# Research-Based Innovation with Industry: Project Experience and Lessons Learned

Lionel Briand, IEEE Fellow
FNR PEARL chair

May 18th, 2013

*MISE 2013, San Francisco, CA*

# Talk Objectives

- Identify success criteria in research projects with industrial collaborators
- Share concrete and practical guidelines
- Illustrated with recent, personal projects

# Acknowledgements

- Shiva Nejati
- Mehrdad Sabetzadeh
- Nadia Alshahwan
- Domenico Bianculli
- Morayo Adedjouma
- Elizabeta Fourneret
- Reza Matinnejad
- Dan Di Nardo
- Chetan Arora
- Razieh Bejati
- …

# Luxembourg

- Smaller than Rhode Island
- One of the wealthiest countries in the world
- Young research focused university (2003) and Ph.D. programs (2007)
- ICT security and reliability is a national research priority
- Priorities implemented as interdisciplinary centres
- International city and university
- Three official languages: English, French, German


# SnT Centre

- SnT centre, Est. 2009: Interdisciplinary, ICT security, reliability, and trust (SnT)
- 180 scientists and Ph.D. candidates, 20 industry partners
- SVV Lab: Established January 2012, www.svv.lu
- 15 scientists (Research scientists, associates, and PhD candidates)
- Industry-relevant research on system dependability: security, safety, reliability
- Four partners: Cetrel, CTIE, Delphi, SES, …

# Engineering Research

- "Engineering: The application of scientific and mathematical principles to practical ends such as the design, manufacture, and operation of efficient and economical structures, machines, processes, and systems." (American Heritage Dictionary)
- Engineering research: Innovative engineering solutions
  - Problem driven
  - Real world requirements
  - Scalability
  - Human factors, where it matters
  - Economic tradeoffs and cost-benefit analysis
  - Actually doing it on real artifacts, not just talking about it

# Motivation

- Closer industry involvement in MDE research:
  - Research informed by practice
  - Well-defined problems in context
  - Realistic evaluation
  - Long term industrial collaborations => Impact
  - Focus on pain points in industry
- Lessen dichotomy between research and innovation
- Research-driven innovation
- How do we do that?

Editor: **Philippe Kruchten**
University of British Columbia
pbk@ece.ubc.ca

# Embracing the Engineering Side of Software Engineering

Lionel Briand

**I HAVE NOW** been a professional researcher in software engineering for roughly 20 years. Throughout that time, I've worked at universities and in research institutes and collaborated on research projects with 30-odd private companies and public institutions. Over the years, I have increasingly questioned and reflected on the impact and usefulness of my research work and, as a result, made it a priority to combine my research with a genuine involvement in actual engineering problems. This short piece aims to reflect on my experiences in performing industry-relevant software engineering research across several countries and institutions.

## Not So Hot Anymore

I suppose a logical start for this article is to assess, albeit concisely, the current state of software engineering research. As software engineering is widely taught in many universities, due in large part to a strong demand for software engineers in industry, the number of software engineering academics is substantial. The *Journal of Systems and Software* ranks researchers every year, usually accounting for roughly 4,000 individuals actively publishing in major journals.

When I started my career, software engineering was definitely a hot topic in academia: funding was plentiful, and universities and research institutes were hiring in record numbers. This clearly isn't the case anymore. Public funding for software engineering research has at best stagnated, and in many countries, declined significantly. Hiring for research positions is limited and falls far below the number of software engineering graduates seeking research careers. Industry attendance at scientific software engineering conferences is roughly 10 percent, including the scientists from corporate research centers. Adding insult to injury, in many academic and industry circles, software engineering research isn't even considered to be a real scientific discipline. I'll spare you the numerous unpleasant comments about the credibility and scientific underpinning of software engineering research that I've heard over the years.

This situation isn't due to the subject matter's lack of relevance. Software systems are pervasive in all industry sectors and have become increasingly complex and critical. The software engineering profession repeatedly tops job-ranking surveys. In many cases, most of a product's innovation lies in its software components—for an example, think of the automotive industry. In all my recent industry collaborations, I've observed that all the issues and challenges traditionally faced in software development are becoming more acute.

So how can we explain the paradox of being both highly relevant and increasingly underfunded and discredited?

## Looking for Some Answers

Like other disciplines before us, because we're a young and still-maturing engineering field, we lack the credibility of more

*continued on p. 93*



0740-7459/12/$31.00 © 2012 IEEE

# Related work

- **[Mohagheghi & Dehlen 2008], [Hutchinson et al. 2011]:**
  - Investigate success and failure factors for MDE in industry
  - Methods: literature reviews, surveys, and interviews

- **[Selic 2012]:**
  - Reflection on both technical and non-technical considerations to improve MDE penetration in the industry

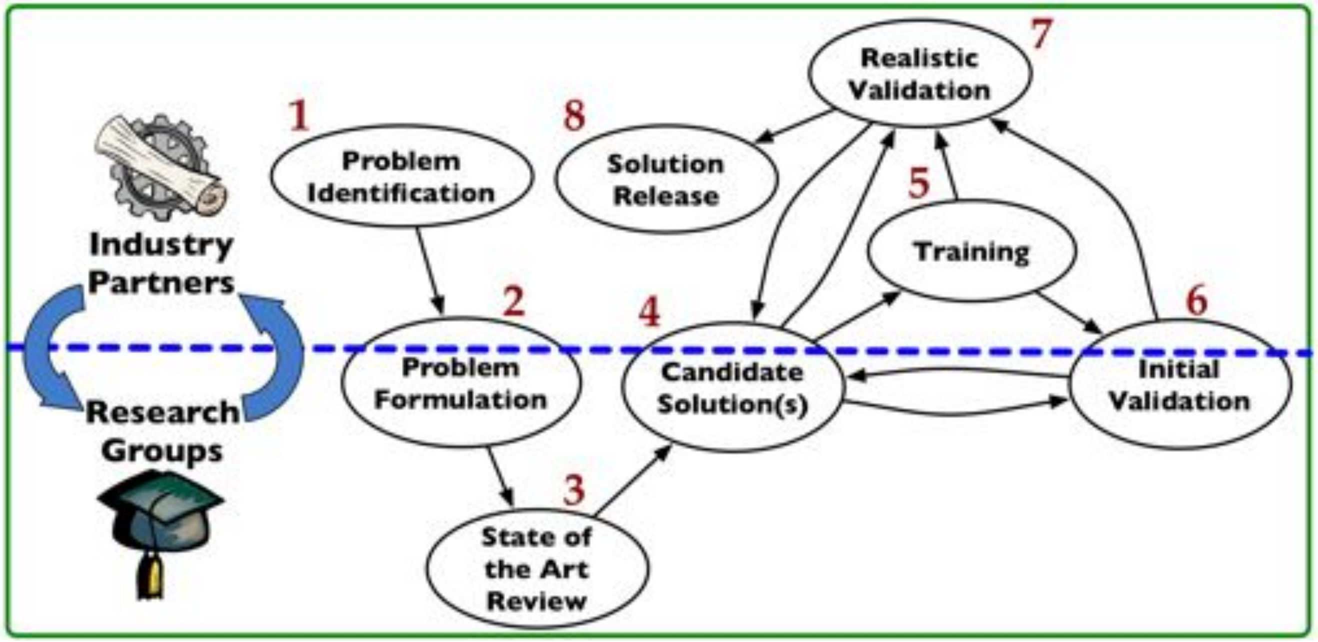
- **Our focus:**
  - MDE research in collaboration with the industry
  - Personal experience across many projects
  - Illustrated with (detailed) examples

# MDE Projects Overview (< 5 years)

| Company | Domain | Objective | Notation | Automation |
|---|---|---|---|---|
| ABB | Robot controller | Testing | UML | Model analysis for coverage criteria |
| Cisco | Video conference | Testing (robustness) | UML profile | Metaheuristic search |
| Kongsberg Maritime | Fire and gas safety control system | Certification | SysML + traceability | Model slicing algorithm |
| Kongsberg Maritime | Oil&gas, safety critical drivers | CPU usage analysis | UML+MARTE | Constraint Solver |
| FMC | Subsea system | Automated configuration | UML profile | Constraint solver |
| WesternGeco | Marine seismic acquisition | Testing | UML profile + MARTE | Metaheuristic search |
| DNV | Marine and Energy, certification body | Compliance with safety standards | UML profile | Constraint verification |
| SES | Satellite operator | Testing | UML profile | Metaheuristic search |
| Delphi | Automotive systems | Testing (safety +performance) | Matlab/Simulink | Metaheuristic search |
| Lux. Tax department | Legal & financial | Legal Req. QA & testing | Under investigation | Under investigation |

# Industrial collaboration model

- Adapted from [Gorschek et al 2006]




- Similar to action research
  - Solving a real-world problem while studying the experience of solving it [Davison et al 2004]
- Difference with action research:
  - More conservative in terms of intervention
  - Researchers are not the agents of change

# Defining the research problems

***Lesson: The stated problem is often a manifestation of one or more fundamental problems***

- An early observational study can identify and decompose these fundamental problems

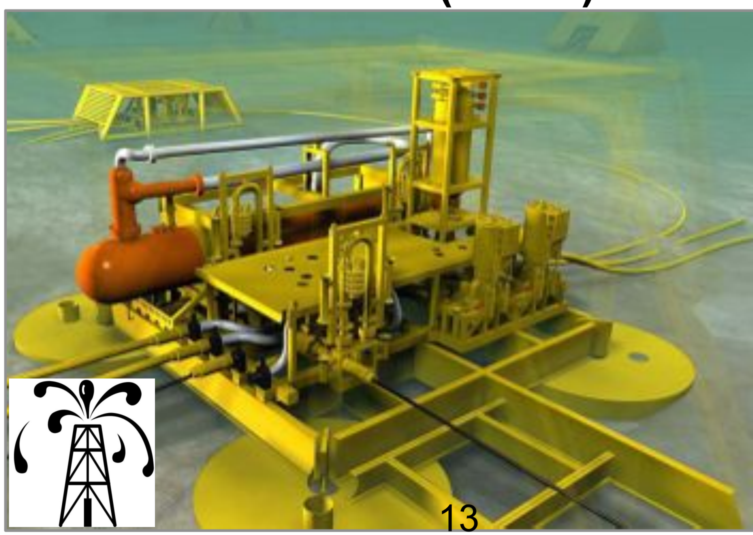
**Example:** Integration problem in subsea/automotive systems

- Investigation pointed to root causes of "integration problems"
- Subsea: Configuration
- Automotive: Balancing CPU usage

- Observational study also essential for mapping the terms used by an industry partner to the terms used in the research literature

# Subsea Integrated Control Systems

**Subsea Christmas (Xmas) tree**

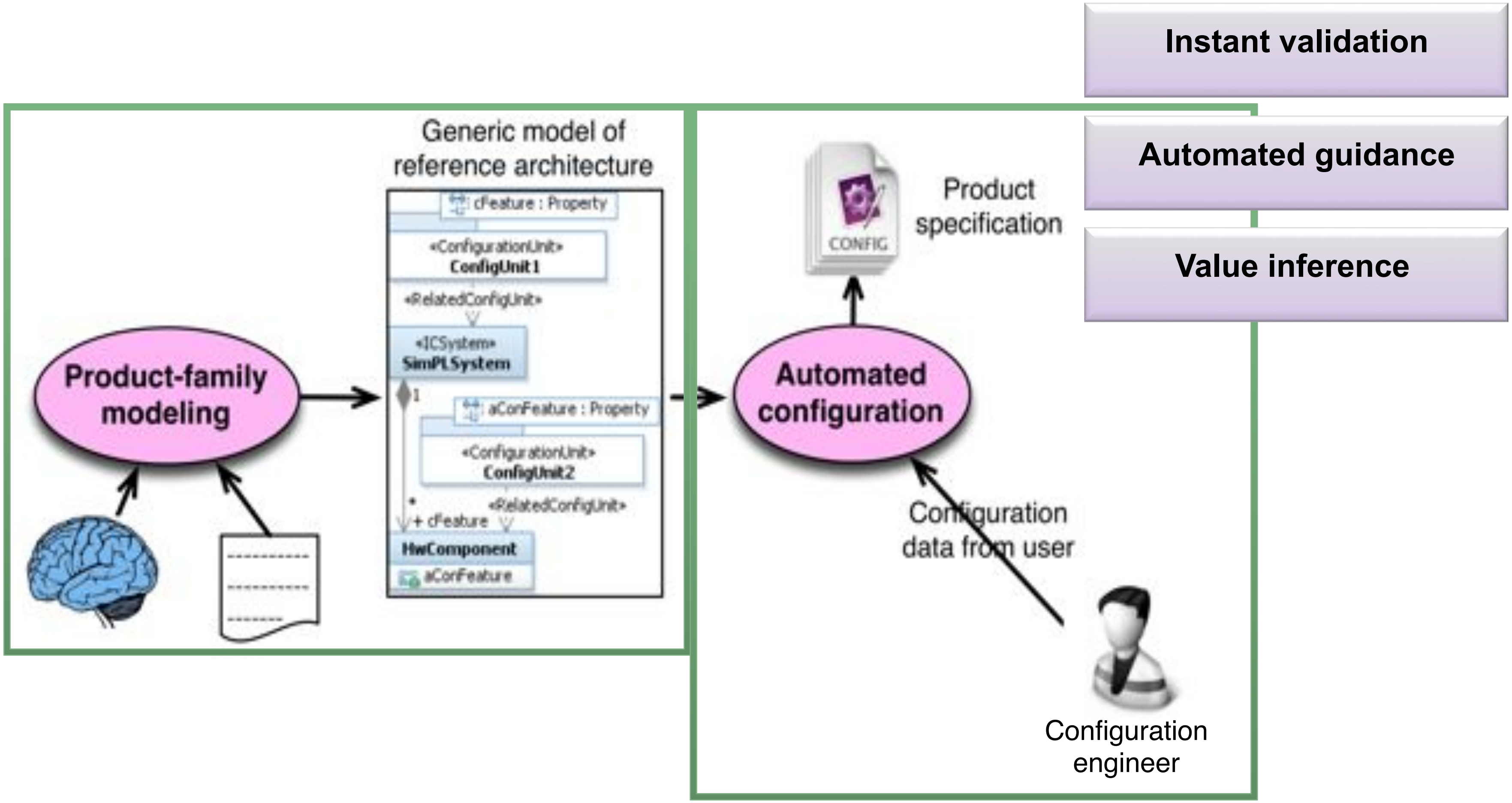




*http://subseaworldnews.com/*

Integration problems!

# Model-Based Configuration



# Integration in Power Train Systems

SnT


OEM

Original Equipment Manufacturer

AUTOSAR Models

+

AUTOSAR Models

sw runnables

+

Glue

+

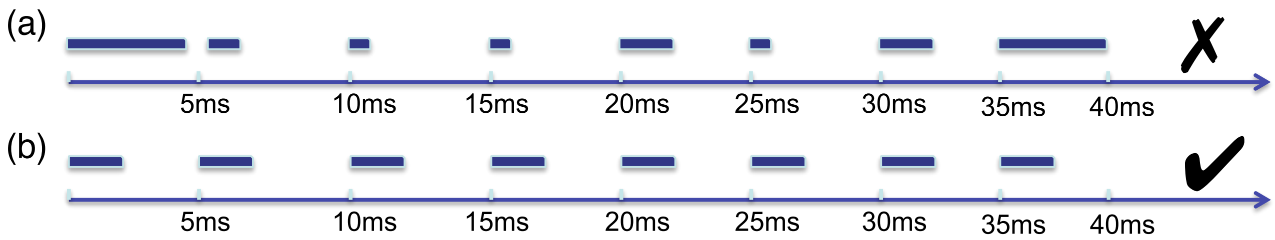
sw runnables

DELPHI

Automotive Systems

# Balancing CPU Usage Across OS Cycles

- Challenge
  - Integration problems!
  - Many OS tasks and their many runnables run within a limited available CPU time
    - The execution time of the runnables may exceed the OS cycles
- Our goal
  - Reducing the maximum CPU time used per time slot to be able to
    - Reduce the possibility of overloading the CPU in practice
    - Minimize the hardware cost
    - Enable addition of new functions incrementally



# Contextual factors

**_Lesson: Context matters!_**

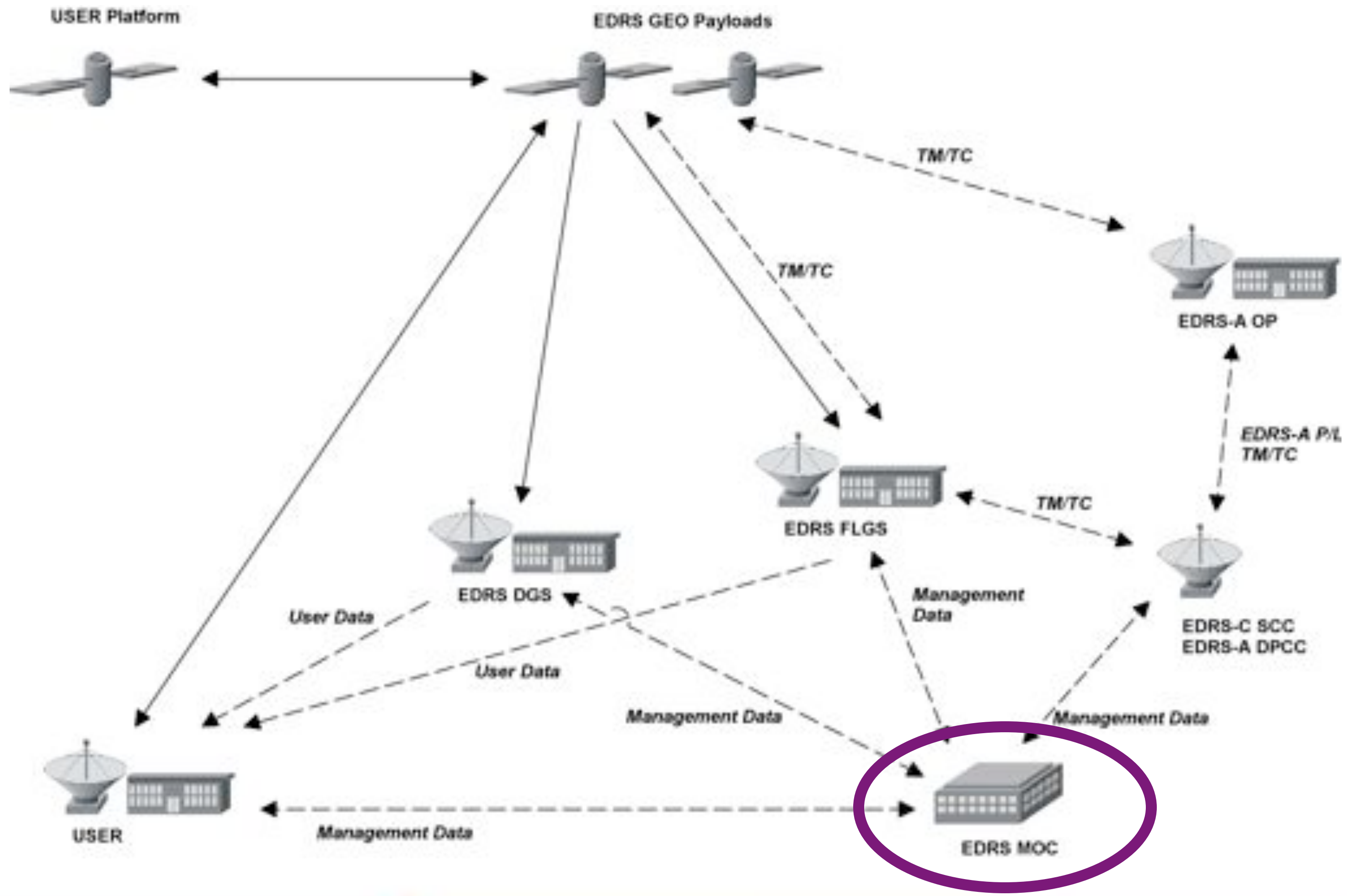
- Contextual factors (incl. assumptions) determine
  - what is feasible and what is not
  - what is cost-effective and what is not
  - what can be reused from the existing literature and what needs a novel solution

**Examples:**
- Automotive: Use of Matlab/Simulink, test phases
- Model-driven Testing based on Matlab/Simulink

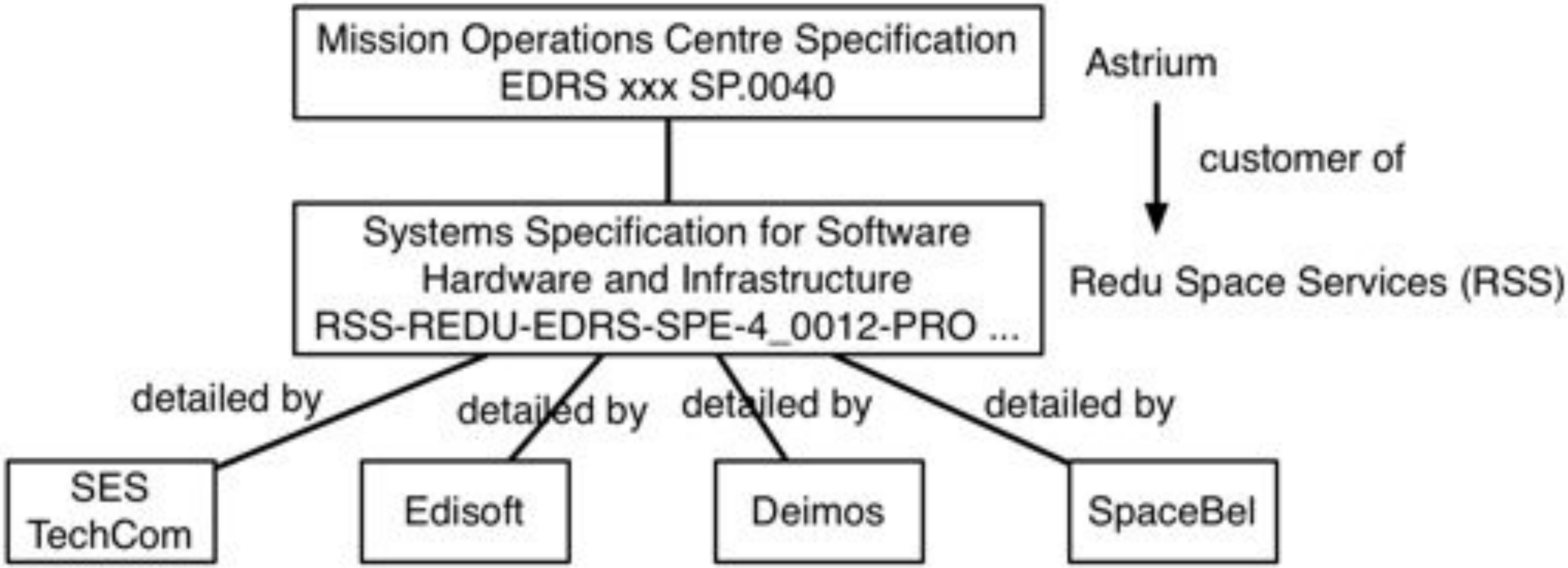
- Satellite: Many stakeholders, requirements in natural language
- Quality assurance for natural language requirements?

# Satellite Ground Control Systems

- MOC (Mission Operations Centre) system in EDRS (European Data Relay Satellite)



# Many stakeholders, Three Tier Requirements

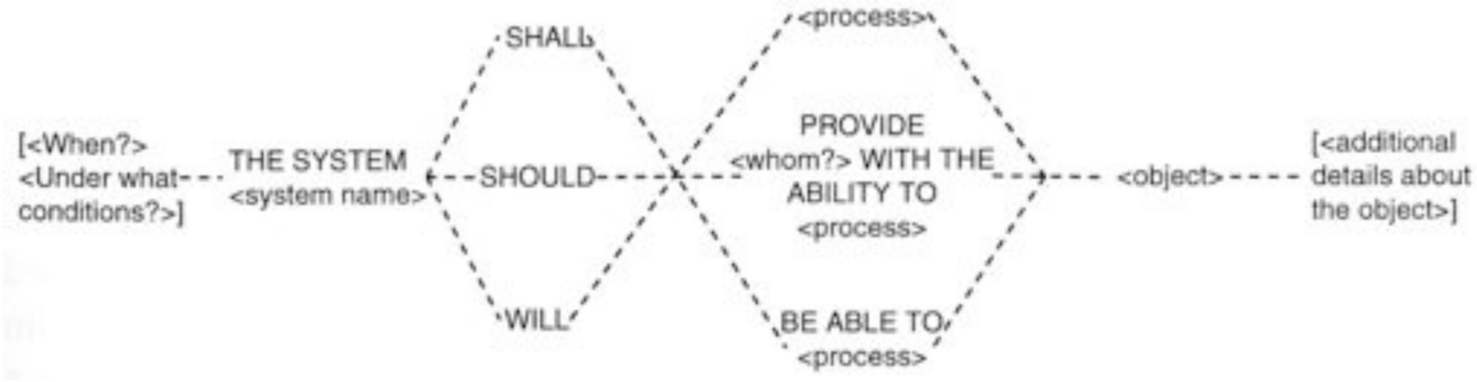



- Many opportunities for misunderstanding and changes
- Focus on automated requirements quality assurance
- Natural language requirements

# Applying Templates

• Motivation: Requirements statements are often expected to follow a sentence template to maximize comprehension and minimize ambiguity

• Example [Rupp 2009]




• Manually checking of conformance to templates is very tedious
• Especially in the context of many stakeholders and changes
• Questions: Can conformance to templates be checked automatically? Do we need a glossary of key phrases (domain concepts)?

# Syntactic checking (continued)

**Example:**



*The Monitoring and Control component shall provide the system operator with the ability to configure the database polling interval.*

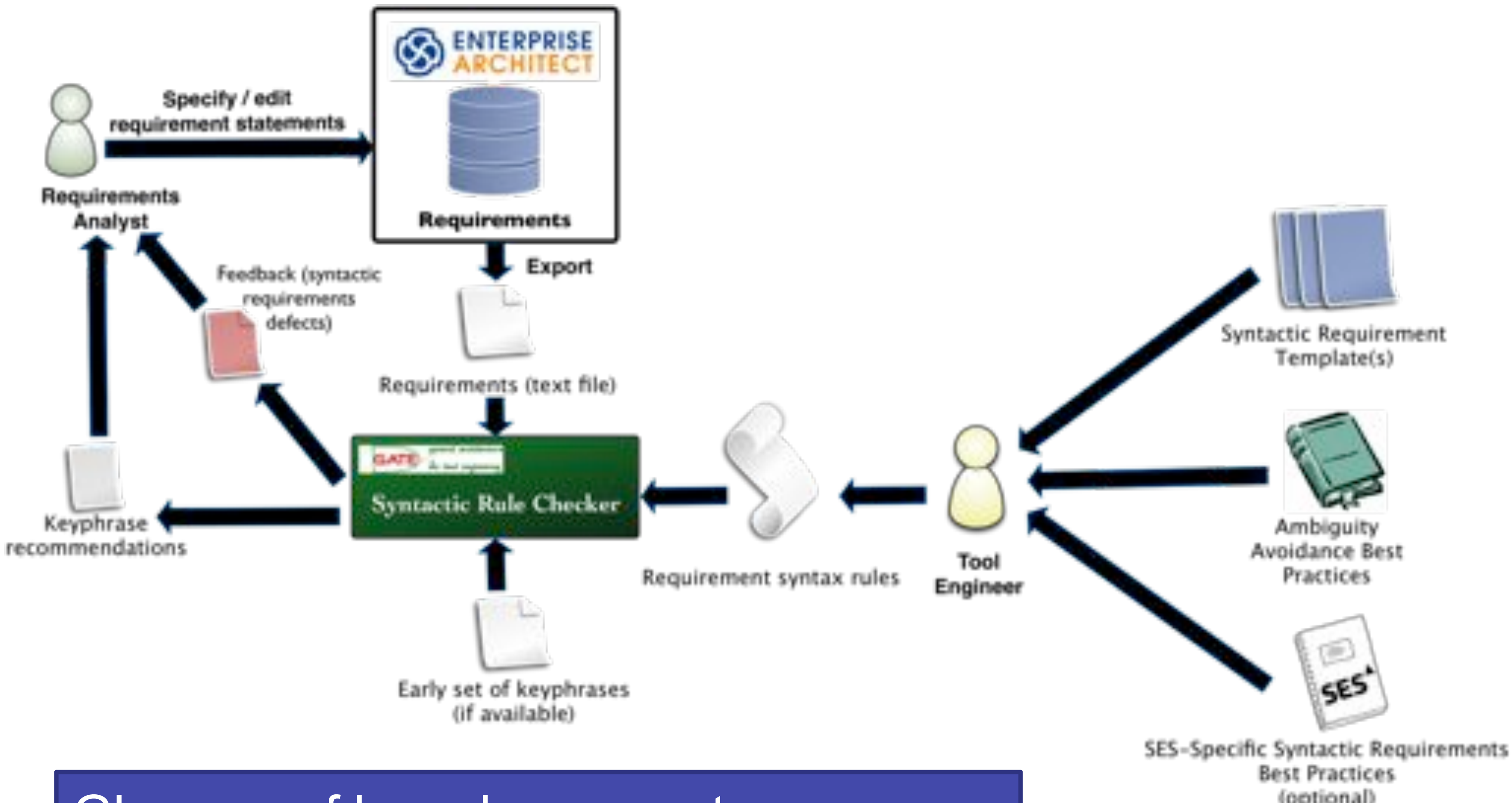


Conforms to template? **Yes**

- Commercial tools already exist for syntax checking **BUT …**
- Results are poor when the glossary terms have not been specified
- … research suggests that most projects have substantial omissions in glossaries, particularly in early stages

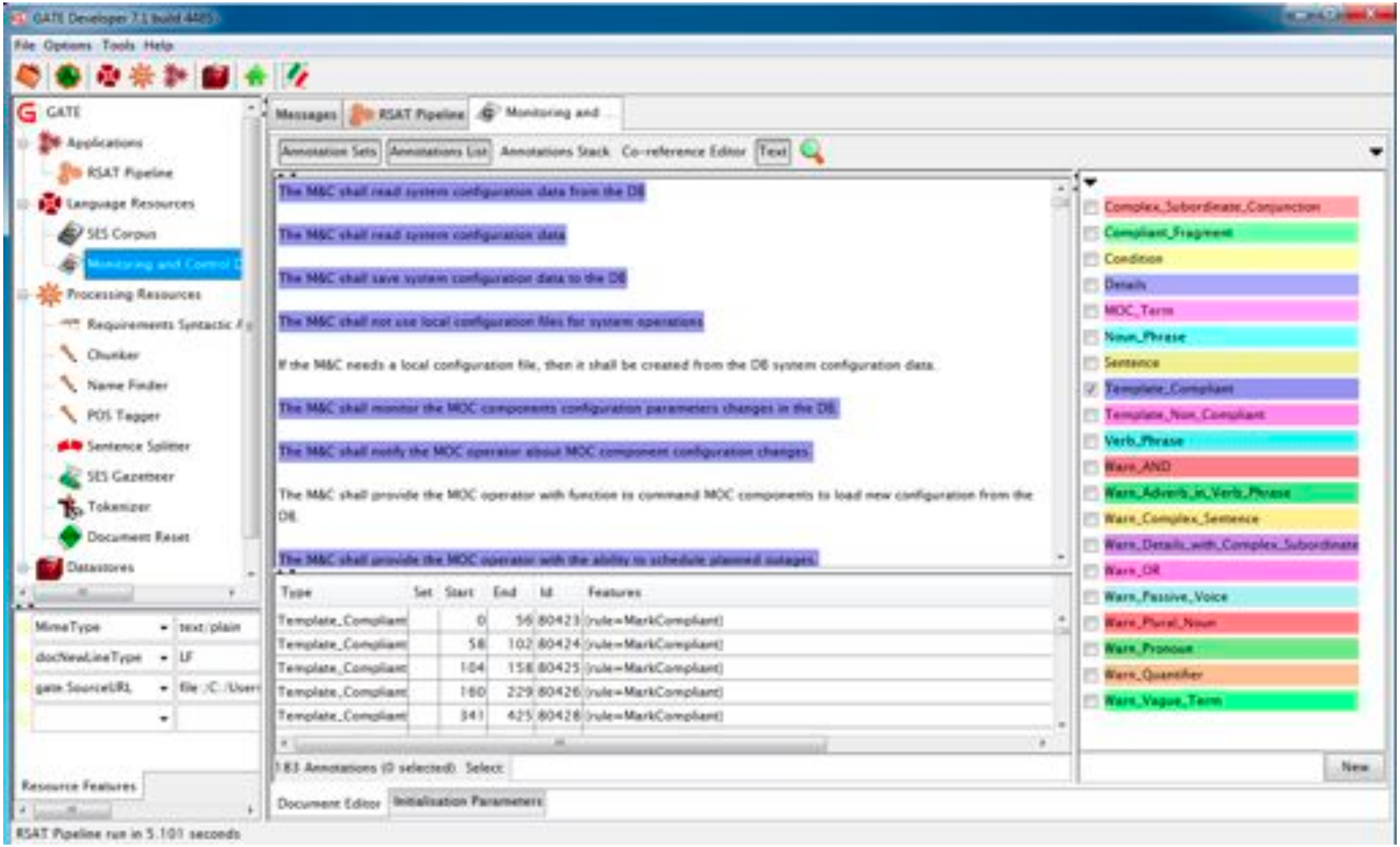
- **Our goal:** syntax checking with **minimal** reliance on glossaries
- … and provide recommendations for glossary terms

# Solution Overview




Glossary of key phrases not necessary

# Tool snapshot



# Context Factors?

- Natural languages requirements
- Many stakeholders
- Frequent changes
- Requirements documents approved as contractually binding
- …

# Complexity and amount of software used on vehicles' Electronic Control Units (ECUs) grow rapidly

*Comfort and variety*

*More functions*

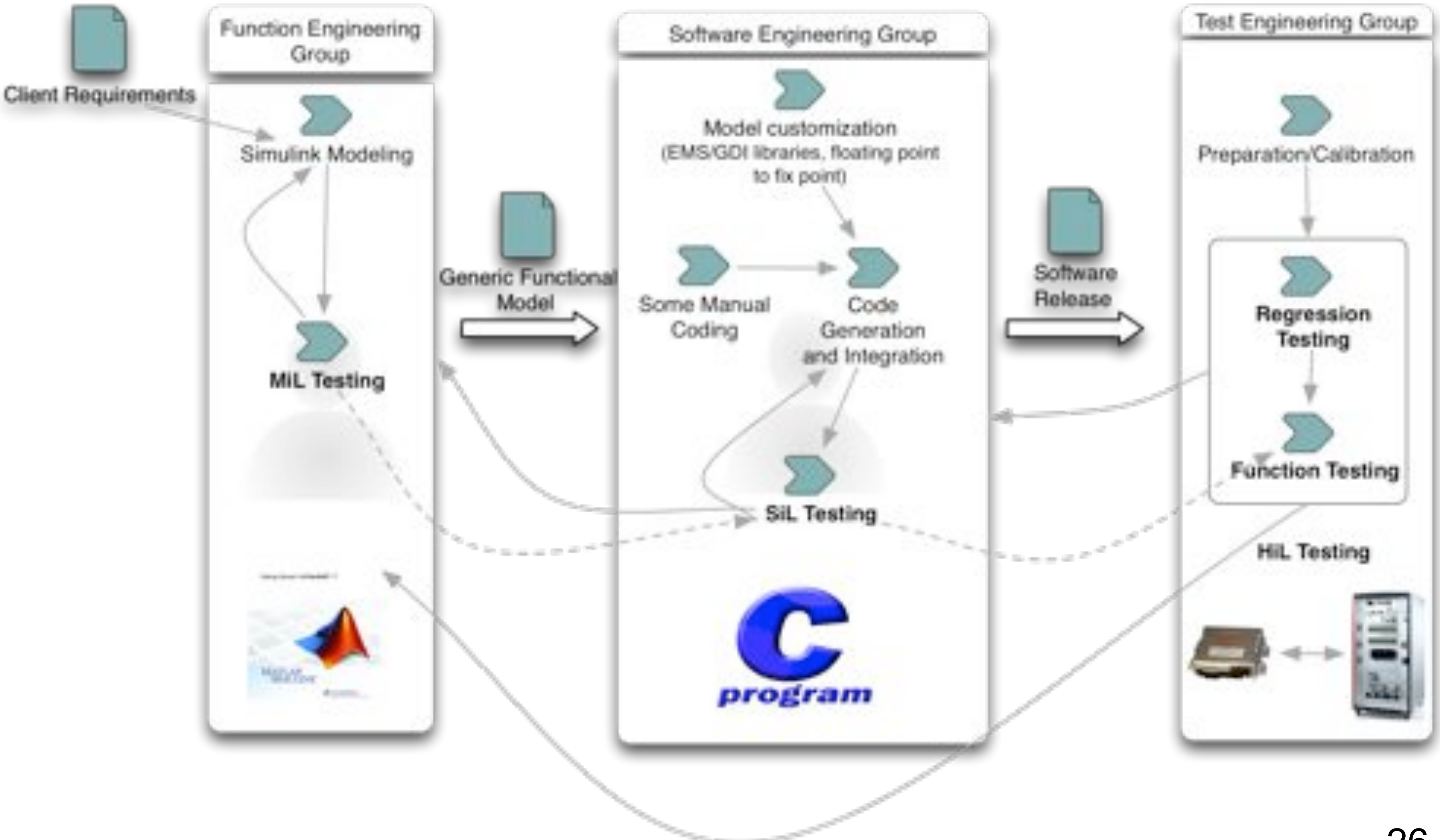
*Safety and reliability*

*Faster time-to-market*

*Greenhouse gas emission laws*

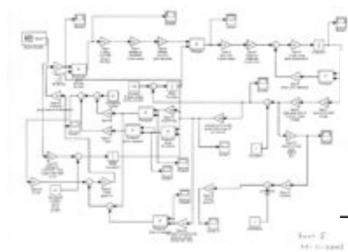
*Less fuel consumption*

# Three major software development stages in the automotive domain

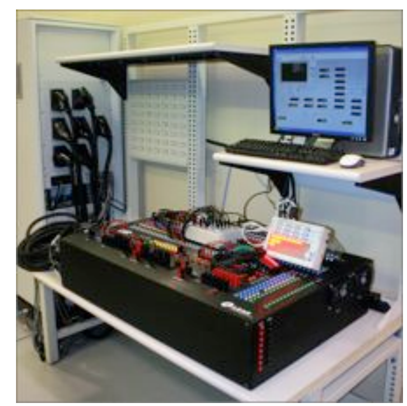


# MiL testing

***Requirements***



***Individual Functions***



- The ultimate goal of MiL testing is to ensure that individual functions behave correctly and timely on any hardware configuration
- But is also a mechanism to select HiL test cases

# Context Factors?

- Stages of testing
- Final testing stage extremely expensive and time consuming
- Systematic use of Matlab/Simulink at early stages

# Improving domain understanding and communication

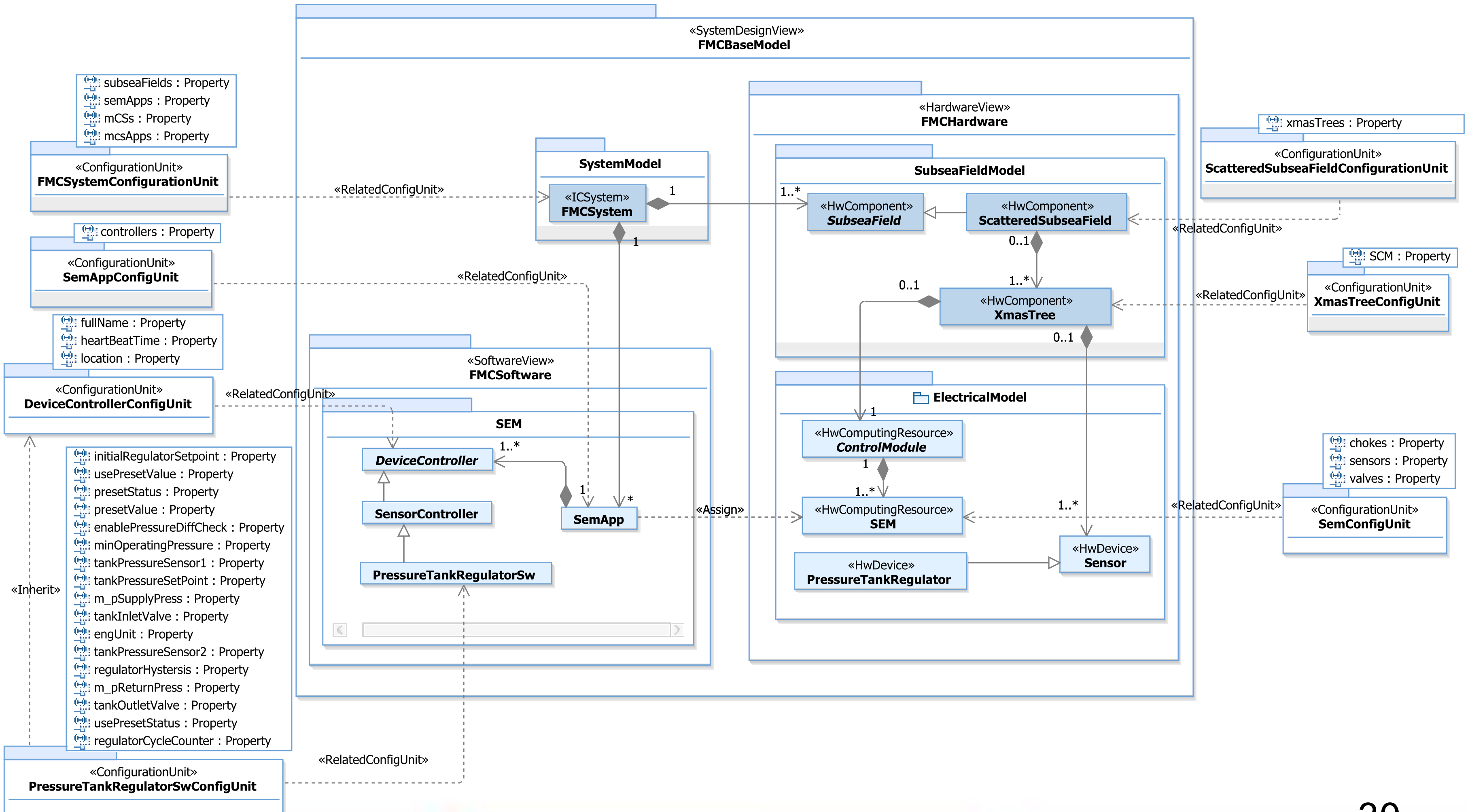
***Lesson: Build a domain model as early as possible.***

- Helps researchers better understand the domain
- An essential communication tool between partners and practitioners
- By product: Helps practitioners better organize, refine and share their knowledge

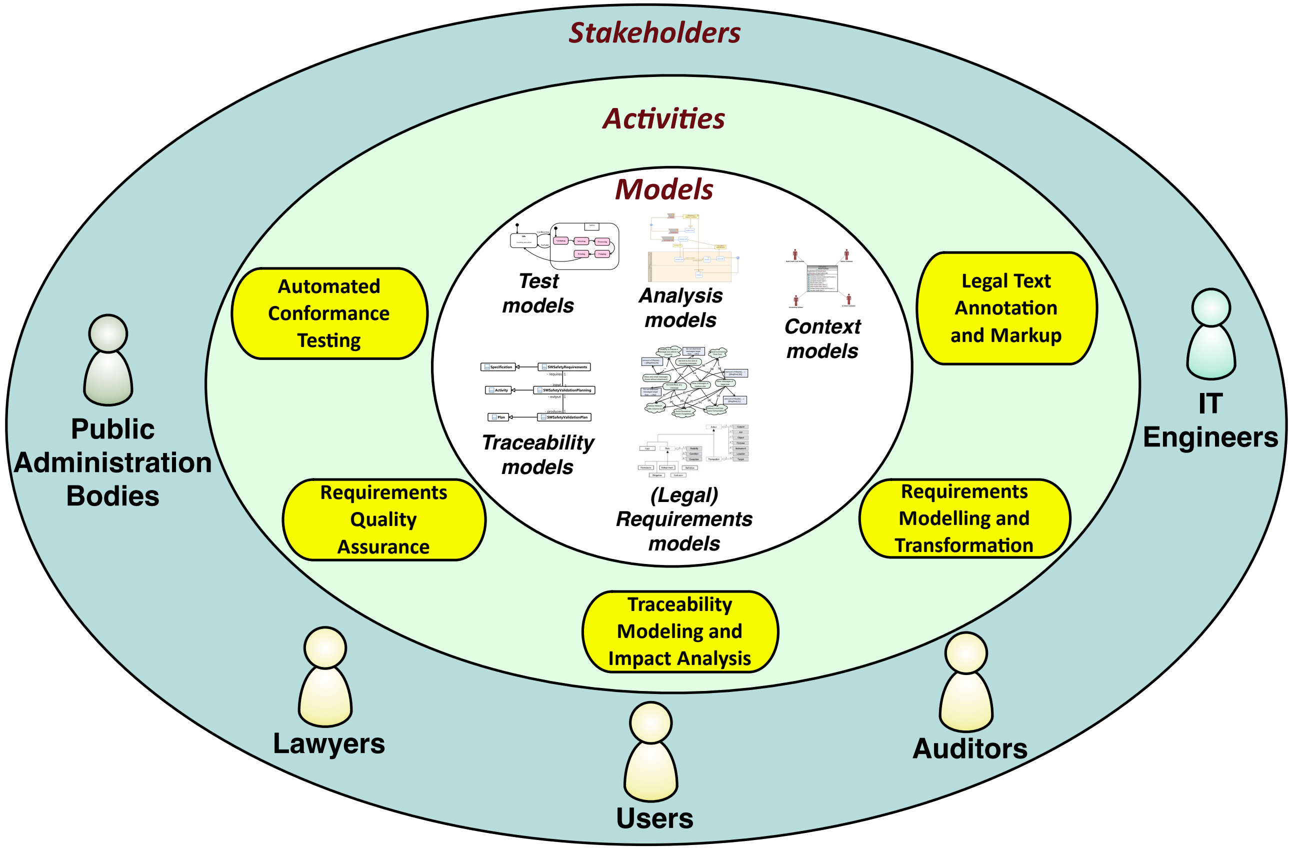
**Examples:**

- **Subsea:** 71 classes: 46 for software and 24 for hardware
- Captures SW-HW relationships, configurable parameters, variability points (60 – 80 person hours to build)
- **Tax law: next**

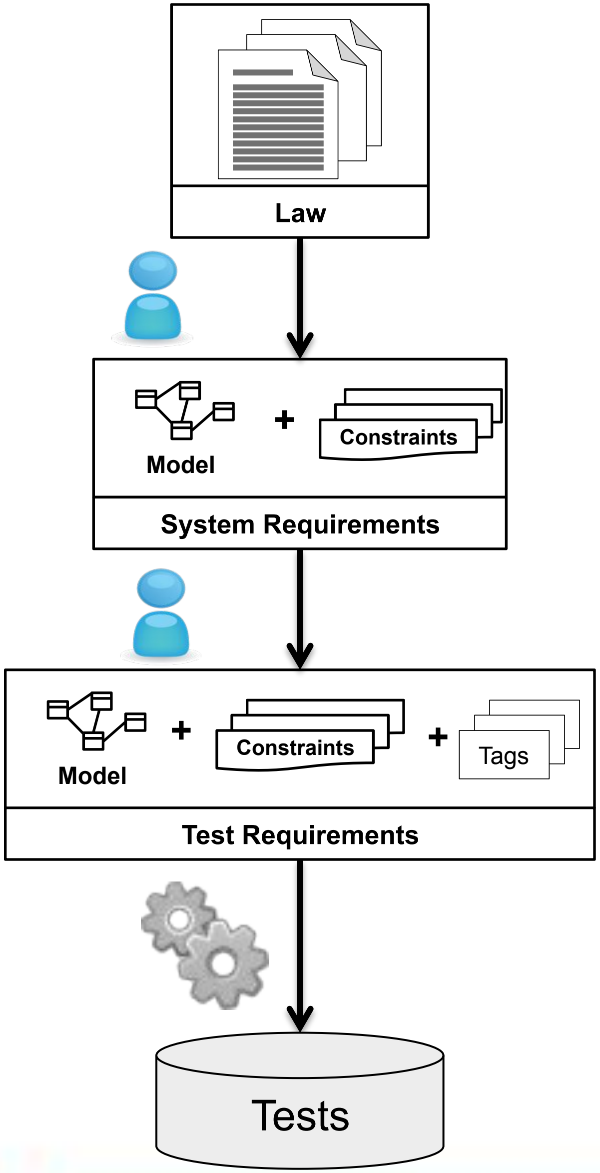
# Subsea Domain Model (Using SimPL)



# Verification of Testing and Legal Requirements



# Legal Requirements: Traceability



# Analysis and Verification of Legal Requirements

- ***Consistency checking of legal requirements***
  - The requirements are "interpretations" of the law and could therefore be inconsistent
- ***Handling the constant evolution of the law***
  - How does the evolution of the law impact legal requirements and the developed system
- ***Automated testing and run-time verification***
  - Identification of defects in the developed software both before and after deployment


LE GOUVERNEMENT
DU GRAND-DUCHÉ DE LUXEMBOURG
Administration des contributions directes

**Loi modifiée du 4 décembre 1967**
**concernant l´impôt sur le revenu**

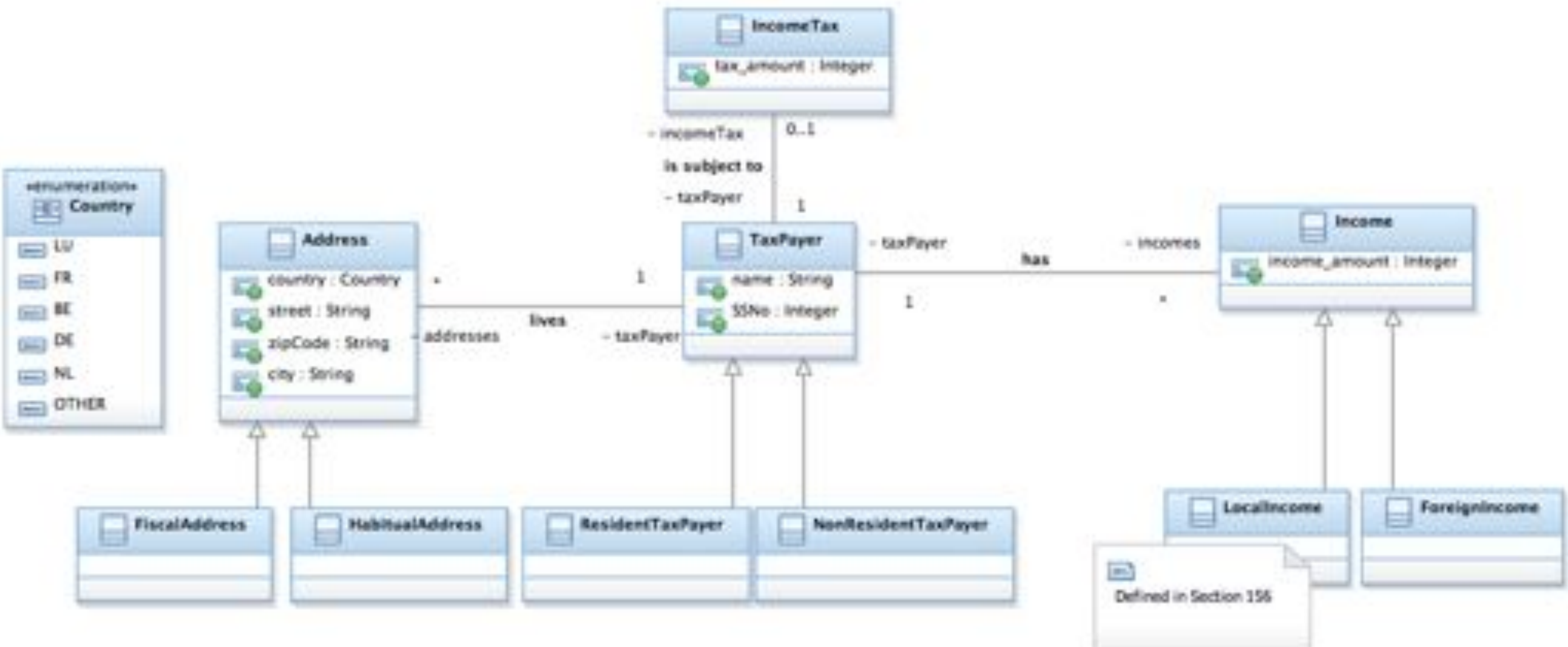
**Texte coordonné au 1er janvier 2013**

# (Very) Partial Domain model for one Article

**Art.2** : Individuals are considered resident taxpayers if they have either their fiscal or habitual residence in the Grand Duchy. Individuals are considered non-resident taxpayers if they neither have their fiscal nor their habitual residence in the Grand Duchy and if they have local income within the meaning of section 156.

Resident taxpayers are subject to income tax because of their income, both local and foreign.

Non-resident taxpayers are subject to income tax only because of their local income within the meaning of section156 below.



# Questions for Legal Experts

- Many interesting questions arise during modeling as the result of following a systematic process:

- Examples:

  - How many fiscal and habitual residences can a tax payer have?
    - Multiplicities over associations between taxpayer and the relevant concepts
  - What is local income:
    - Do we refer to income paid by a Luxembourgish company ? or
    - to income earned over work rendered in Luxembourg?

Important as in early stages, researchers need to mentor by doing what they preach!

# Considerations about input

***Lesson: Input models should be feasible to build!***

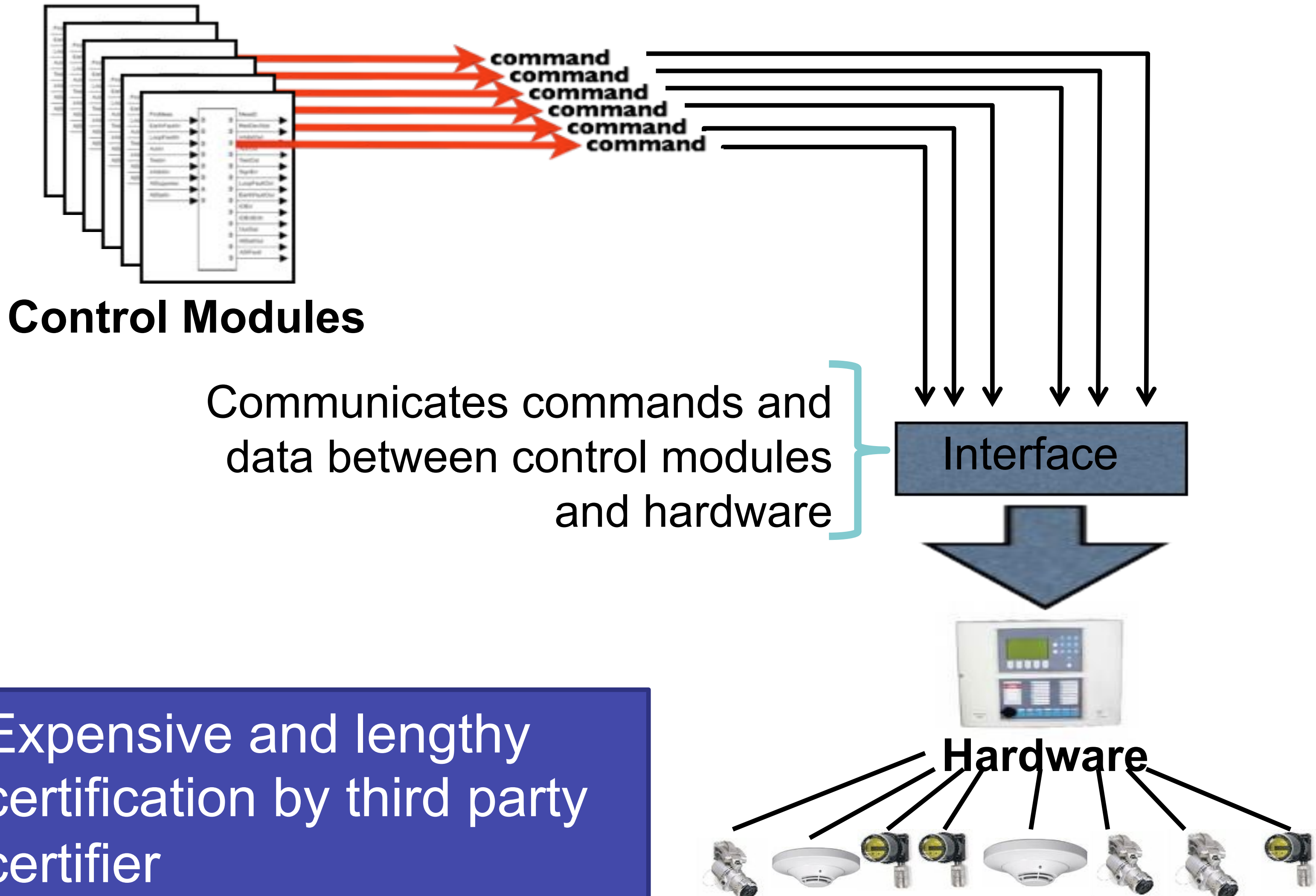
• The input has to match the expertise, culture, and processes at the industry partner

***Lesson: Use standardized notations when possible***

• Allows building on existing tools
• Avoids technology lock-in
• Reduces communication issues in a multi-organization setting

**Example: safety-critical drivers in fire&gas monitoring**
• SysML was used as the basis for modeling
• A general methodology was developed and later simplified according to project needs and expertise of the partner
• Tool support was developed over a short time building on Enterprise Architect and its SysML plugin

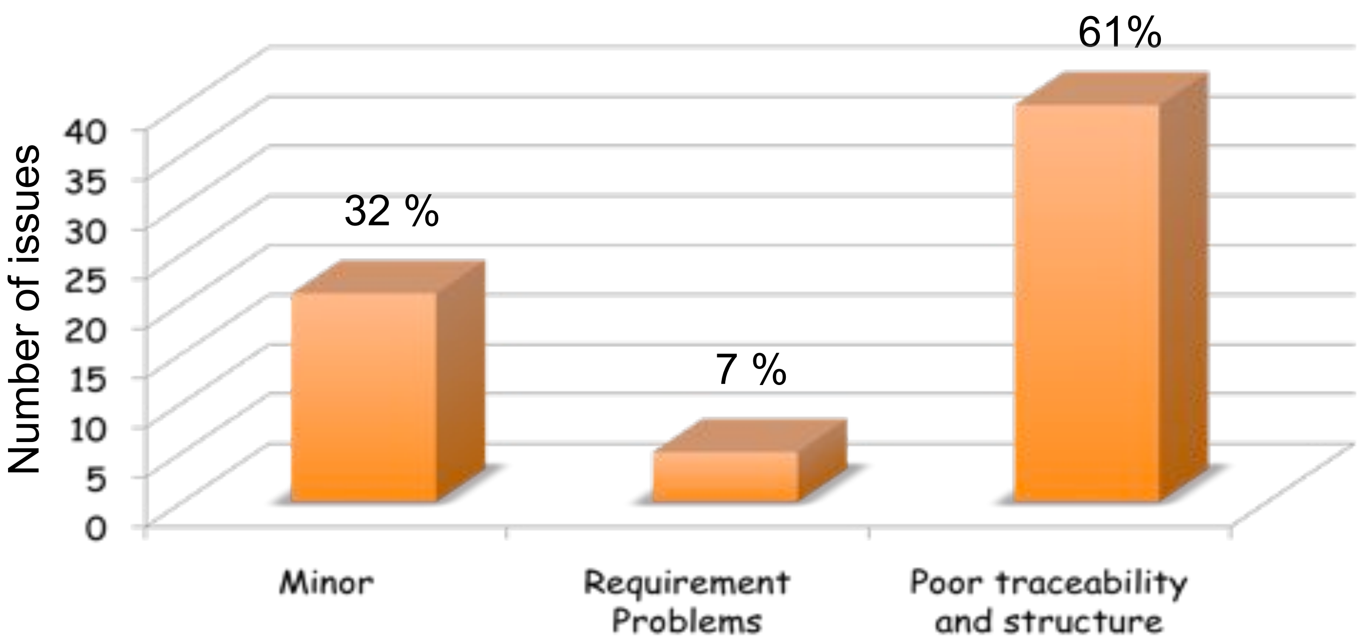
# Gas&Fire Monitoring and Emergency Shutdown




Expensive and lengthy certification by third party certifier

# Observation Study of Certification Meetings

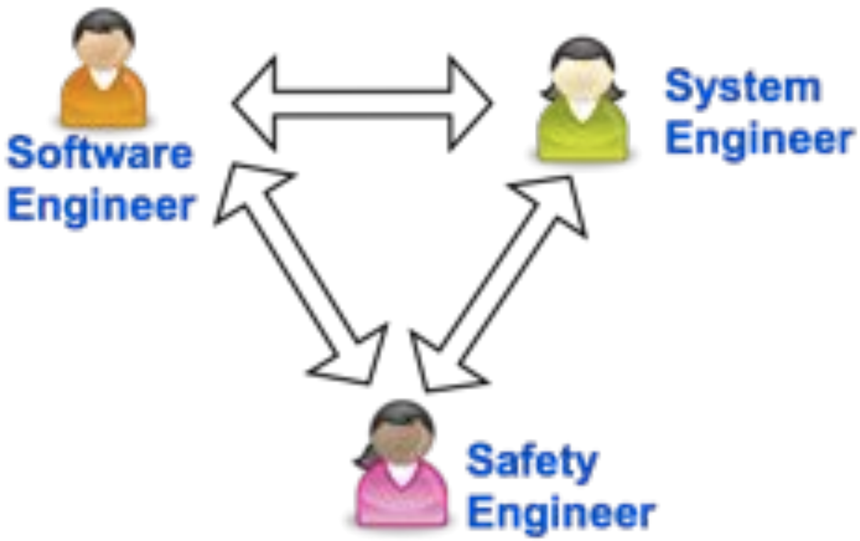
- System certified by third party
- Drivers are SIL 3
- Attended certification meetings (observational study)
- Meetings focused on requirements, architecture, and design documents
- Analyzed 66 distinct certification issues




Various forms of traceability mandated by standards

# Objectives, Language, Tool

Traceability
Methodology
**to relate safety**
**Requirements to design**

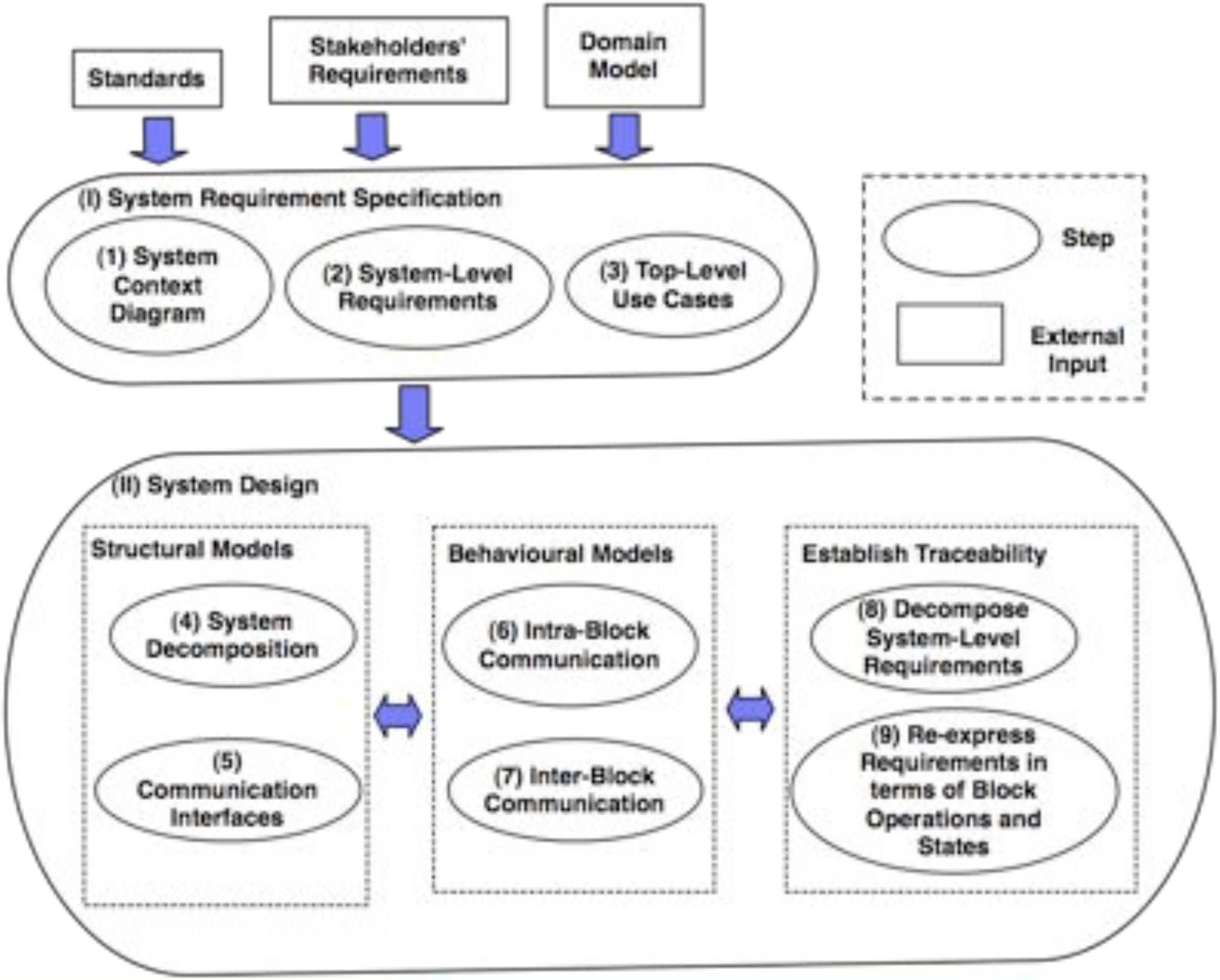



Slicing
Algorithm
**to extract a design slice**
**relevant to a given**
**safety requirement**

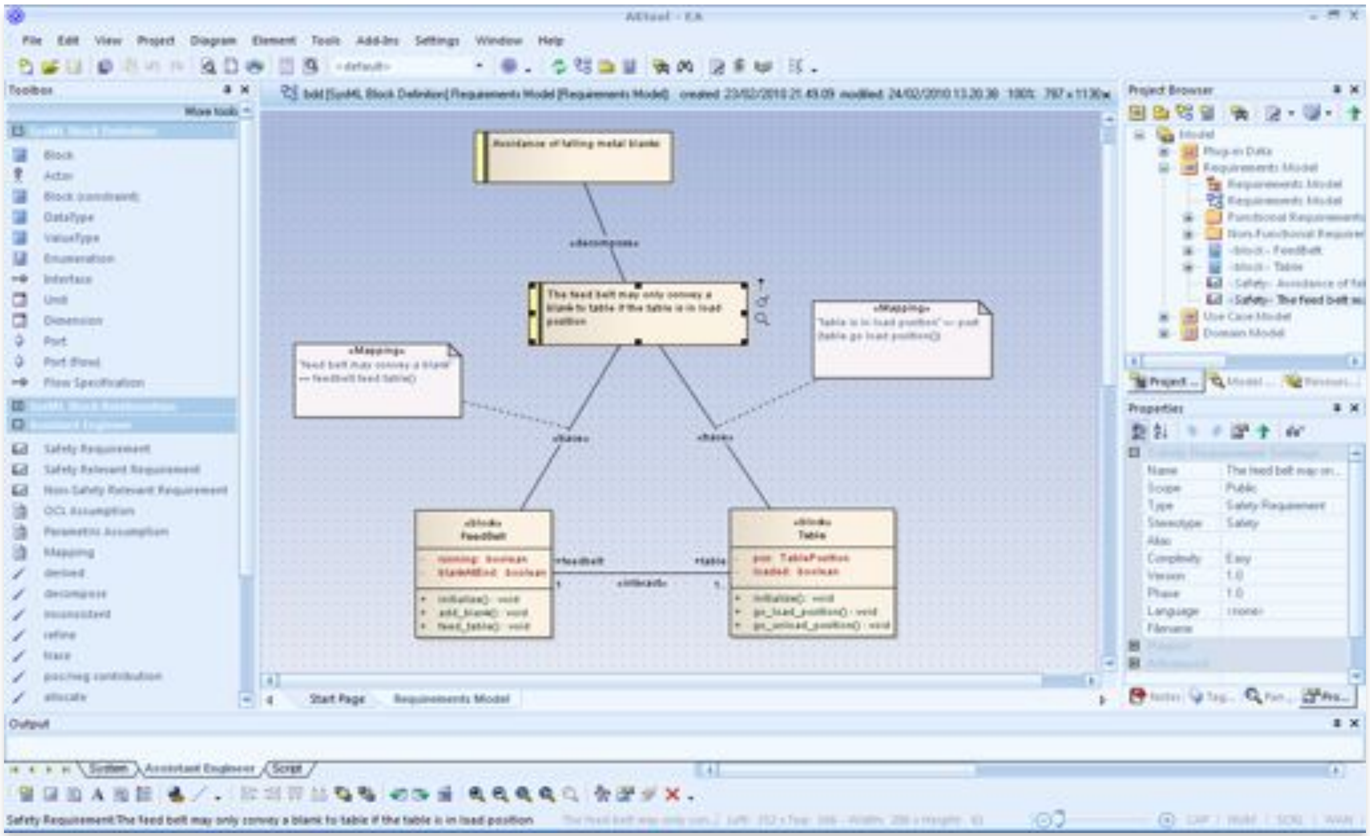
OMG SYSTEMS MODELING LANGUAGE SysML™

ENTERPRISE ARCHITECT

# Modeling Methodology



# Automation: SafeSlice




- Support Establishing Traceability
- Generate Slices
- Inspection Assistant

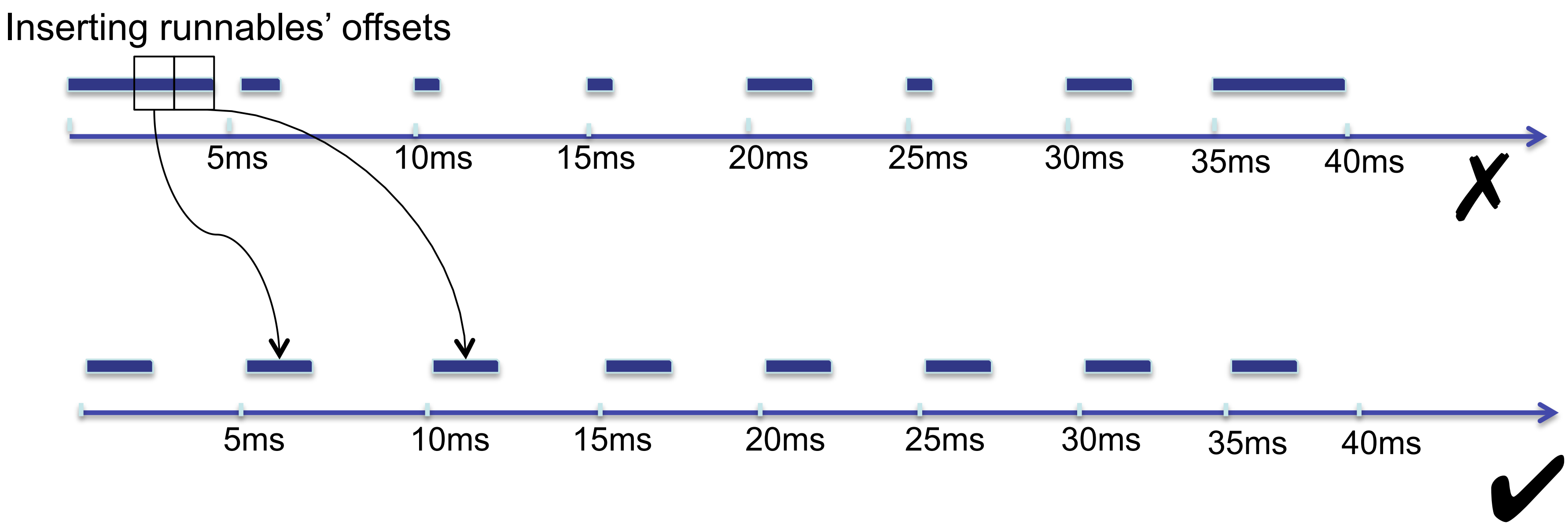
# Less than perfect input

***Be prepared for less than perfect models***

- Better models lead to better analysis results BUT …
- When scales are large, compromises often have to be made
- **Question:** Can the proposed approach still be useful in the presence of incompleteness and imprecision?

**Example:**
- CPU usage analysis for runnables in power train systems
- Can runnables run out of CPU?
- Can we minimize the risk?
- Imprecise WCET estimates
- Approximate knowledge of dependencies between runnables
- No possibility of using model checking (also related to previous points about feasibility of inputs)

# We minimize the maximum CPU usage using runnables offsets (delay times)

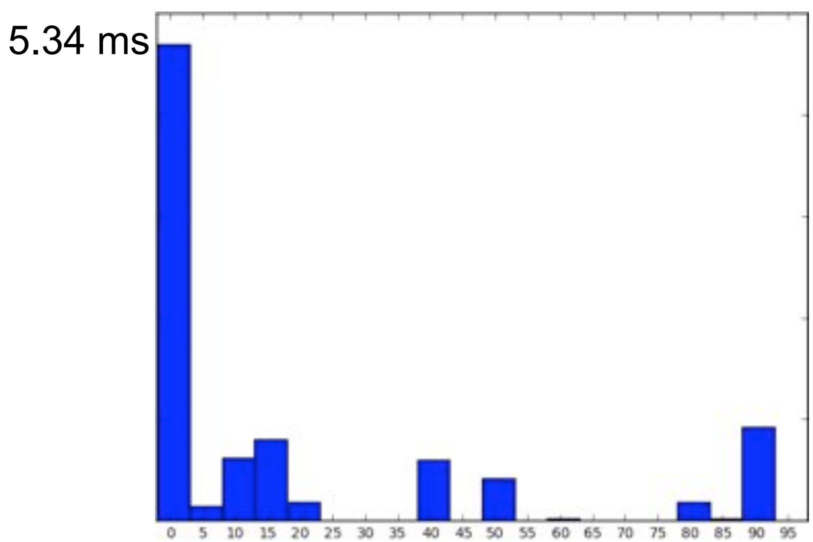



Offsets have to be chosen such that
- the maximum CPU usage per time slot is minimized, and further,
  - the runnables respect their period
  - the runnables respect the OS cycles
  - the runnnables satisfy their synchronization constraints

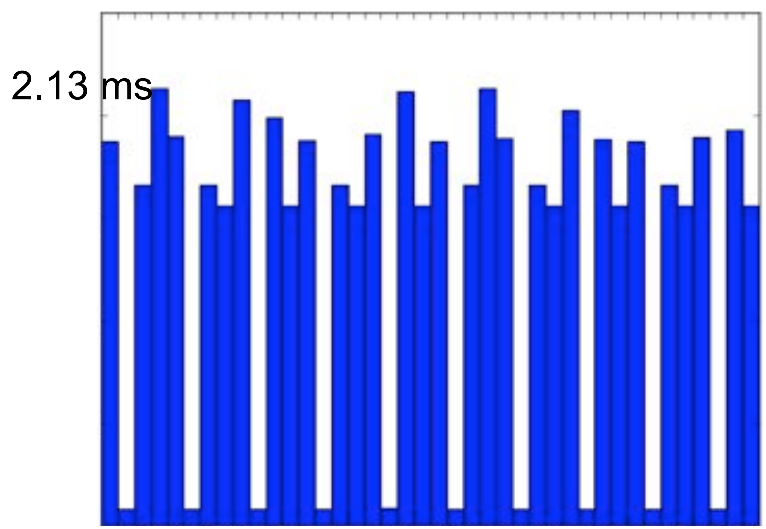
# Meta heuristic search algorithms

- The objective function is the max CPU usage of a 2s-simulation of runnables
- Single-state search algorithms for discrete spaces (e.g., Tabu)
- The search modifies one offset at a time, and updates other offsets only if timing constraints are violated
- Used restart option to make them more explorative

Case Study: an automotive software system with 430 runnables




Running the system without offsets




Our optimized offset assignment

# Training

**_Lesson: Train incrementally and based on needs_**

- Long course on the whole (UML) is not a good idea!
- Training must match what the proposed solutions need

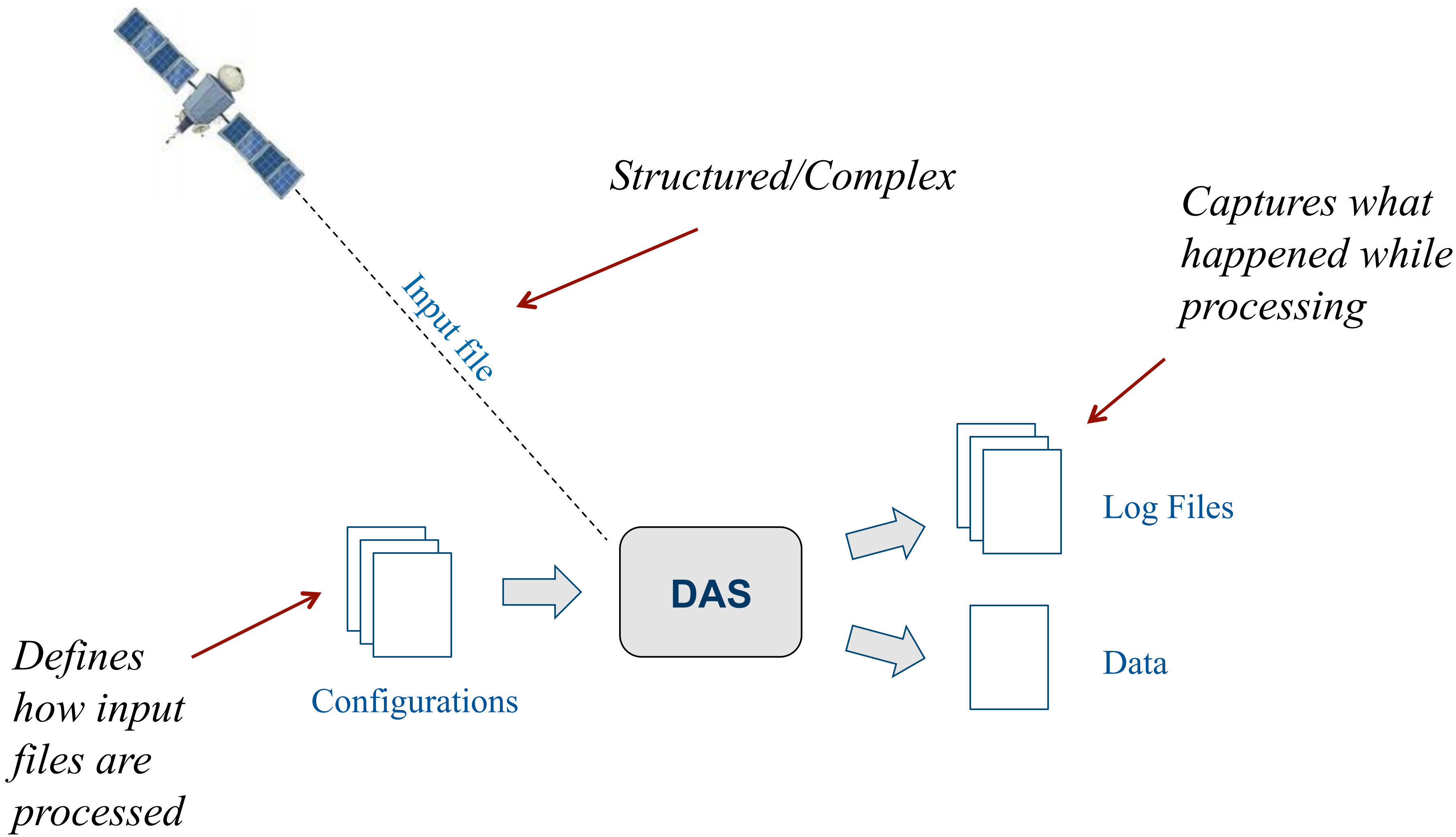
**_Lesson: Use examples and illustrations from the industry partner's application domain_**

- Textbook examples are often met with yawns!
  - They have seen them all, several times!
- Quote from practitioner: *"All these courses I attend use the ATM example. I want to see how UML is applied to **our** system".*

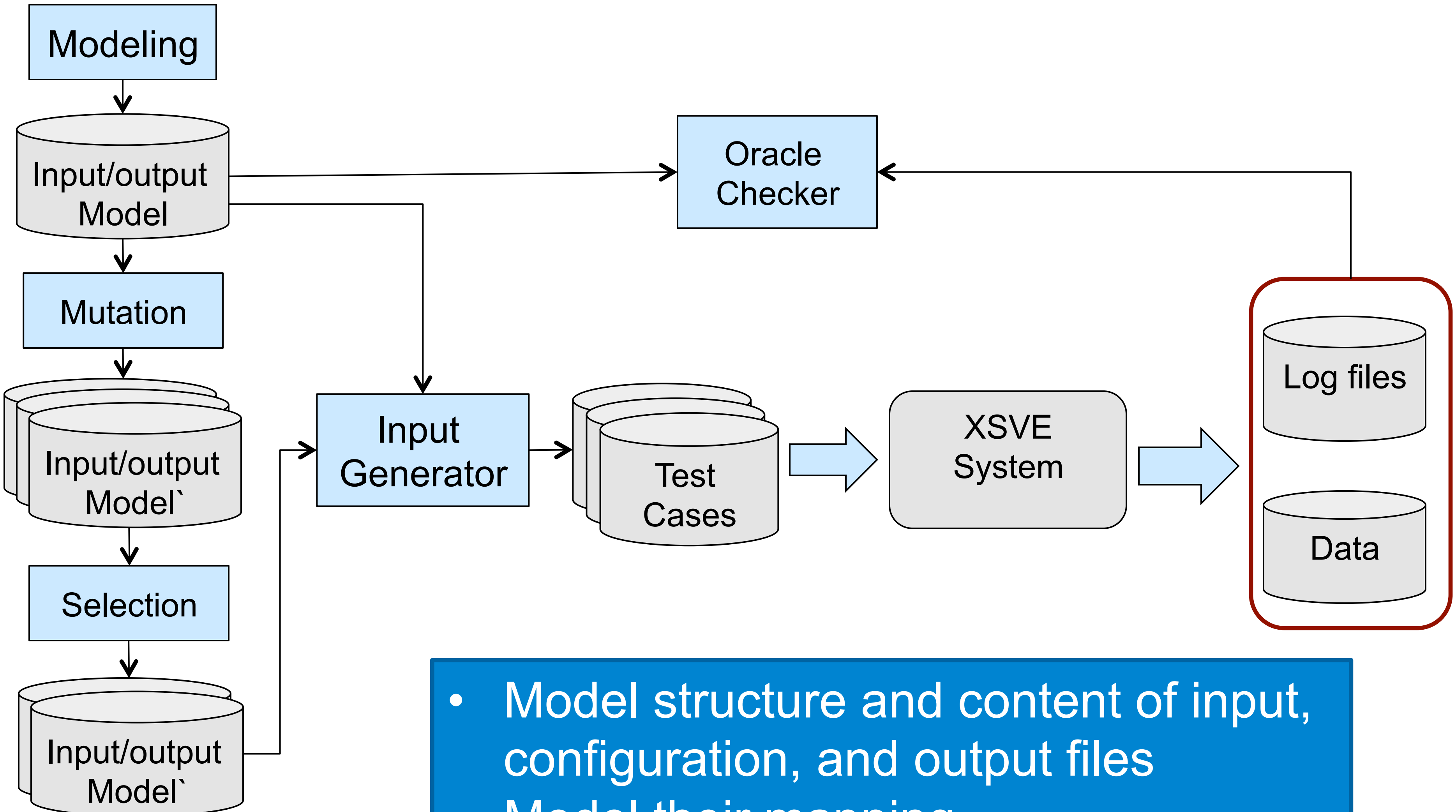
**Example: Data Acquisition Systems (Satellites)**

- Specific modeling methodology to support model-based testing in this domain
- Course on how to use UML and OCL with that methodology
- Used an actual DAS for training

# Context: Data Acquisition Systems



# Modeling and Test Automation Approach




- Model structure and content of input, configuration, and output files
- Model their mapping

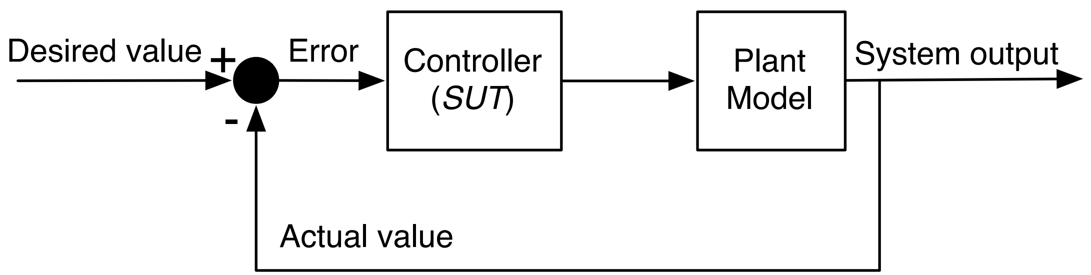
# Early validation not synonymous with artificial validation

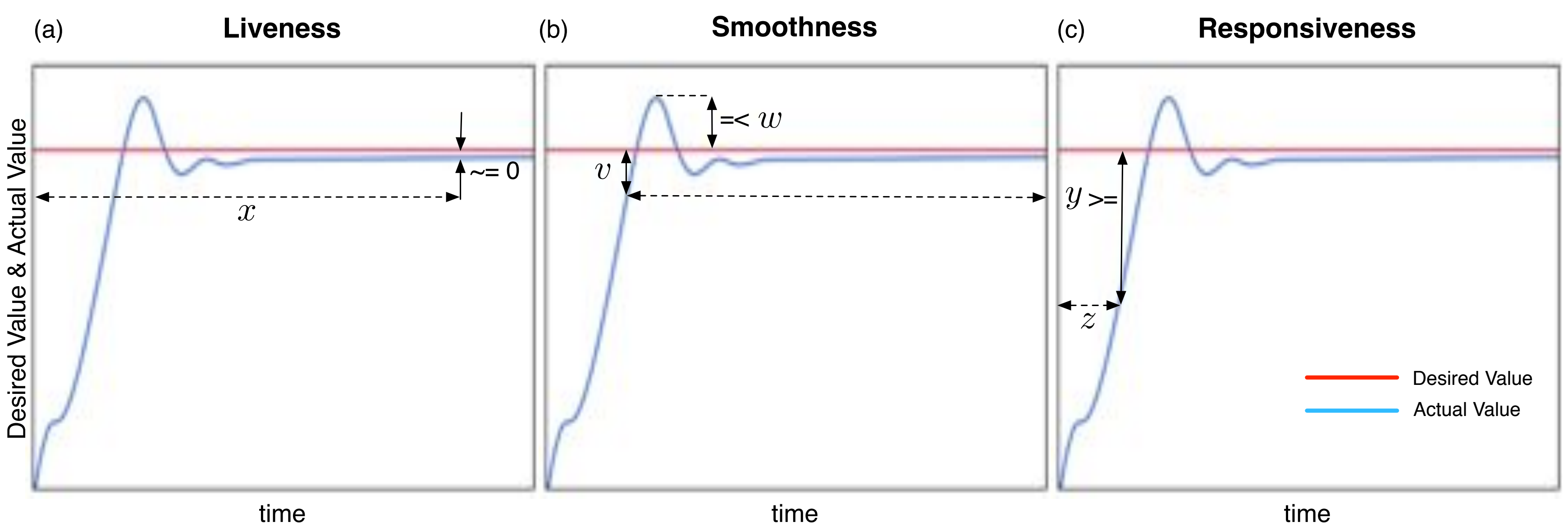
***Validation in an artificial setting may have limited value or may be impossible***

• Artificial validation is not useful if benchmarks are non-existent or are found to be unsuitable
• Contextual factors and level of complexity of the benchmarks should be a match for the project
• Meaningful artificial validation may even be impossible!

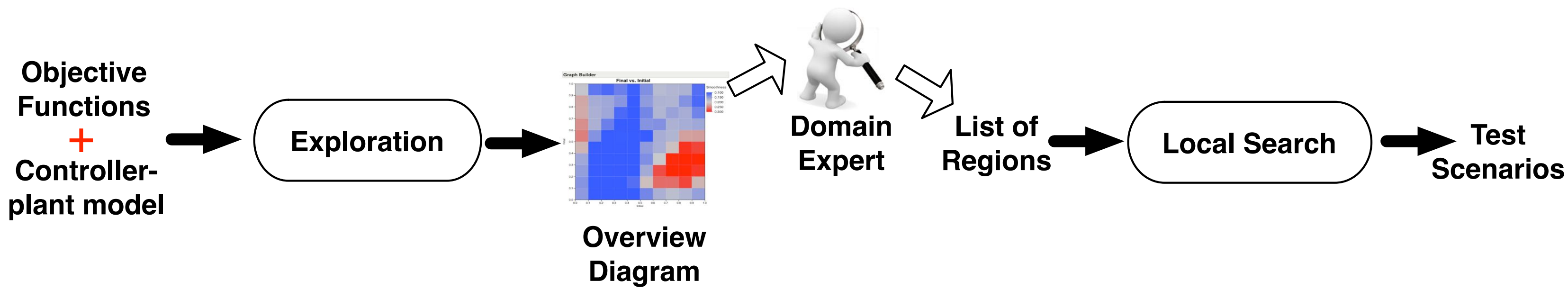
**Example: MiL and HiL Testing ECU control software**
• MiL testing requires actual Matlab/Simulink models
• Realistic search performance and fault detection
• HiL testing: dedicated hardware

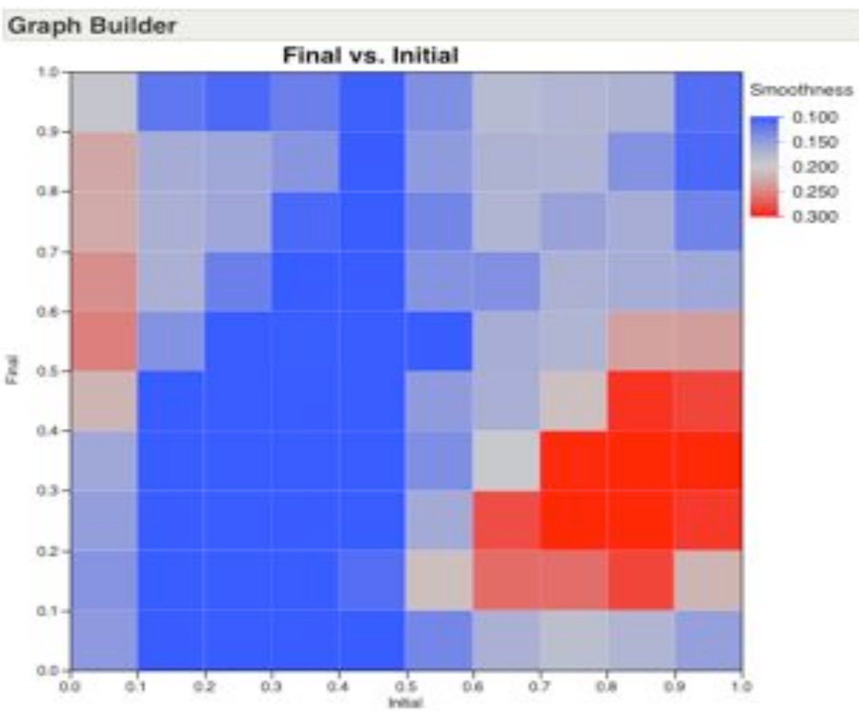
# Testing Controller and Plant Simulink Models

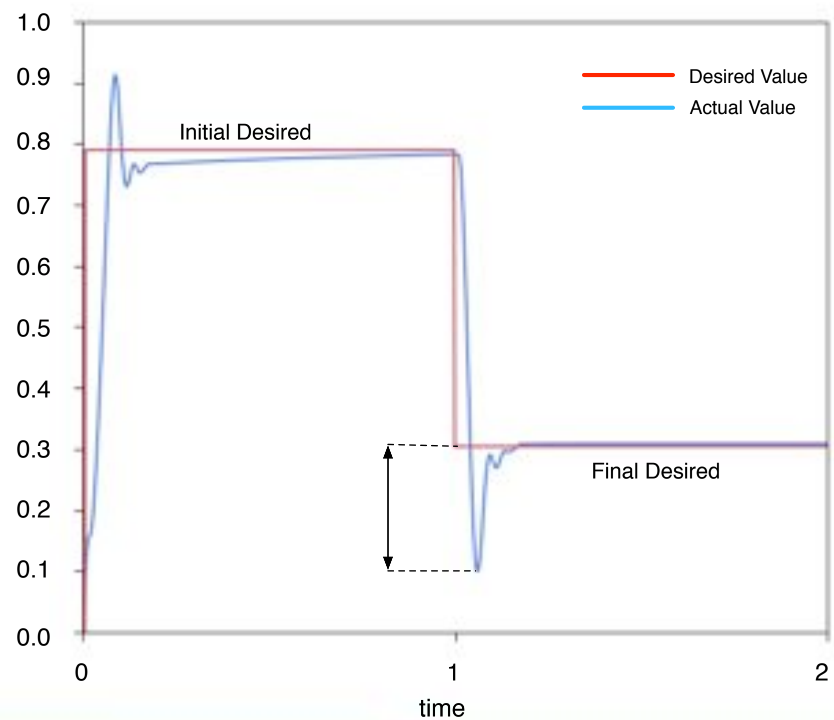


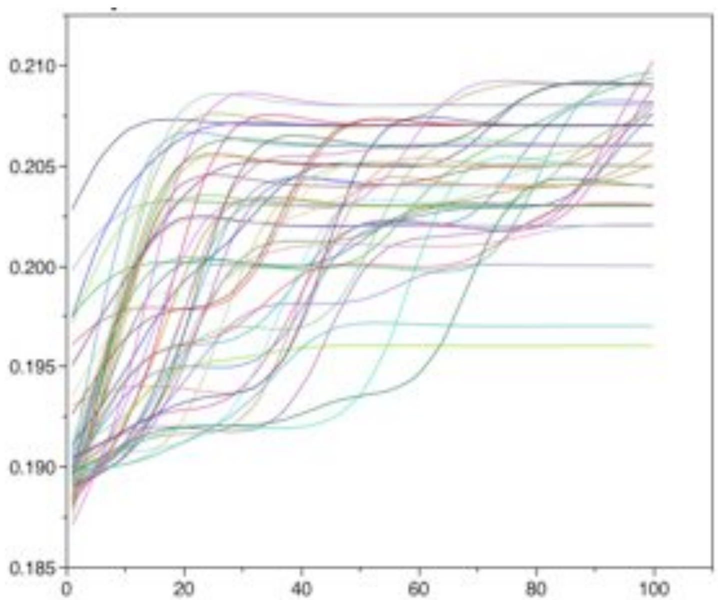
# MiL-Testing of Continuous Controllers



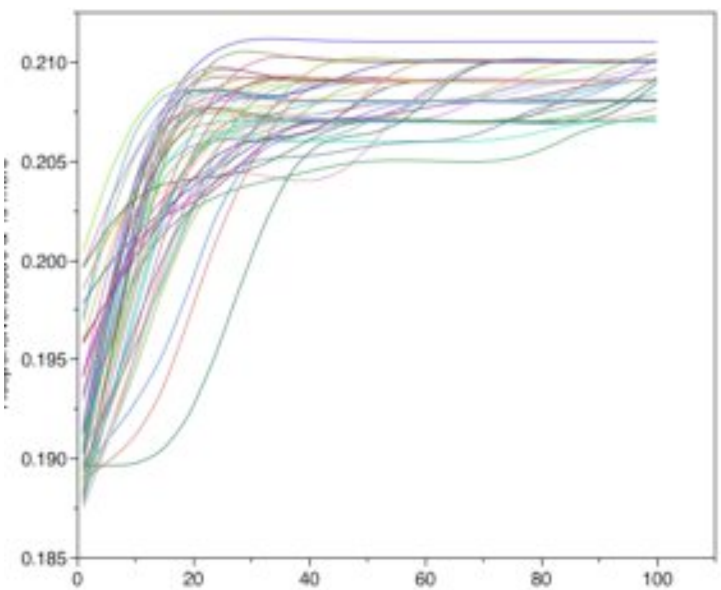



# Random Search vs. (1+1)EA
# Example with Responsiveness Analysis




Random




(1+1) EA

Random search might have done well with a simple artificial example

# Choosing pilot studies

***A good pilot study should be (1) representative, (2) feasible, and (3) relevant to current needs***

- **Representative**: reflective of the characteristics of the industry partner's systems
- **Feasible**: commensurate with pilot study resources
- **Relevant:** dealing with ongoing activities, planned future activities, or past activities with a horizon for future reuse
- **Quick Research impact!**

**Example:**

- Data Acquisition System: system currently under maintenance, transmitted files in typical range, real test suite
- ECU software: typical closed loop controller, being tested, in terms of configuration parameters
- Satellite ground control: Requirements were being defined, verified, and modified

# Mentorship

***Lesson: Mentor by doing!***

- Be ready to provide a lot of help during realistic validation
  - To show commitment and set a good example
- Mentoring is paramount if the partner is not using MDE regularly
- Once practitioners become proficient, researchers' help should be phased out

**Example: Data Acquisition System**

- **Modeling tutorials**
- **Modeled an actual DAS with their help**
- **Mentoring on modeling other DAS**

# Drumming up support

***Lesson: Find internal champions for the solution.***

- One or more champions are needed to:
  - spread the word about the research
  - connect the dots for the management
  - develop a strategy for integrating the solution
- Champions need to have developed a strong sense of trust in the researchers and the research
- Champions are usually people with a genuine intellectual interest in the topic
- You need to support them and make them excited and proud about what they do. Show your appreciation.
- It may take time to find the right person in the organization

# Managing the relationship with industry partners

- Focus must be kept on recurring, long-lasting, and unsolved problems

- **Expectations that cannot be met**
  - Working as a consultant
  - Building professional tools
- **Avoid short-term partnerships**
  - At least 3 years of commitment to be expected
    - Bare minimum to conclude the research for a PhD
  - This usually tips the balance in favor of collaboration with larger organizations
- **Communicate and follow up frequently**
  - If you have not talked to a partner company in 3 months then something is wrong!

# Publishing the results

- Possible tension between the research model and publication:
  - Industry is interested in end-to-end solutions
  - A solution often has different components, each belonging to a different (SE) research community
  - Not always easy to determine how to report and what venue to aim for
  - Example: Metaheuristic search and MDE in Delphi example on CPU balancing
- More thorny issue: interdisciplinary results!
- Many real-world problems are at a **systems** level
  - Software is only one part and may be difficult to isolate from hardware and mechanical devices
  - Interdisciplinary work is still notoriously hard to publish!
  - More room is needed for systems engineering research

# FAQ

- Aren’t your research results specific to an industry partner? How do they generalize?
  - SE solutions are generally not applicable across domains
  - Partners are not unique, they capture the practices of an industry
  - We are better off with solutions that apply to a domain, than solutions that have not shown to apply anywhere
  - Generalization comes from replications
  - Make your working assumptions explicit
- Aren’t we constraining our creativity as researchers? Shouldn’t we focus on “good” idea?
  - In engineering, it is as “good” as it works
  - Everything else is marketing or academic hype
- Shouldn’t we work, as researchers, on future problems the industry will face?
  - How can anybody claim to know future needs without understanding current ones?

# Summary

- Introduced a research paradigm & process, and general lessons learned that can be useful to other researchers
- Research is coupled with knowledge transfer
- It promotes intertwining of research and industrial innovation to increase the impact of research
- Emphasis is placed on early involvement of industry
  - This increases chances of impact and adoption, mentoring opportunities, and creates a sense of ownership
- Work to be viewed as a step towards reducing the gap between software engineering research and practice

Thank you!

Questions?

**Contact Info:**

Lionel Briand
lionel.briand@uni.lu

Software Verification and Validation Laboratory (http://www.svv.lu)
Centre for ICT Security, Reliability, and Trust (SnT)
University of Luxembourg

# References on MDE Adoption and Transfer


- [Mohagheghi & Dehlen 2008]: P. Mohagheghi, V. Dehlen. Where Is the Proof? - A Review of Experiences from Applying MDE in Industry. ECMDA-FA 2008.
- [Hutchinson et al. 2011]: J. Hutchinson, J. Whittle, M. Rouncefield, S. Kristoffersen. Empirical assessment of MDE in industry. ICSE 2011.
- [Selic 2012]: B. Selic. What will it take? A view on adoption of model-based methods in practice. Softw Syst Model, 2012.
- [Gorschek et al 2006]: T. Gorschek, P. Garre, S. Larsson, C. Wohlin. A Model for Technology Transfer in Practice. IEEE Software 23(6), 2006.
- [Davison et al 2004]: R. Davison, M. Martinsons, N. Kock. Principles of canonical action research. Information Systems Journal (14), 2004.

# Selected Personal References


- L. Briand, Y. Labiche, and M. Shousha, “Using genetic algorithms for early schedulability analysis and stress testing in real-time systems”, Genetic Programming and Evolvable Machines, vol. 7 no. 2, pp. 145-170, 2006
- M. Shousha, L. Briand, and Y. Labiche, “UML/MARTE Model Analysis Method for Uncovering Scenarios Leading to Starvation and Deadlocks in Concurrent Systems”, IEEE Transactions on Software Engineering 38(2), 2012.
- Z. Iqbal, A. Arcuri, L. Briand, “Empirical Investigation of Search Algorithms for Environment Model-Based Testing of Real-Time Embedded Software”, ACM ISSTA 2012
- S. Nejati, S. Di Alesio, M. Sabetzadeh, L. Briand, “Modeling and Analysis of CPU Usage in Safety-Critical Embedded Systems to Support Stress Testing”, ACM/IEEE MODELS 2012
- Shiva Nejati, Mehrdad Sabetzadeh, Davide Falessi, Lionel C. Briand, Thierry Coq, “A SysML-based approach to traceability management and design slicing in support of safety certification: Framework, tool support, and case studies”, Information & Software Technology 54(6): 569-590 (2012)
- Lionel Briand et al., “Traceability and SysML Design Slices to Support Safety Inspections: A Controlled Experiment”, forthcoming in ACM Transactions on Software Engineering and Methodology, 2013

# Selected Personal References (cont.)


- Rajwinder Kaur Panesar-Walawege, Mehrdad Sabetzadeh, Lionel C. Briand: Supporting the verification of compliance to safety standards via model-driven engineering: Approach, tool-support and empirical validation. Information & Software Technology 55(5): 836-864 (2013)
- Razieh Behjati, Tao Yue, Lionel C. Briand, Bran Selic: SimPL: A product-line modeling methodology for families of integrated control systems. Information & Software Technology 55(3): 607-629 (2013)
- Hadi Hemmati, Andrea Arcuri, Lionel C. Briand: Achieving scalable model-based testing through test case diversity. ACM Trans. Softw. Eng. Methodol. 22(1): 6 (2013)
- Nina Elisabeth Holt, Richard Torkar, Lionel C. Briand, Kai Hansen: State-Based Testing: Industrial Evaluation of the Cost-Effectiveness of Round-Trip Path and Sneak-Path Strategies. ISSRE 2012: 321-330
- Razieh Behjati, Tao Yue, Lionel C. Briand: A Modeling Approach to Support the Similarity-Based Reuse of Configuration Data. MoDELS 2012: 497-513
- Shaukat Ali, Lionel C. Briand, Andrea Arcuri, Suneth Walawege: An Industrial Application of Robustness Testing Using Aspect-Oriented Modeling, UML/ MARTE, and Search Algorithms. MoDELS 2011: 108-122