A SUITE OF SPECIFICATIONS

(source: executable uml roadmap http://www.omg.org/members/cgi-bin/doc?ad/14-09-06.pptx)

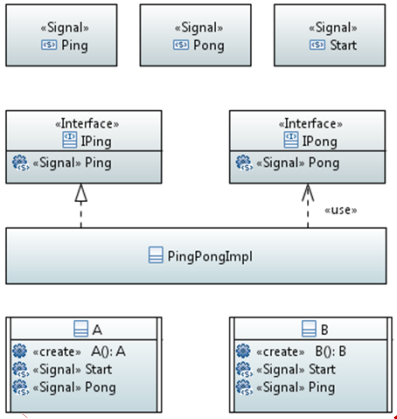
- **Improve specification description**
  - **UML 2.5**: Complete revision of its text description to simplify its presentation and disambiguate as much as possible its semantics.
- **Enable text-based specification => Alf**
  - Textual surface representation for UML modeling elements with the primary purpose of acting as the surface notation for specifying executable (fUML) behaviors within an overall graphical UML model.
  - Also provides an extended textual notation for structural modeling within the fUML subset.
- **Towards a formal semantics of UML**
  - **fUML**: Foundational UML is an executable subset of standard UML with formal/operational semantics.
  - **PSCS**: Precise Semantics of UML Composite Structure. Extension of fUML for composite structure modeling and execution
  - **PSSM**: Precise Semantics of UML State "Machines. Extension of fUML/PSCS for state machine modeling and execution

fUML EXAMPLE

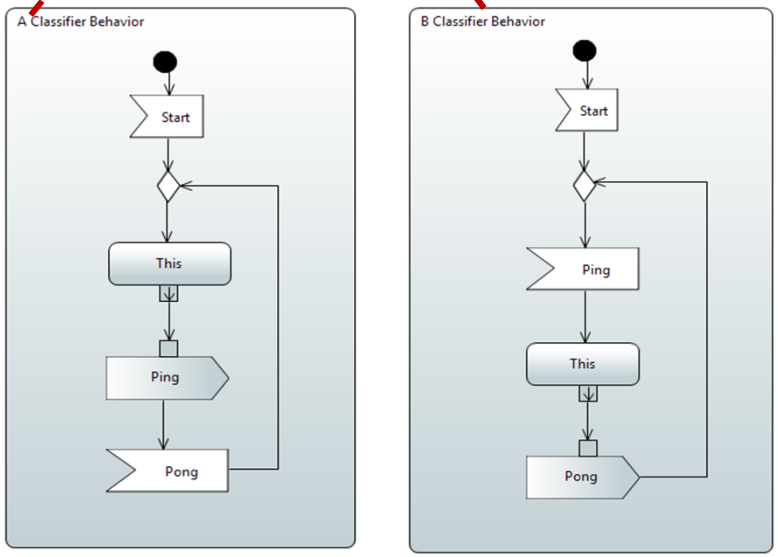
Structure

Behavior

1. Subset of class diagram



3. Subset of activity diagram

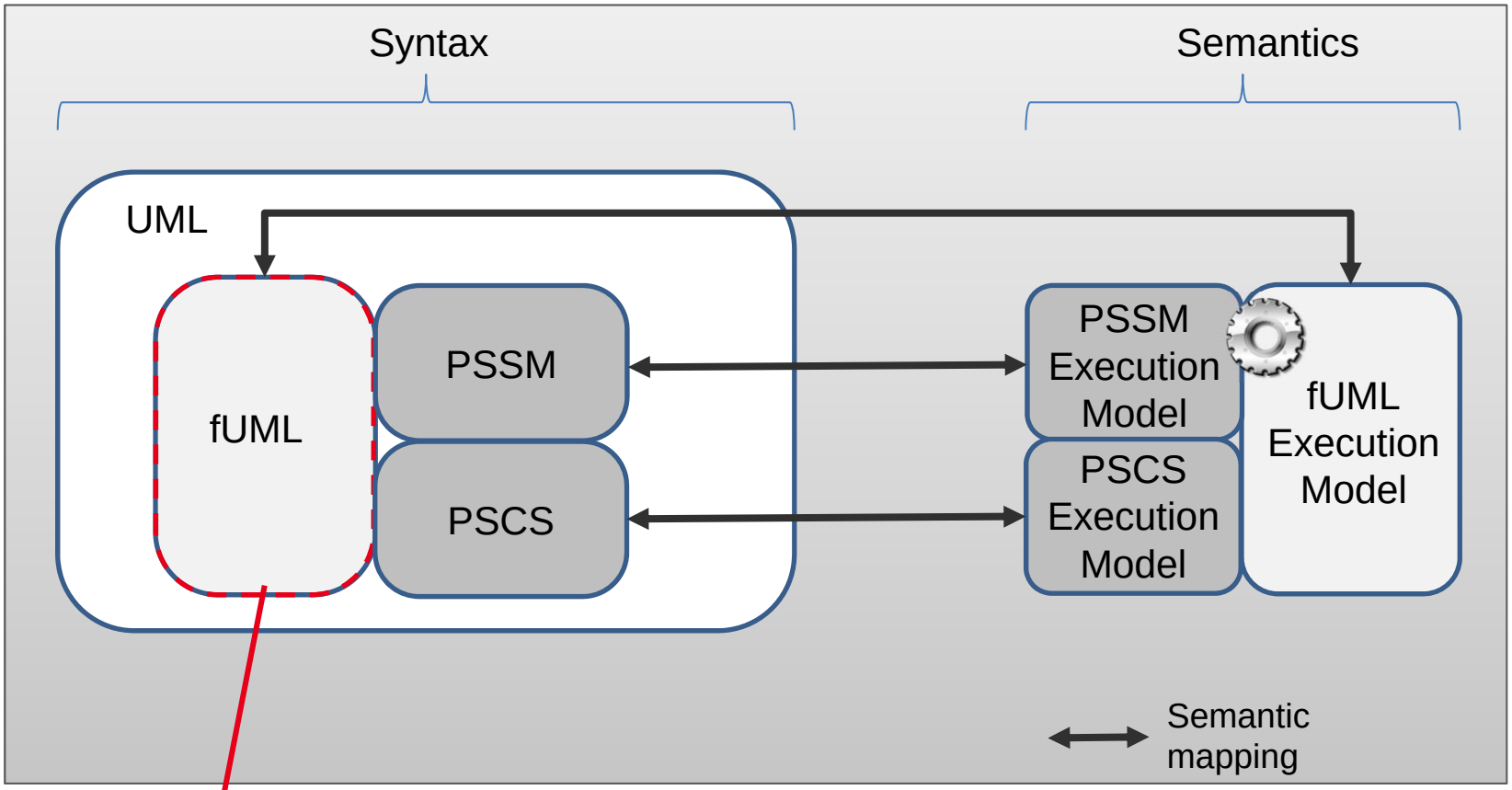


Alf (Action Language for fUML):
= Textual surface notation for the fUML subset.

```
activity LaunchPingPongExample() {
  game = new Game() ;
  game.b.Start() ;
  game.a.Start() ;
}
```

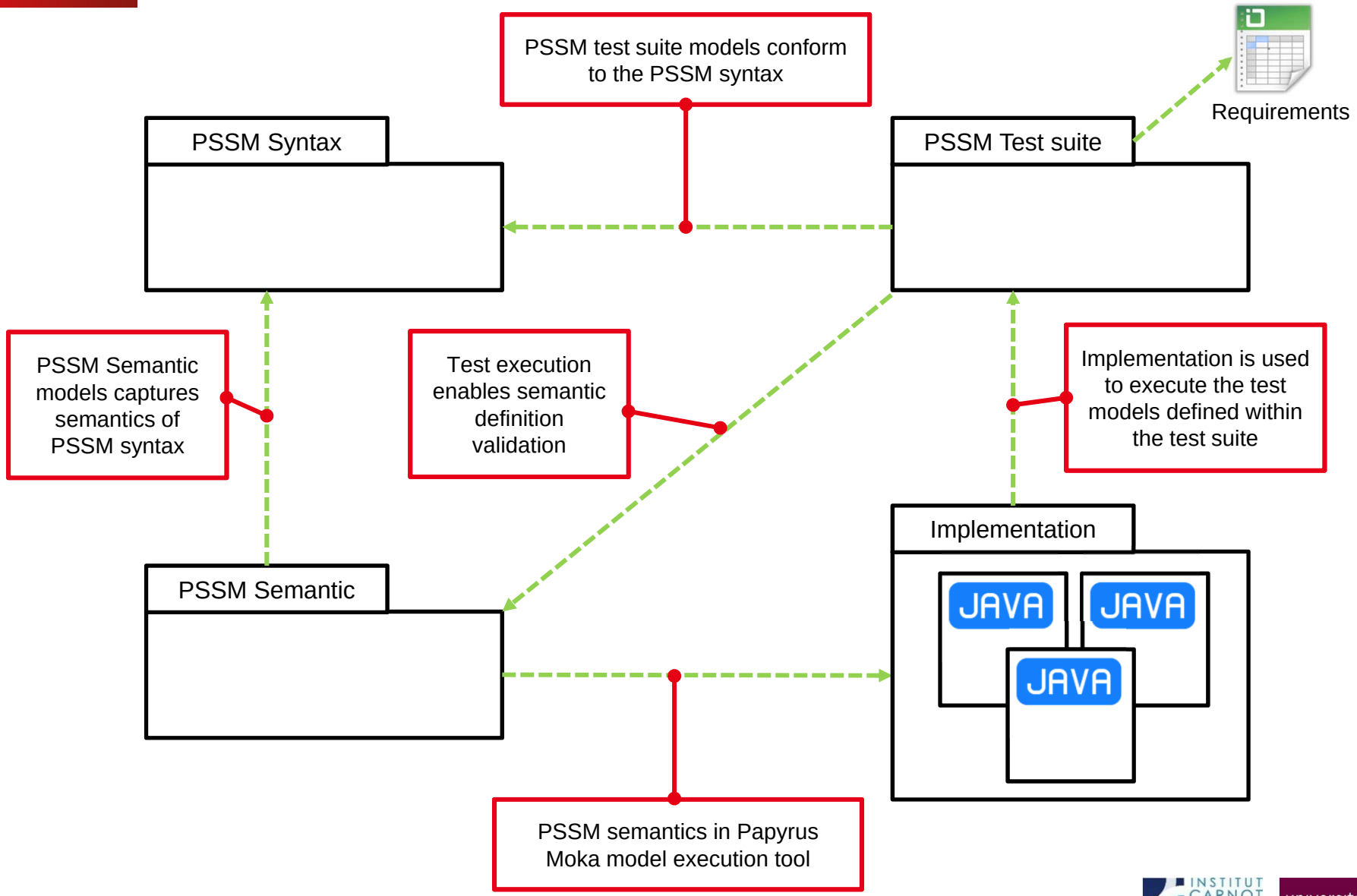
4. Alf Specification of an Activity

EXECUTABLE UML OMG SPECIFICATIONS



Alf (Action Language for fUML):
- Textual surface notation for the fUML subset

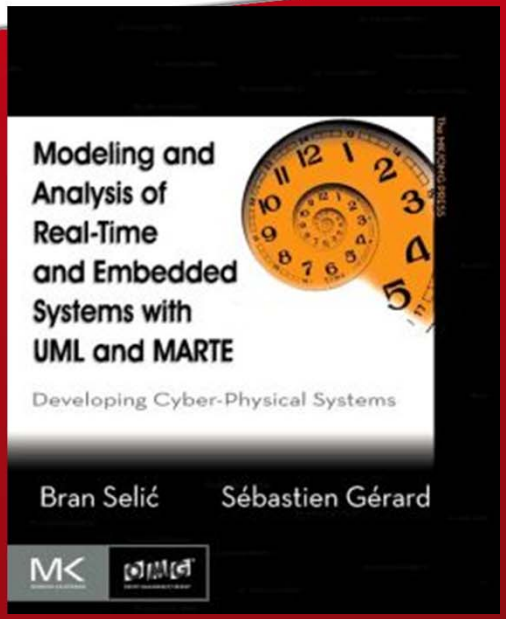
PSSM SUBMISSION STRUCTURE



Papyrus is the official open-source Eclipse UML2 modeling tool:

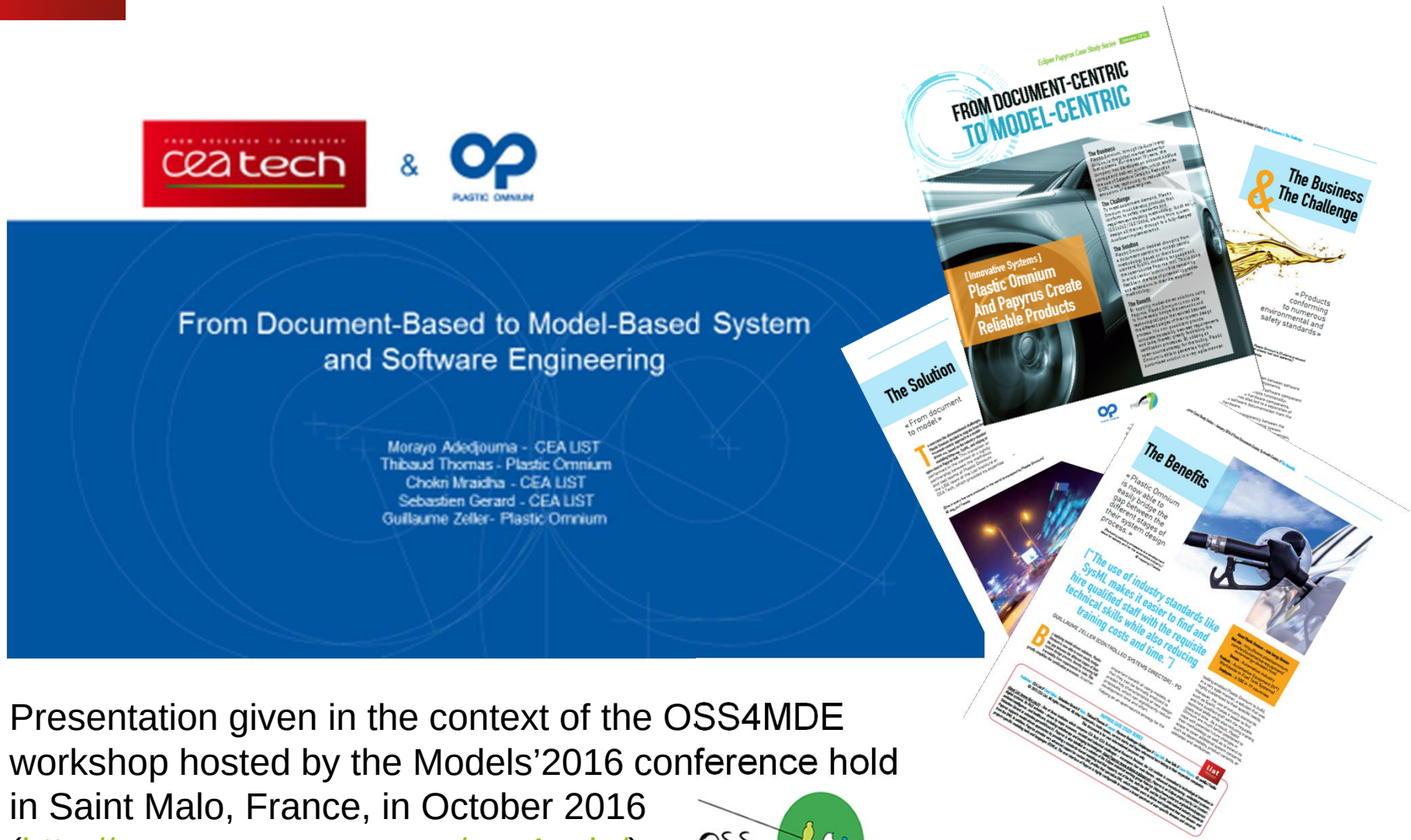
www.eclipse.org/papyrus

- Papyrus provides a complete graphical editor for both UML and SysML standards based on the MDT::UML2 component for its repository.
- Papyrus addresses the two key features expected from a UML2 graphical editor: modeling and profiling.
- Papyrus is highly customizable and extensible enabling DSML definitions based on standard UML profiles!
- Papyrus provides a support to MARTE 1.1 (including a rich text editor for VSL).



Commissariat à l'énergie atomique et aux énergies alternatives
Institut List | CEA SACLAY NANO-INNOV | BAT. 861 – PC142
91191 Gif-sur-Yvette Cedex - FRANCE
www-list.cea.fr

USE CASE STORY: MDE & PAPYRUS @ PO-INERGY



Presentation given in the context of the OSS4MDE workshop hosted by the Models'2016 conference hold in Saint Malo, France, in October 2016 (http://mase.cs.queensu.ca/oss4mde/).

USE CASE STORY: MDE & PAPYRUS @ PO-INERGY

CONTENTS

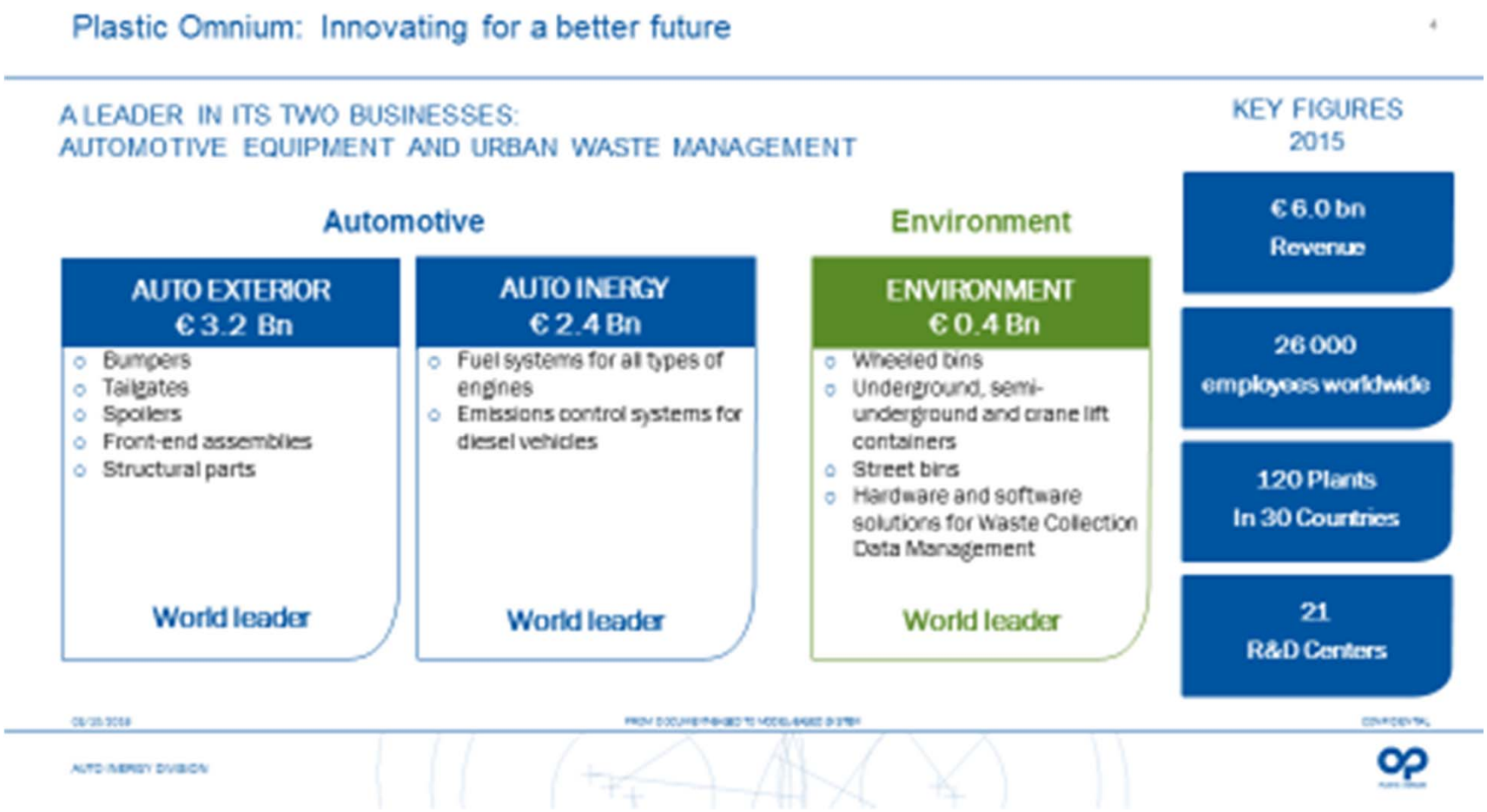
- Context and problem statement
- Solution: Model Based System Engineering
- Perspectives

Presentation given in the context of the OSS4MDE workshop hosted by the Models'2016 conference hold in Saint Malo France, in October 2016 (http://mase.cs.queensu.ca/oss4mde/).

USE CASE STORY: MDE & PAPYRUS @ PO-INERGY

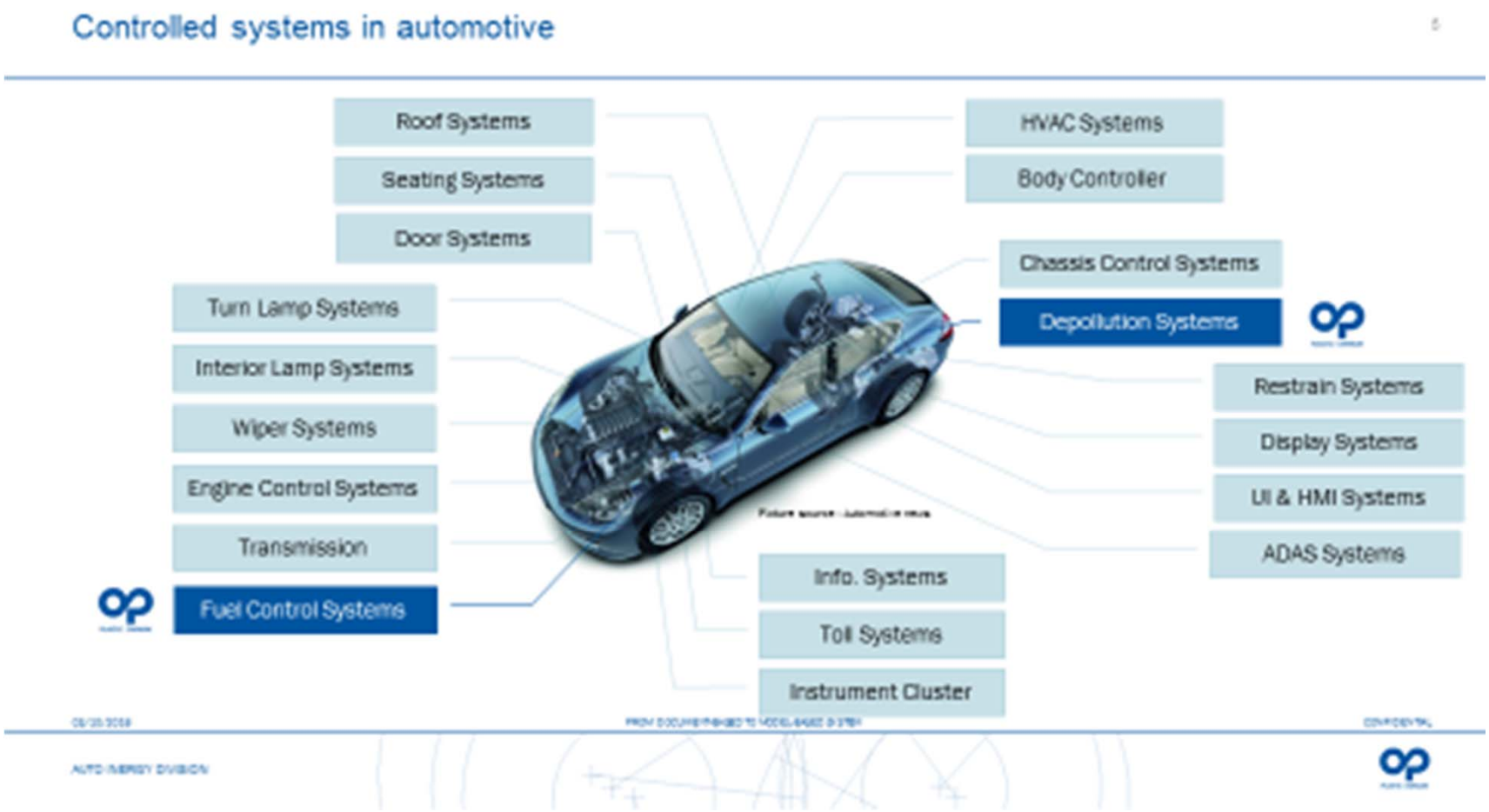
Presentation given in the context of the OSS4MDE workshop hosted by the Models'2016 conference hold in Saint Malo France, in October 2016 (http://mase.cs.queensu.ca/oss4mde/).

USE CASE STORY: MDE & PAPYRUS @ PO-INERGY



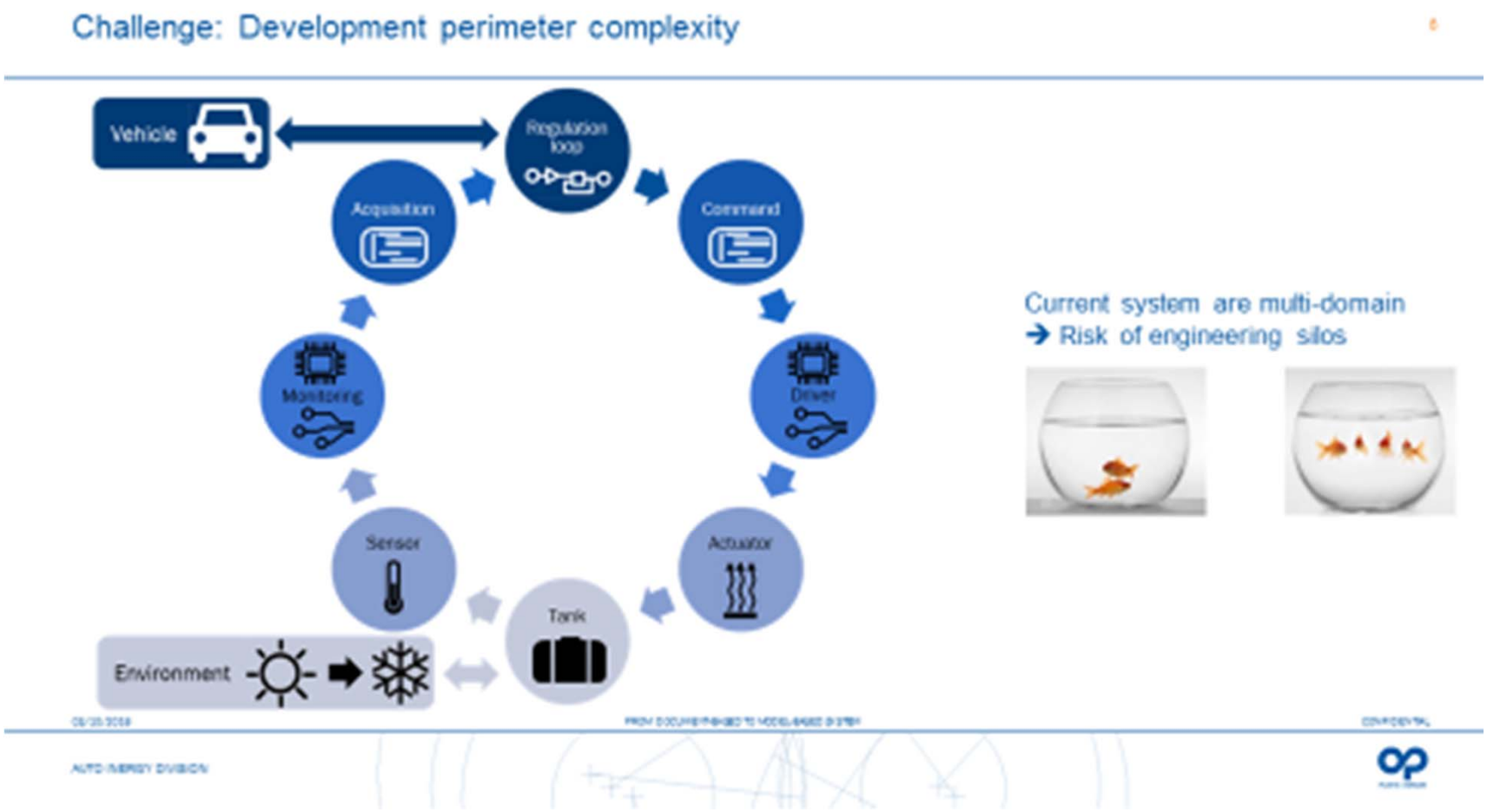
Presentation given in the context of the OSS4MDE workshop hosted by the Models'2016 conference hold in Saint Malo France, in October 2016 (http://mase.cs.queensu.ca/oss4mde/).

USE CASE STORY: MDE & PAPYRUS @ PO-INERGY



Presentation given in the context of the OSS4MDE workshop hosted by the Models'2016 conference hold in Saint Malo France, in October 2016 (http://mase.cs.queensu.ca/oss4mde/).

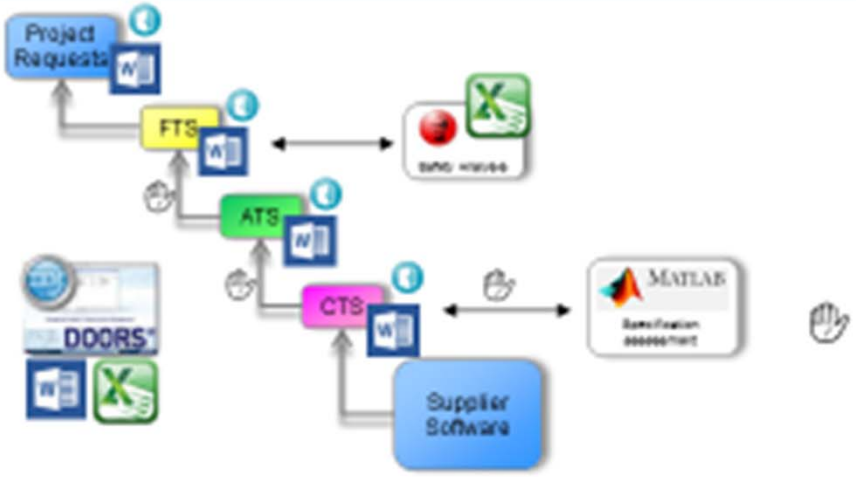
USE CASE STORY: MDE & PAPYRUS @ PO-INERGY



Presentation given in the context of the OSS4MDE workshop hosted by the Models'2016 conference hold in Saint Malo France, in October 2016 (http://mase.cs.queensu.ca/oss4mde/).

USE CASE STORY: MDE & PAPYRUS @ PO-INERGY

Challenge: Document centric development process



Starting point: document-centric

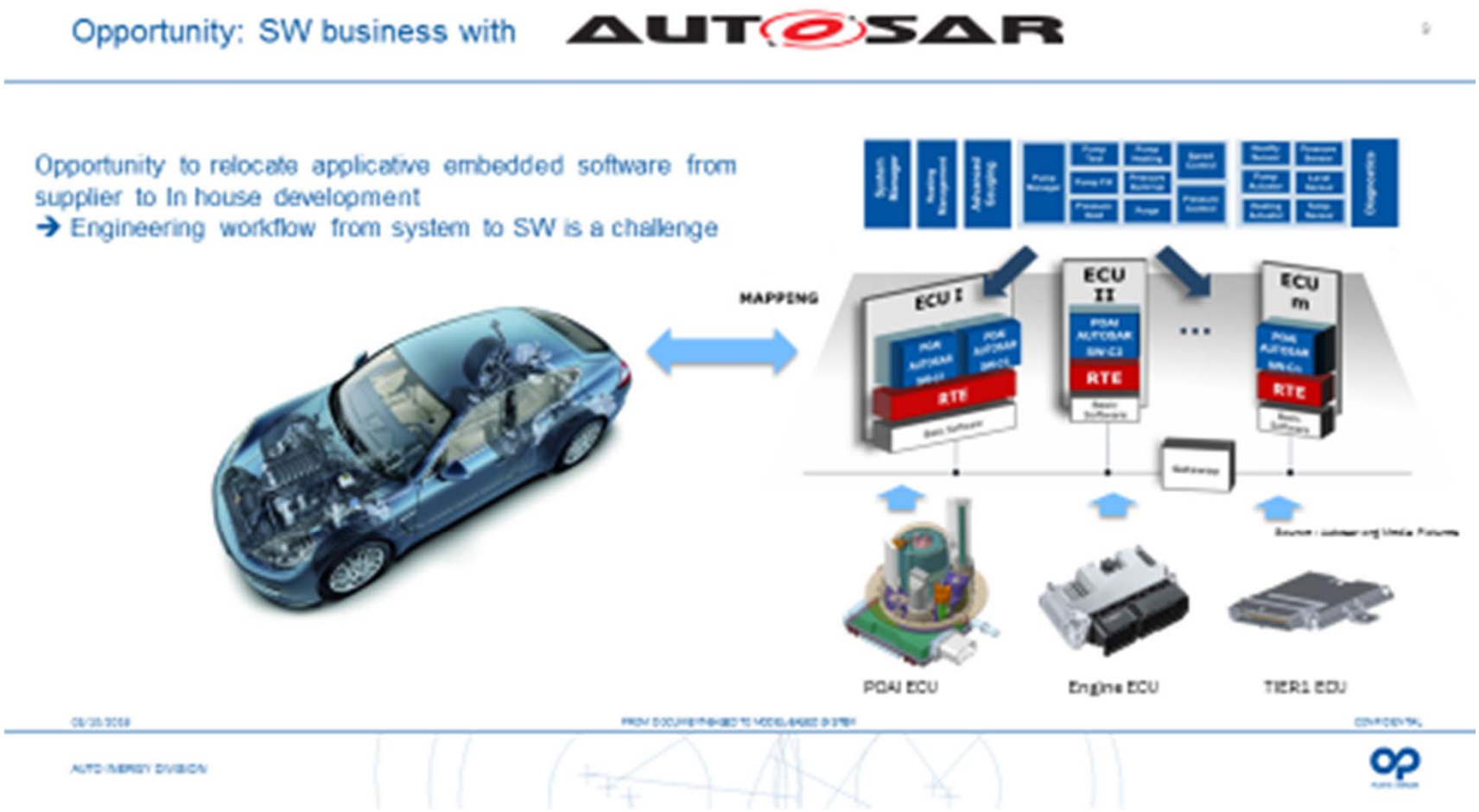
1. Initial methodological flow definition.
2. Formalization of requirements in DOORS (textual format with IDs).
3. Documents generation from DOORS.
4. Specification & Design with DOORS & Office tools!

Main issues

1. Textual syntax
   - Ambiguous
   - Not possible to exploit automatically
   - Not adapted for architecture description
2. Manual tasks
   - Error prone
   - High risk of inconsistencies
3. Duplications of information
   - Difficult to maintain
   - Not adapted for reuse
4. Traceability issues
   - Justification and certification problems

Presentation given in the context of the OSS4MDE workshop hosted by the Models'2016 conference hold in Saint Malo France, in October 2016 (http://mase.cs.queensu.ca/oss4mde/).

USE CASE STORY: MDE & PAPYRUS @ PO-INERGY

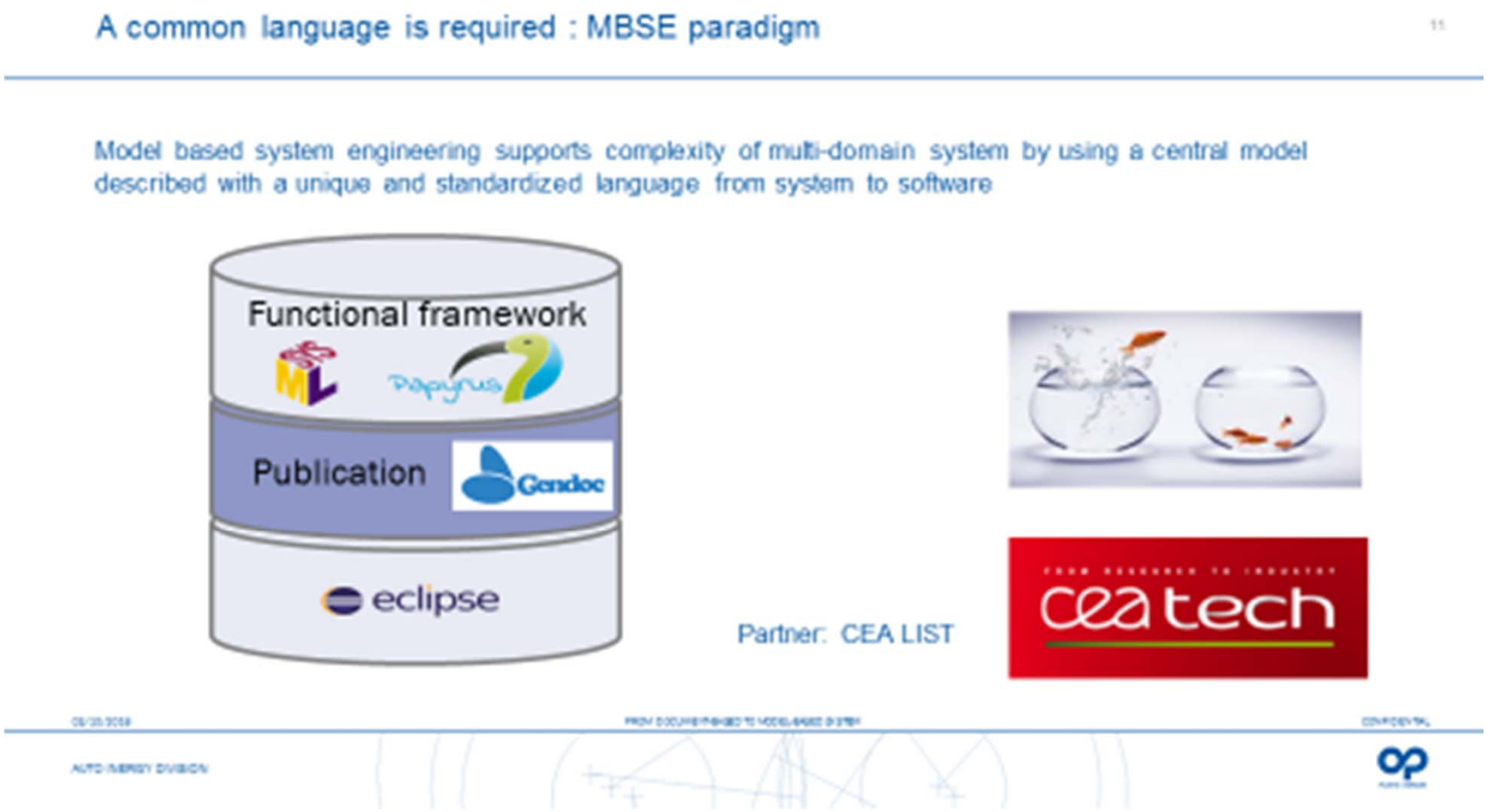


Presentation given in the context of the OSS4MDE workshop hosted by the Models'2016 conference hold in Saint Malo France, in October 2016 (http://mase.cs.queensu.ca/oss4mde/).

USE CASE STORY: MDE & PAPYRUS @ PO-INERGY

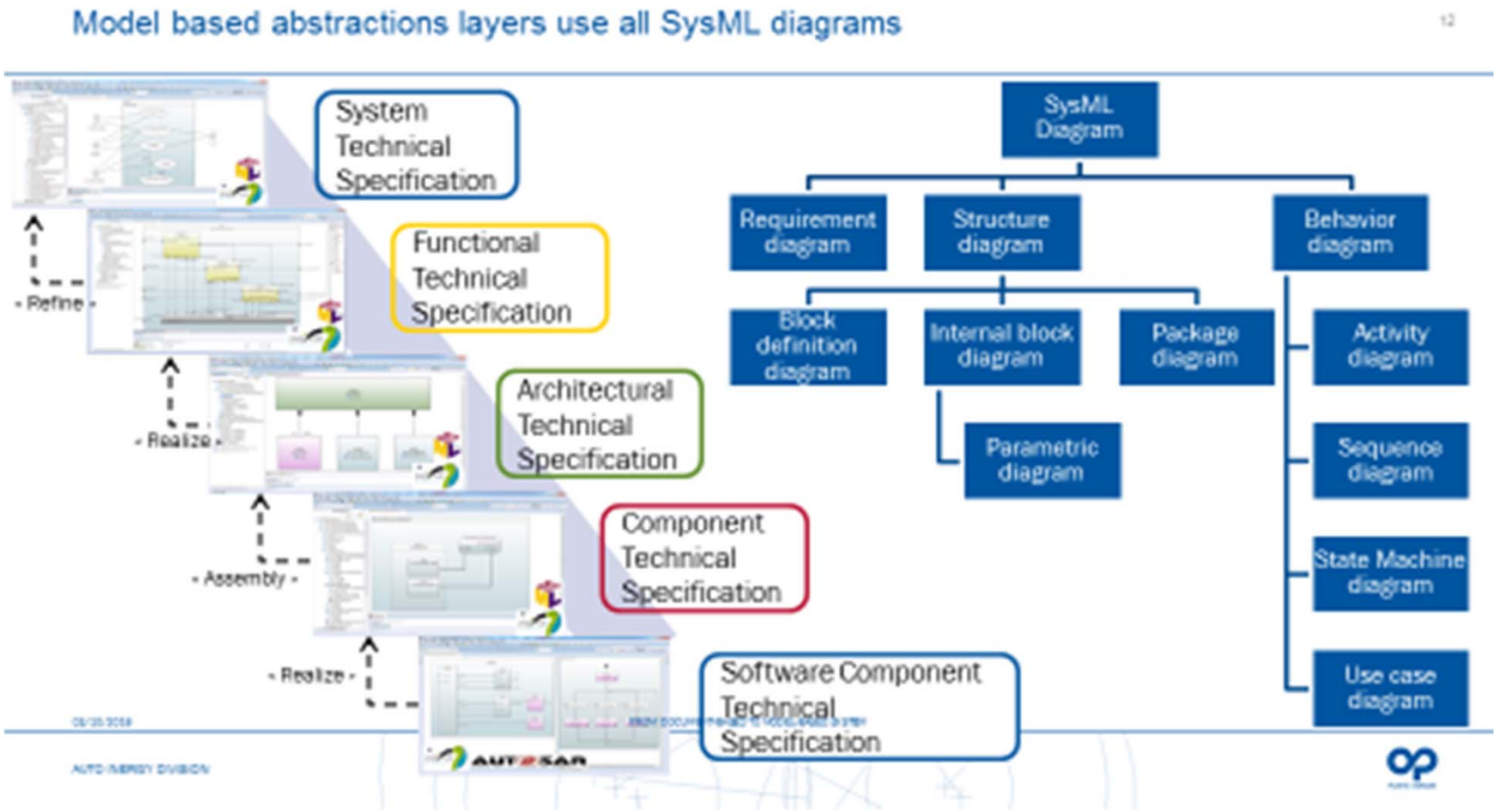
Presentation given in the context of the OSS4MDE workshop hosted by the Models'2016 conference hold in Saint Malo France, in October 2016 (http://mase.cs.queensu.ca/oss4mde/).

USE CASE STORY: MDE & PAPYRUS @ PO-INERGY



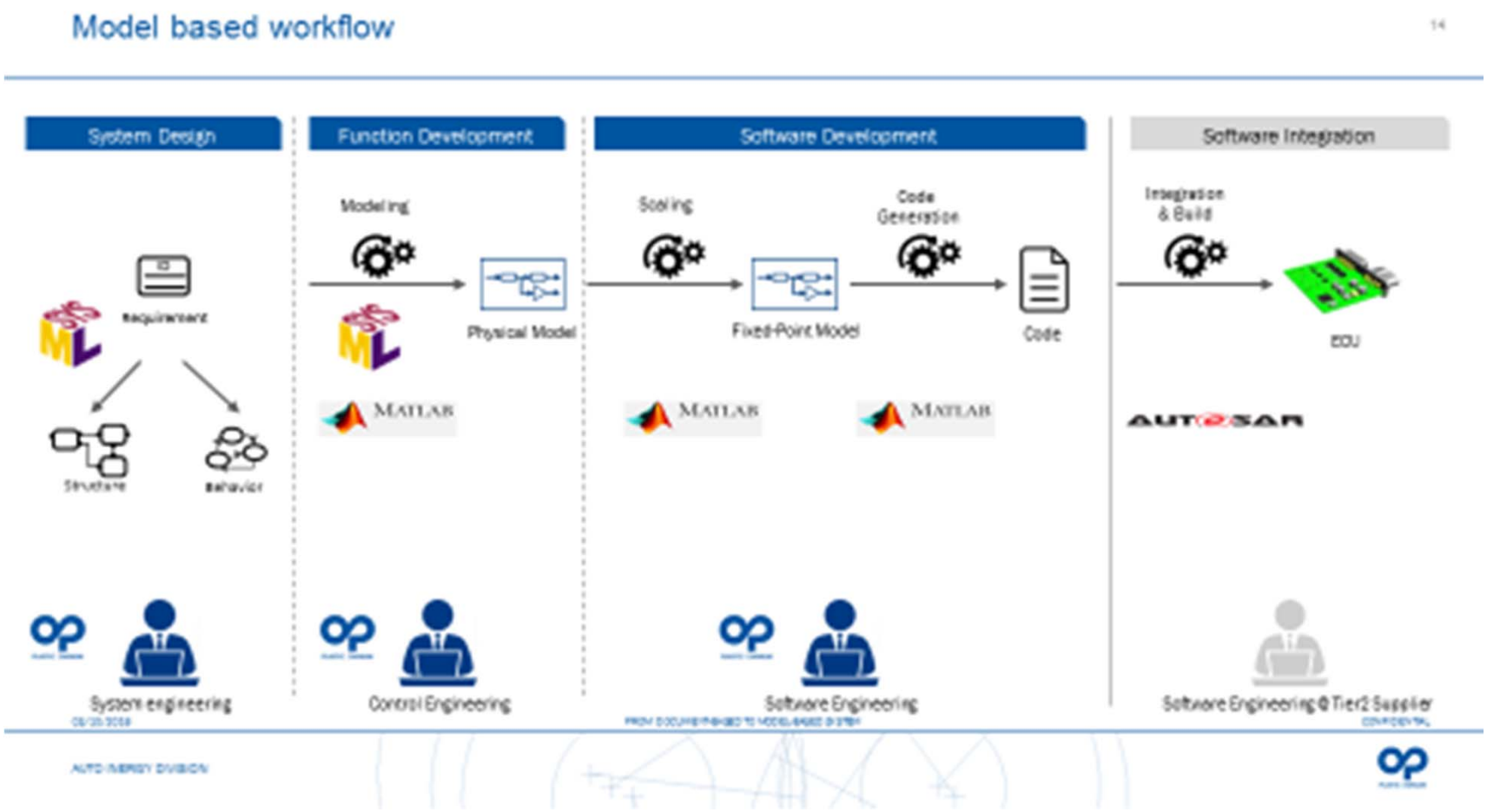
Presentation given in the context of the OSS4MDE workshop hosted by the Models'2016 conference hold in Saint Malo France, in October 2016 (http://mase.cs.queensu.ca/oss4mde/).

USE CASE STORY: MDE & PAPYRUS @ PO-INERGY



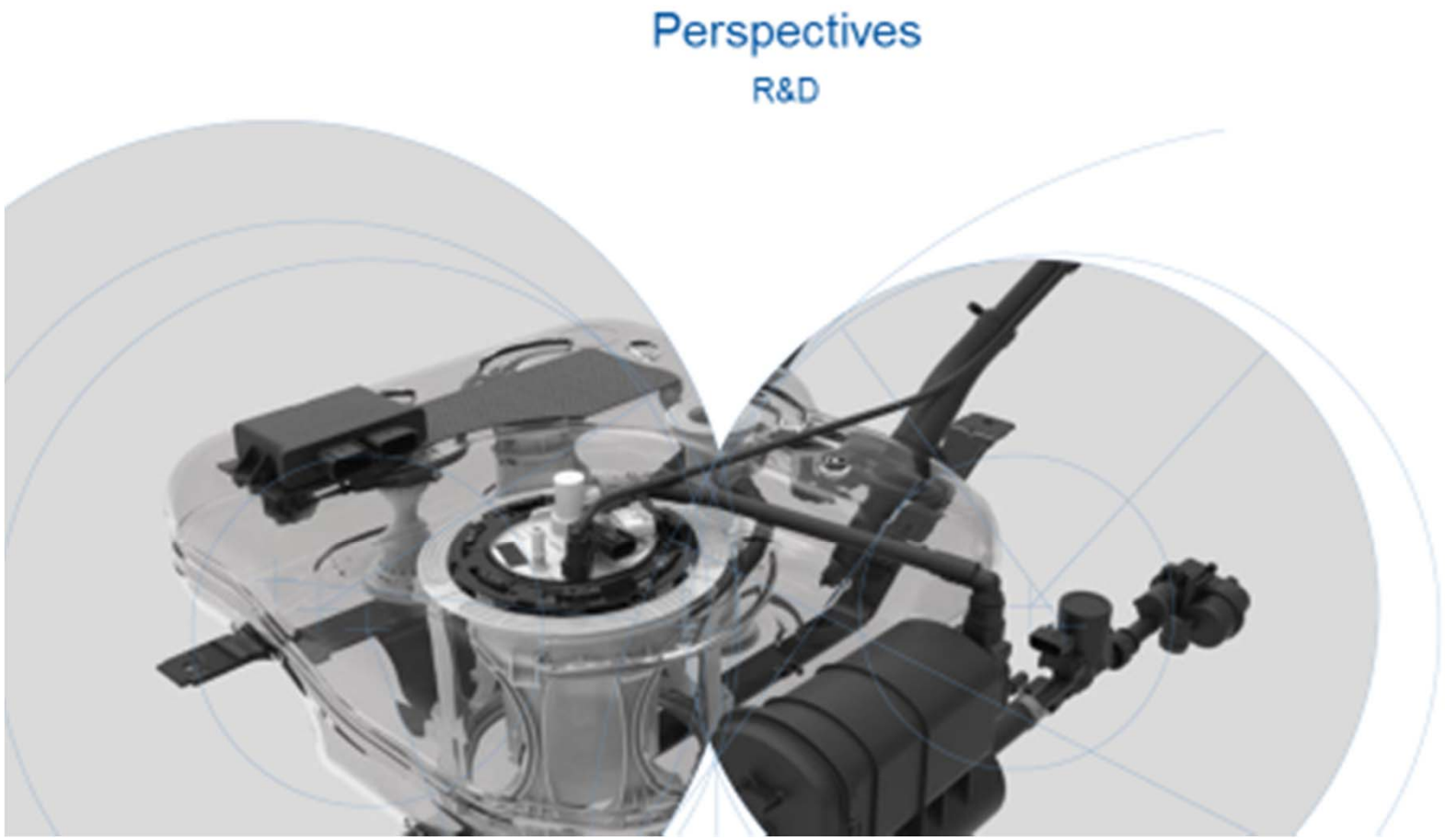
Presentation given in the context of the OSS4MDE workshop hosted by the Models'2016 conference hold in Saint Malo France, in October 2016 (http://mase.cs.queensu.ca/oss4mde/).

USE CASE STORY: MDE & PAPYRUS @ PO-INERGY



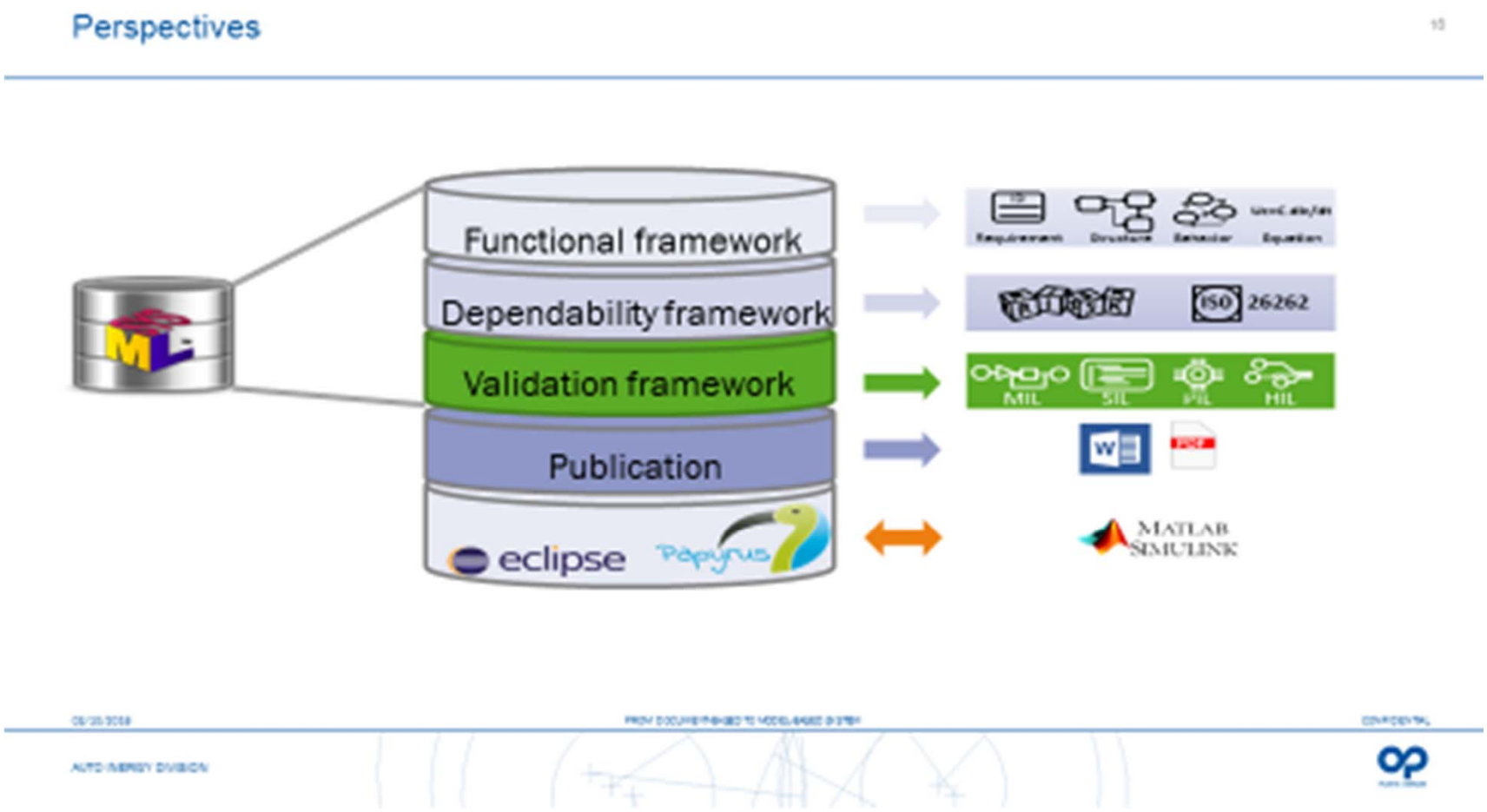
Presentation given in the context of the OSS4MDE workshop hosted by the Models'2016 conference hold in Saint Malo France, in October 2016 (http://mase.cs.queensu.ca/oss4mde/).

USE CASE STORY: MDE & PAPYRUS @ PO-INERGY



Presentation given in the context of the OSS4MDE workshop hosted by the Models'2016 conference hold in Saint Malo France, in October 2016 (http://mase.cs.queensu.ca/oss4mde/).

USE CASE STORY: MDE & PAPYRUS @ PO-INERGY



Presentation given in the context of the OSS4MDE workshop hosted by the Models'2016 conference hold in Saint Malo France, in October 2016 (http://mase.cs.queensu.ca/oss4mde/).