Model-based Approach for Co-optimization of Safety and Security Objectives in Design of Critical Architectures

Kunal Suri [1], Gabriel Pedroza [1], and Patrick Leserf [2]

[1] Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

[2] ESTACA, 12 Rue Paul Delouvrier, Montigny-le-Bretonneux 78180, France

INTERNATIONAL CONFERENCE ON MODEL AND DATA ENGINEERING (MEDI) 2021

Outline

- Introduction
- Research Context & Motivation
- Contribution: Model-based Co-optimization of Objectives
- Experimentation
- Conclusion & Perspectives

Outline

- **Introduction**
- Research Context & Motivation
- Contribution: Model-based Co-optimization of Objectives
- Experimentation
- Conclusion & Perspectives

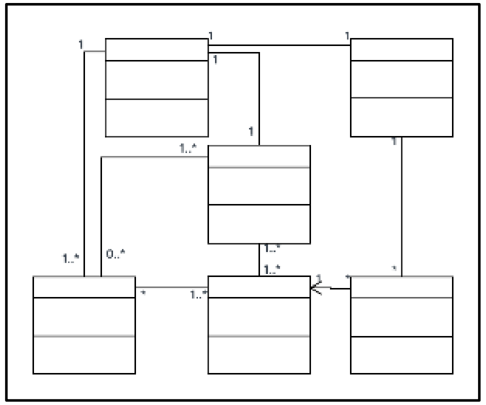
Introduction (1/2)

- **Need for safety objectives in Cyber-Physical Systems (CPS)**? Systems must be safe for humans to use
- **Need for security objectives in CPS**? Heavy dependence on software + highly interconnect, thus a possibility of cyber attack(s)
  - Impacts the safety of systems and its end users
- **Need for multi-objectives analysis**? To perform analysis of both safety & security objectives
  - To understand how the objectives related to security will effect safety features (and vice-versa)
  - Commonalities in both analysis, thus avoidance of duplication of efforts
  - Simple example ***Keyless Car Entry***: If the encryption level is increased (security feature), then it will effect the time taken to decrypt and to send an ACK, thus effecting overall performance (safety feature)

Introduction (2/2)

- **How Model-Driven Engineering (MDE) can help to address safety & security concerns**?
  - MDE supports creation of a coherent model of a system that may be augmented with various relevant information for different stakeholders
    - This model when transformed into different formats allows representing various formalization relevant for different domains
  - MDE provides *principles*, *standards* (e.g. OMG SysML) and *tools* (e.g. Eclipse Papyrus)

MDE



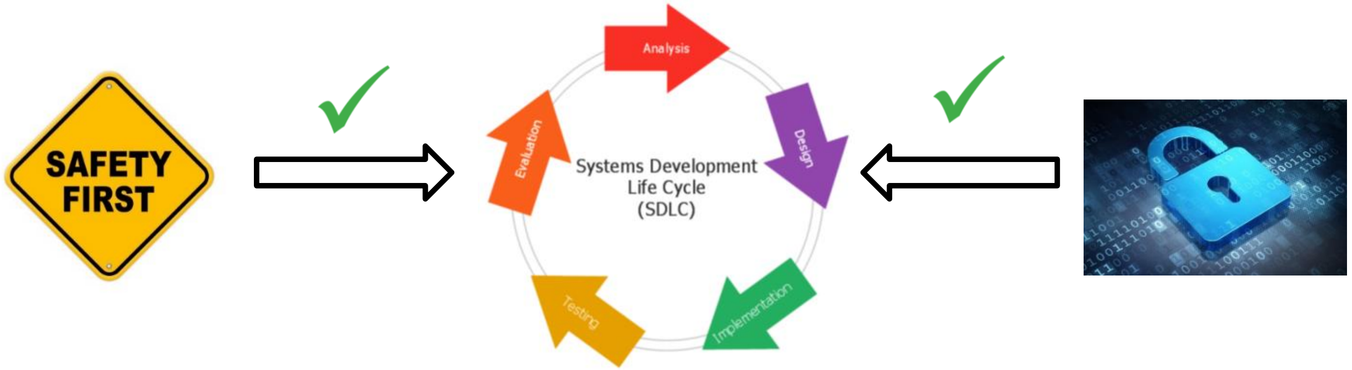
Outline

- Introduction
- **Research Context & Motivation**
- Contribution: Model-based Co-optimization of Objectives
- Experimentation
- Conclusion & Perspectives

Research Context

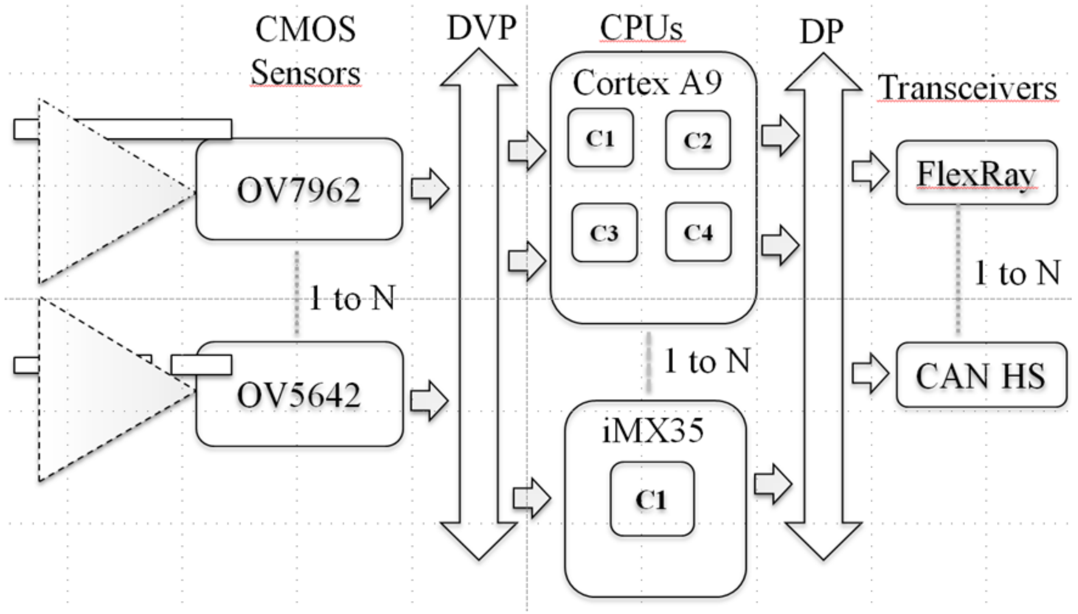
To support efficient modeling of both safety & security objectives during the design phase of the SDLC, i.e., to support Design Space Exploration (DSE)

- Safety & Security objectives can be **conflicting** but **measurable** (Performance, Breakability, Cost) and it is essential to obtain an **acceptable trade-offs** (*Co-Optimize*)
  - E.g. to minimize a vector of objective functions (such as the cost and failure rate) defined by the designer
- To bridge the gap for safety-security co-optimization, which is dominantly due to the unavailability of specific methods and frameworks (language, vocabulary) along with tools
- To support non-savvy engineers without them needing to be an expert in both domains



Research Motivation (1/2)

Motivating Example: A Safety & Security sensitive CPS

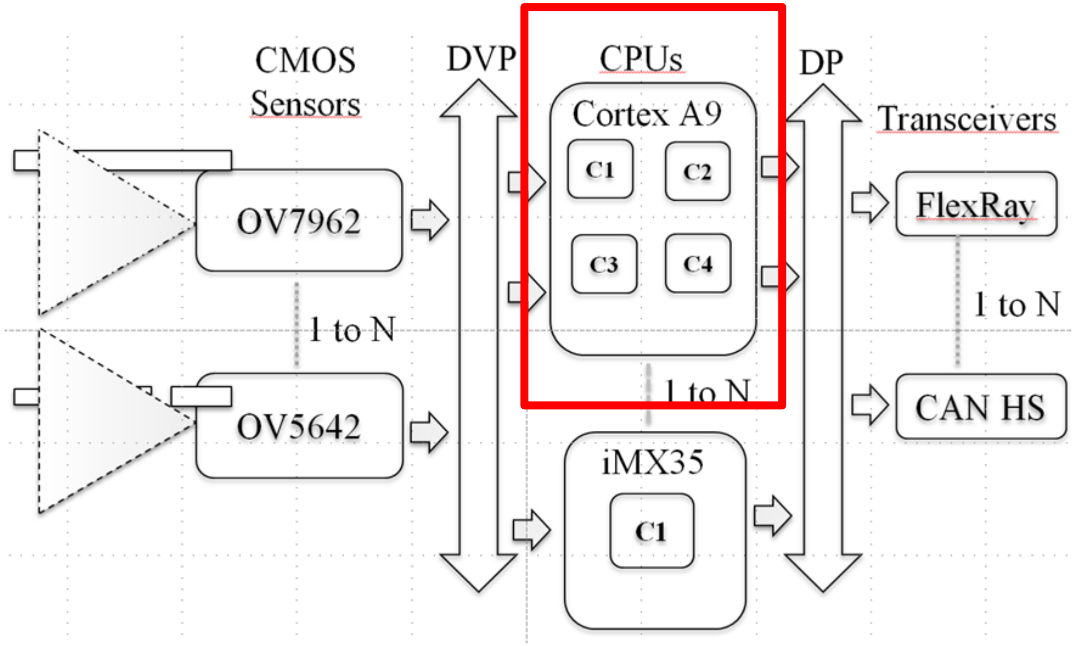


Embedded Cognitive Safety System (ECSS)

- ECSS is used in Automotive safety system
  - CMOS sensors, CPU and embedded networks (FlexRay, CAN)
- ECSS needs to be fail-safe (having objectives such as performance, rate of failure)
  - Safety can be effected by security concerns (such as CAN attacks, other attacks on camera)

Research Motivation (2/2)

Design Space Exploration problem



Embedded Cognitive Safety System (ECSS)

- To select HW components (i.e., **redundancy level**)
- To find the optimal solution having the acceptable trade-off for safety & security concerns along with their effect on cost

Brief describing of the design-space complexity for designing HW architectures

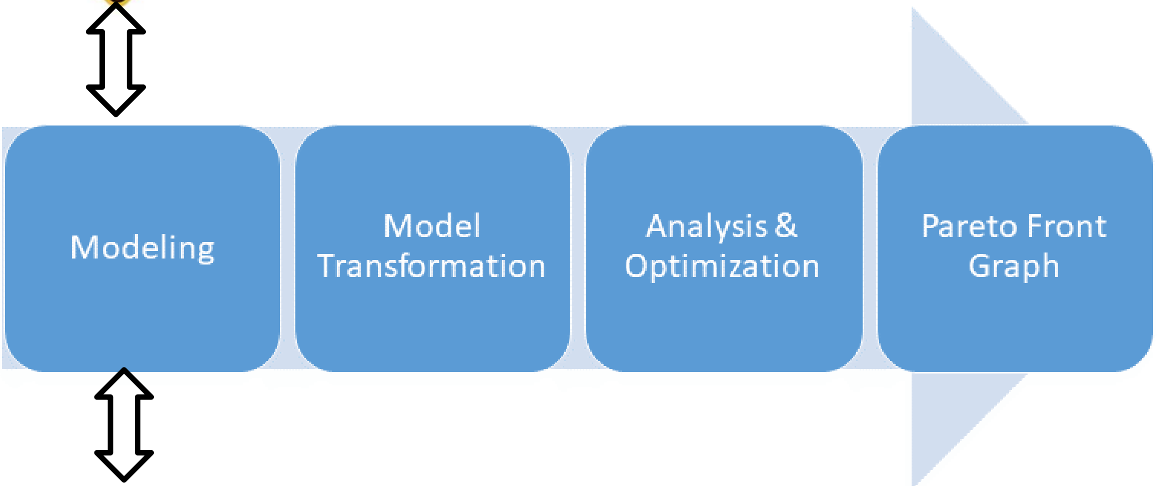
Possible type of variability:

- **No Variability:** Occupied Slots = 1 → Only 1 Samsung OR 1 STMicro chip
- **Instance Variability:** Occupied Slots = 2 → Both 1 Samsung AND 1 Samsung chip
- **Component Variability:** Occupied Slots = 2 → 1 Samsung AND 1 STMicro chip
- **Mixed Variability (both Component + Instance):** Occupied Slots = 3 → 1 Samsung AND 1 Samsung AND 1 MediaTek chip

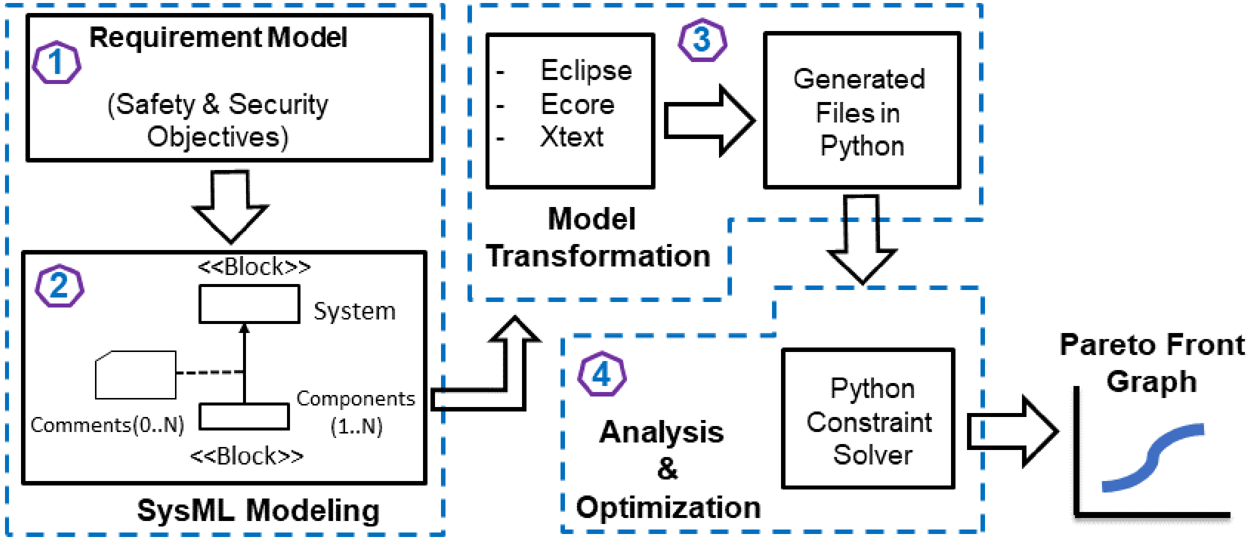
Outline

- Introduction
- Research Context & Motivation
- **Contribution: Model-based Co-optimization of Objectives**
- Experimentation
- Conclusion & Perspectives

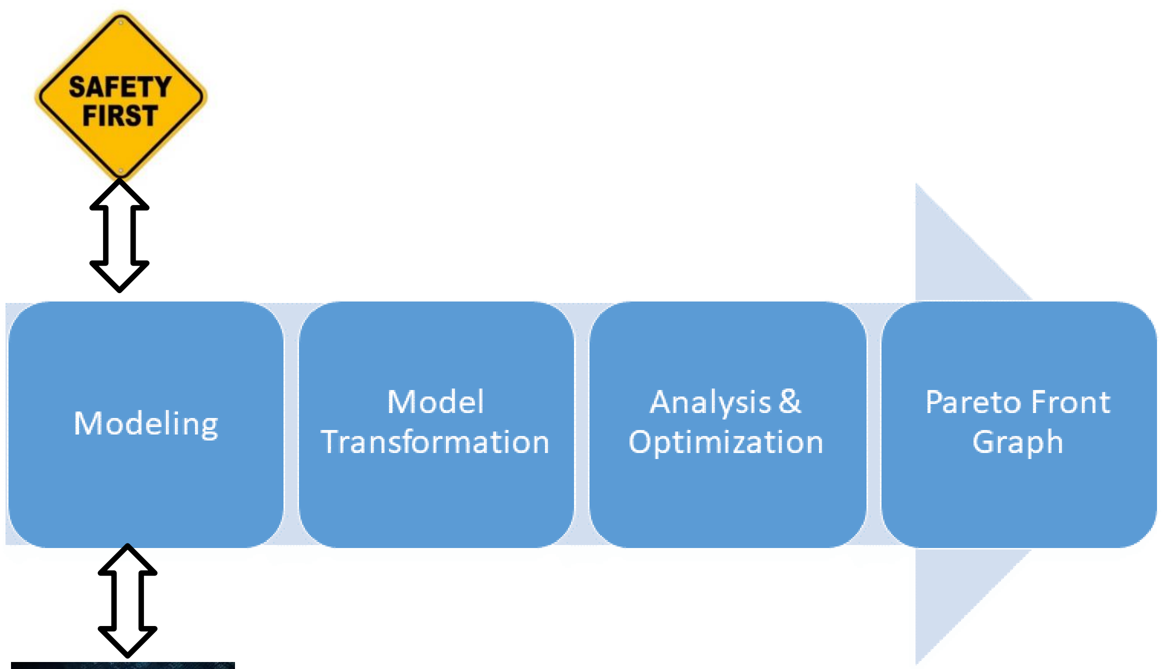
Contribution: Model-based Co-optimization of Objectives



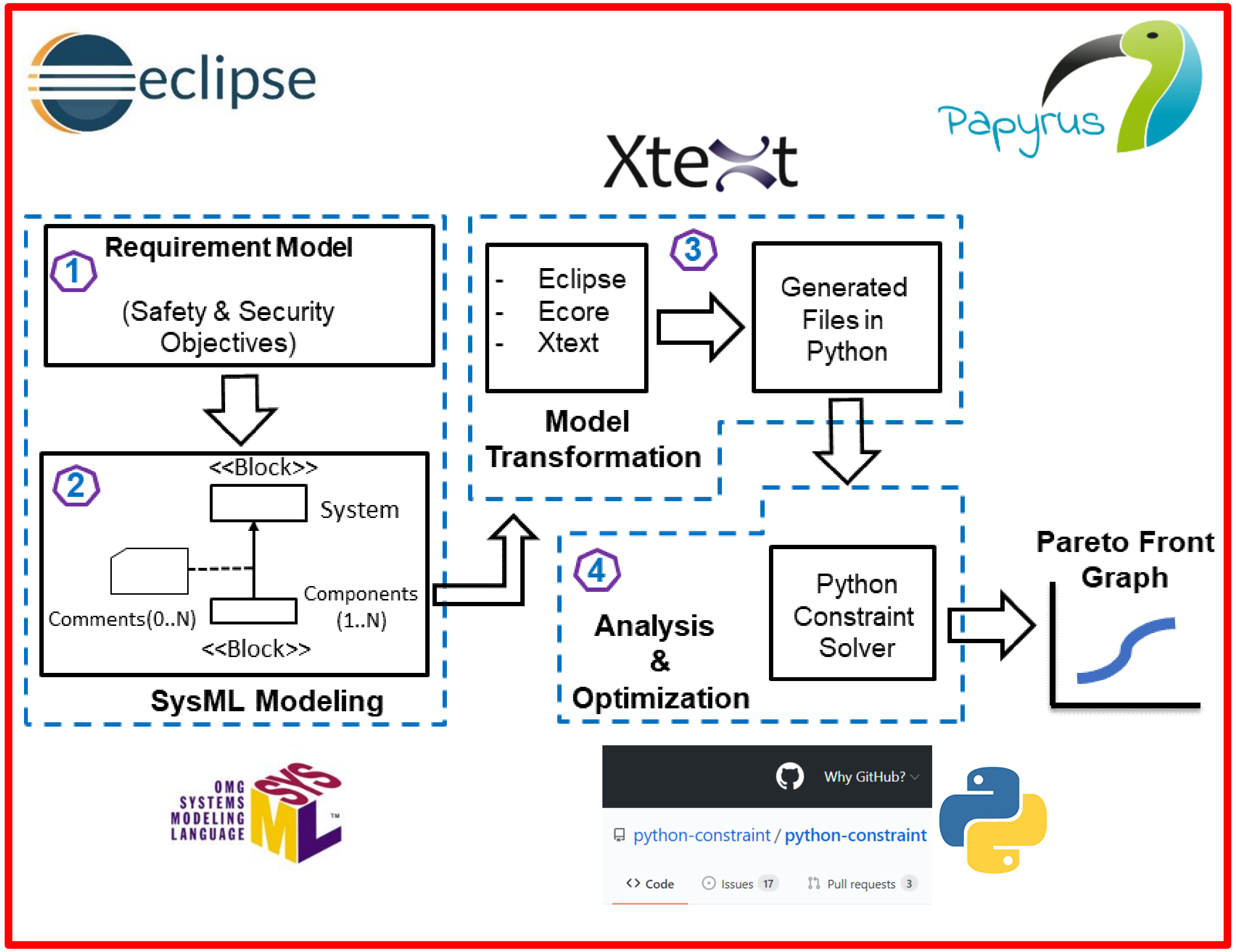
Overall Method



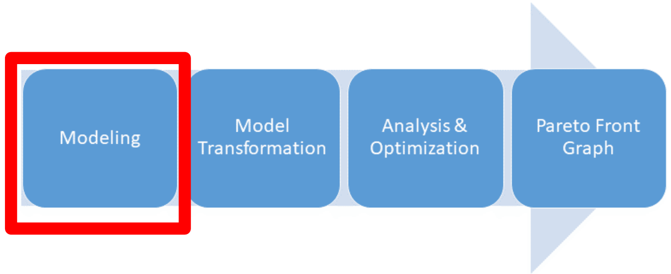
Contribution: Model-based Co-optimization of Objectives

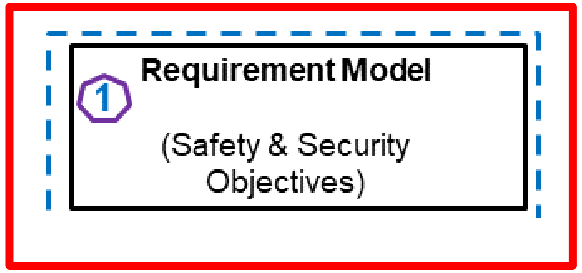


Overall Method

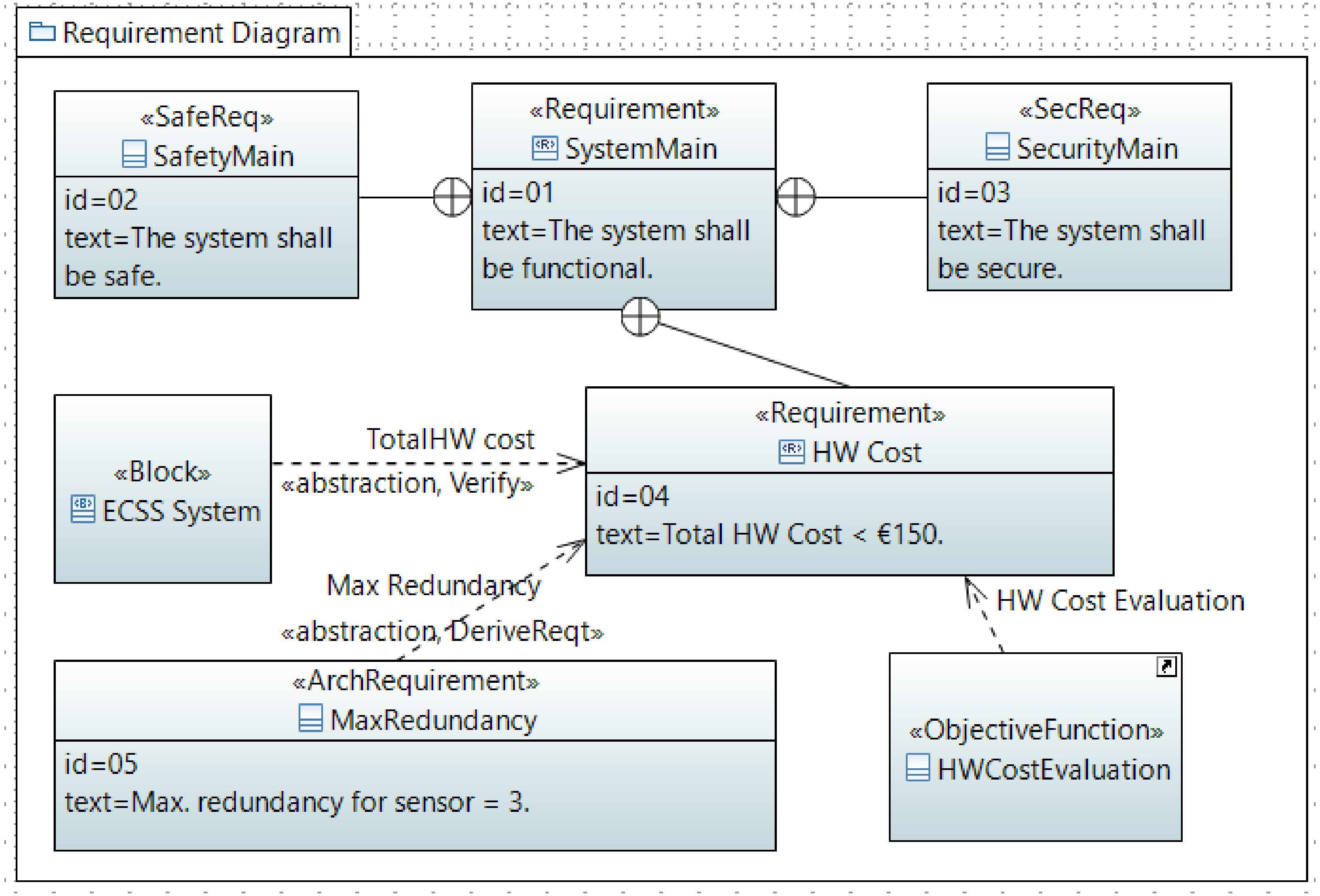


Modeling (1/2)

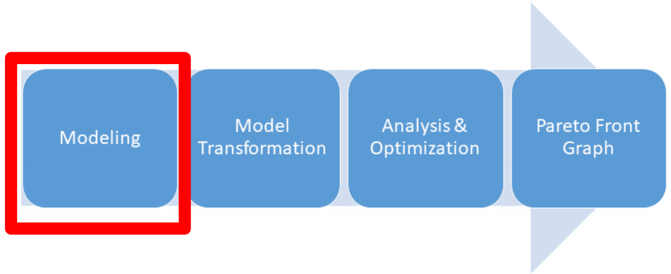


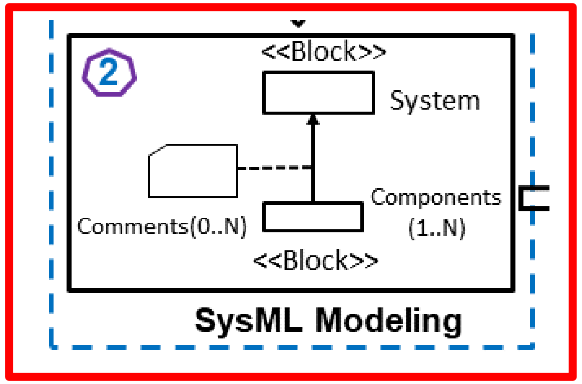


- Modeling general system requirements & relationships
- Modeling Safety & Security requirements via introduction of new stereotypes
  - <<SafeReq>> & <<SecReq>>

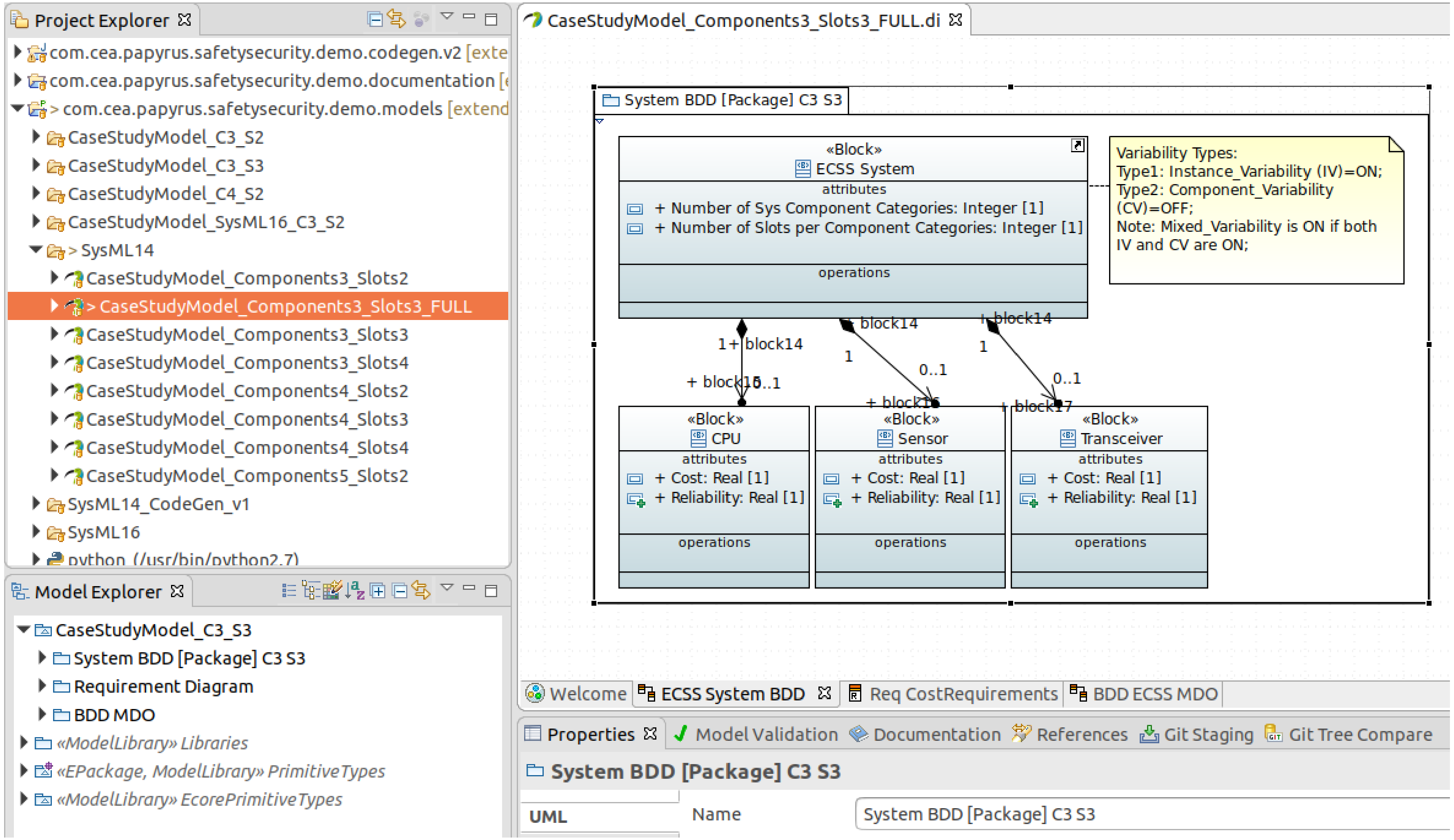


Modeling (2/2)

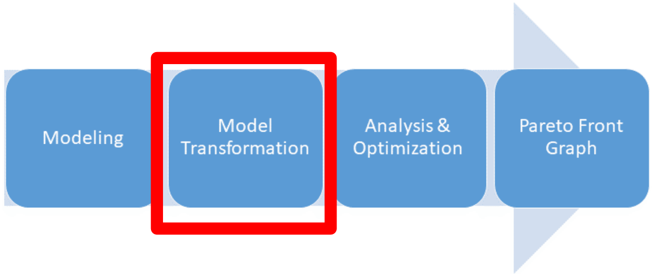


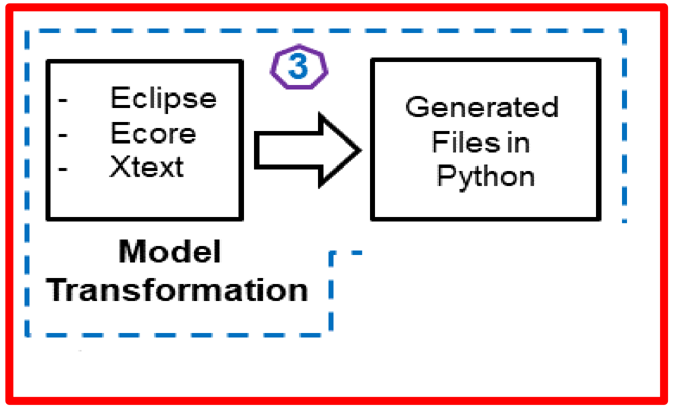


- Modeling the system & relationships via BDD
- Input values such as cost, reliability etc.
- Input the type of constraint needed, i.e., variability type

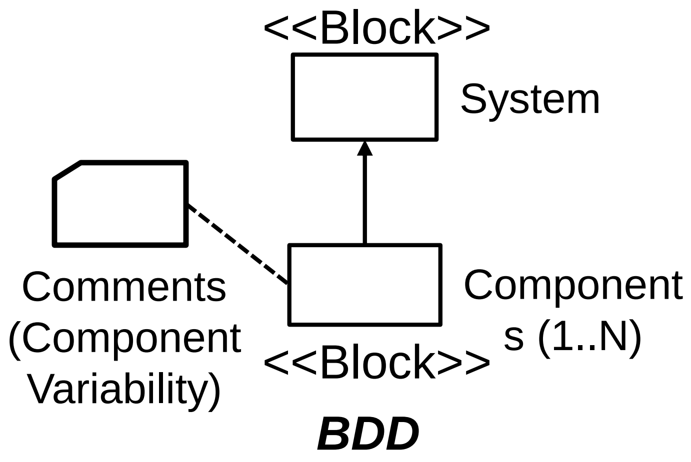


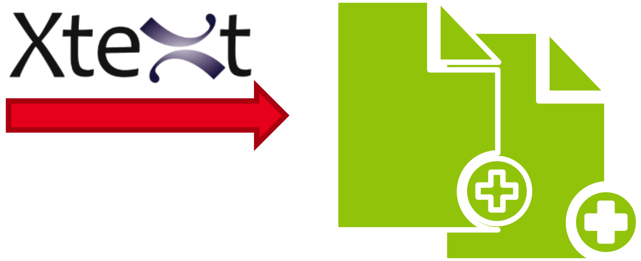
Model Transformation (1/2)





UML is useful for modeling system information (including relationships) but not for conducting mathematical analysis. Thus, model transformation is needed.





Set of generated files in Python format

Component Variability

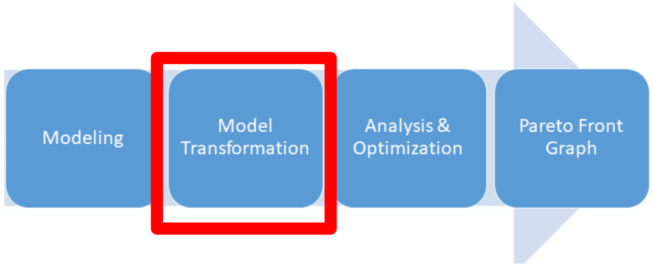
Input: Parent class, Child class, Parameters, comments about component variability

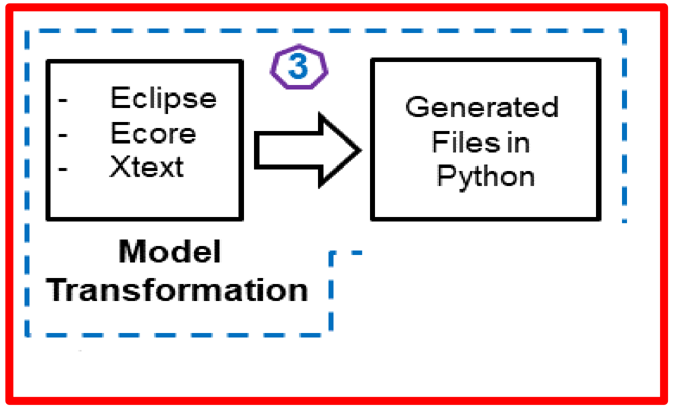
Output: Python File → BDD [Package] ECSS.py

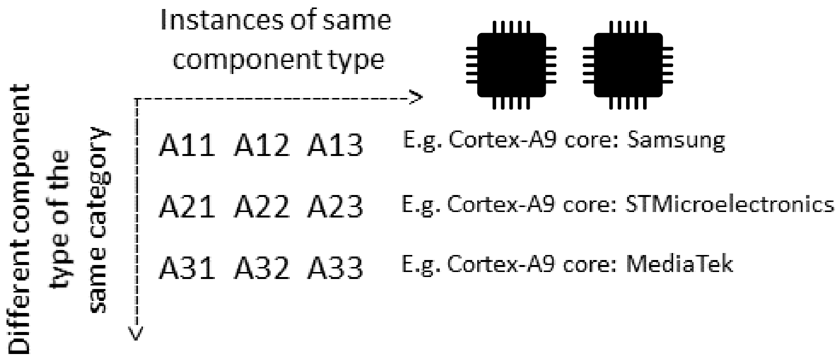
- Python script with CSP formalization based on input values

I	O
Model with comments about component variability associated with the child class blocks	- Python File - Execution of Python code results in possible solution and their graphical representation

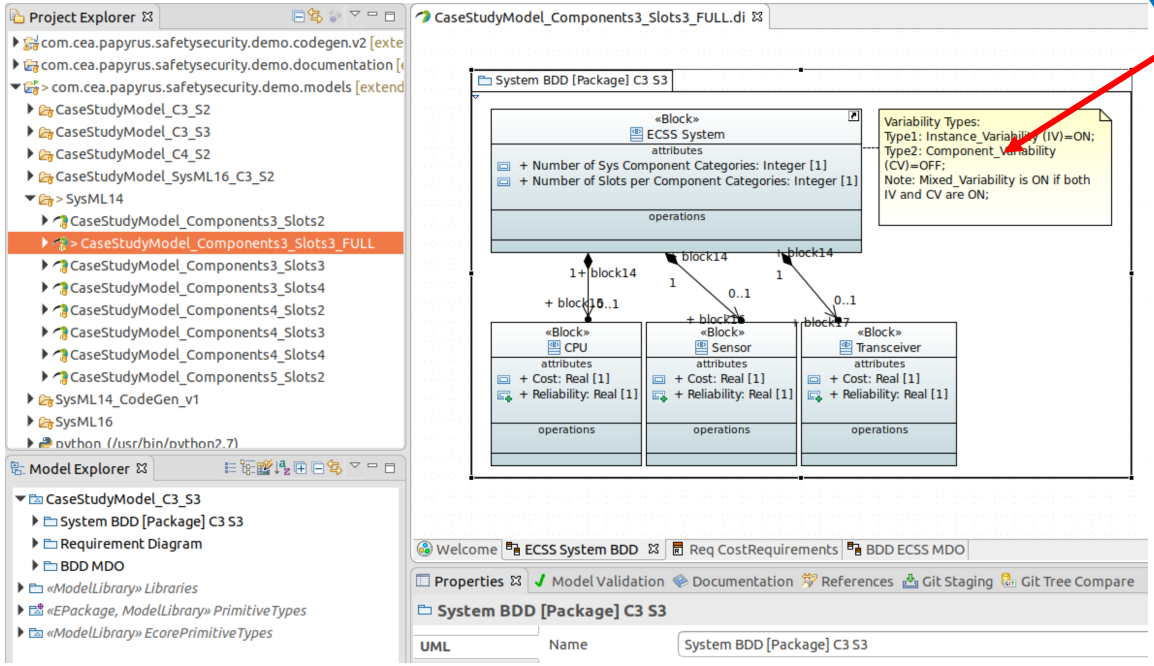
Model Transformation (2/2)







Provide variability type via Comments

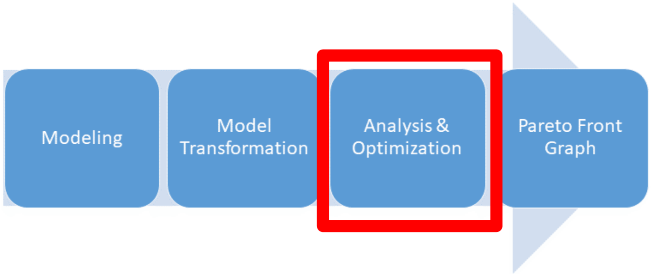


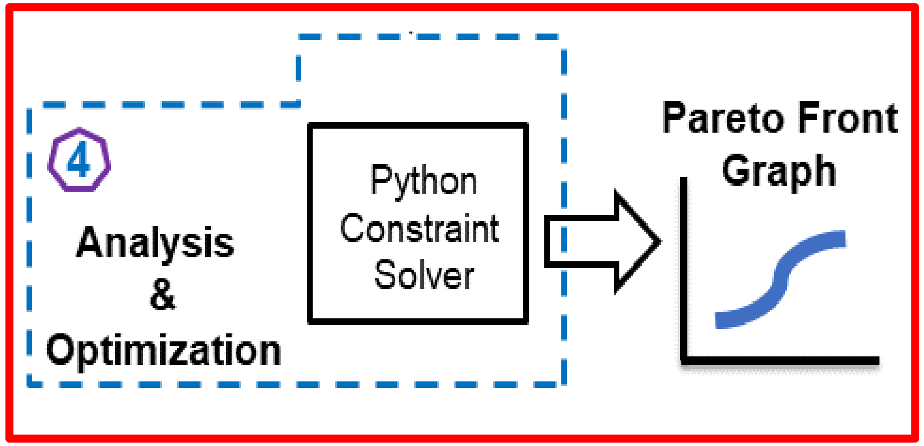
Xtext

Code Generation (Model to Text)

```
Example:
Solve the a + b = 5; a  b = 6 algebraic relation.
from constraint import *
problem = Problem()
problem.addVariable('a', range(5))
problem.addVariable('b', range(5))
problem.addConstraint(lambda a, b: a + b == 5)
problem.addConstraint(lambda a, b: a * b == 6)
solutions = problem.getSolutions()
print solutions
```

Analysis & Optimization

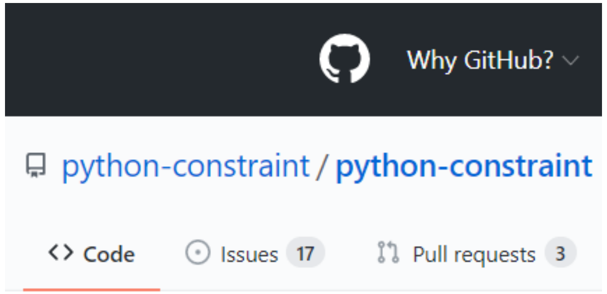


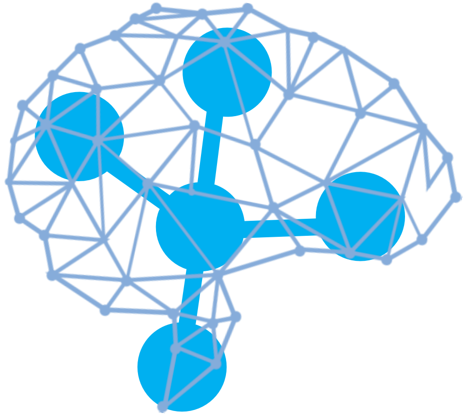


Python script having constraints

```
Example:
Solve the a + b = 5; a  b = 6 algebraic
relation.
from constraint import *
problem = Problem()
problem.addVariable('a', range(5))
problem.addVariable('b', range(5))
problem.addConstraint(lambda a, b: a + b == 5)
problem.addConstraint(lambda a, b: a * b == 6)
solutions = problem.getSolutions()
print solutions
```

Constraints Solver

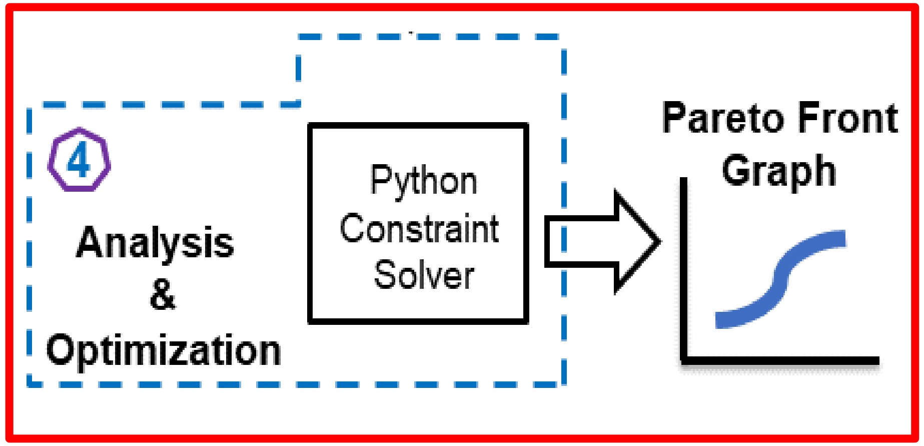




Set of Solutions

[{'a': 3, 'b': 2}, {'a': 2, 'b': 3}]

Pareto Front

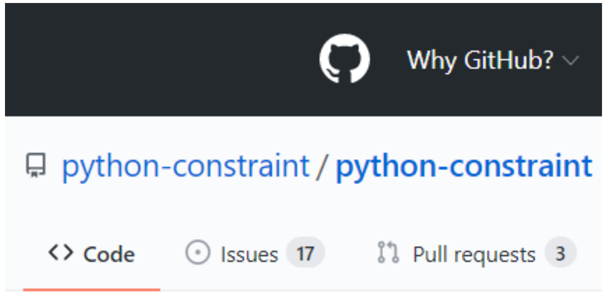


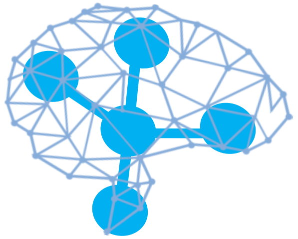
Python script having constraints

Example:
Solve the a + b = 5; a b = 6 algebraic relation.

```
from constraint import *
problem = Problem()
problem.addVariable('a', range(5))
problem.addVariable('b', range(5))
problem.addConstraint(lambda a, b: a + b == 5)
problem.addConstraint(lambda a, b: a * b == 6)
solutions = problem.getSolutions()
print solutions
```

Constraints Solver

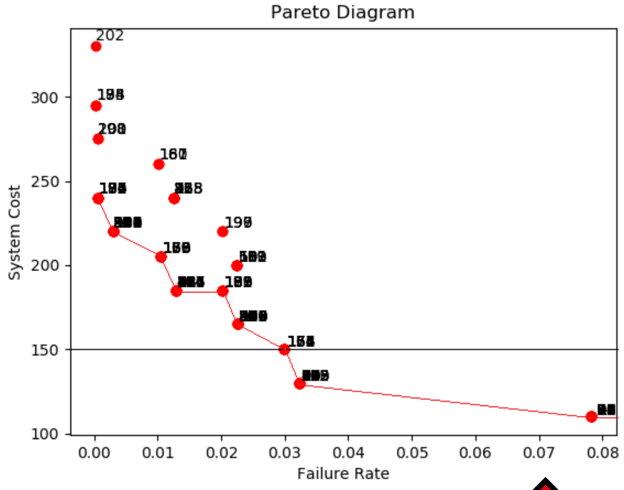




Set of Solutions

[{'a': 3, 'b': 2}, {'a': 2, 'b': 3}]

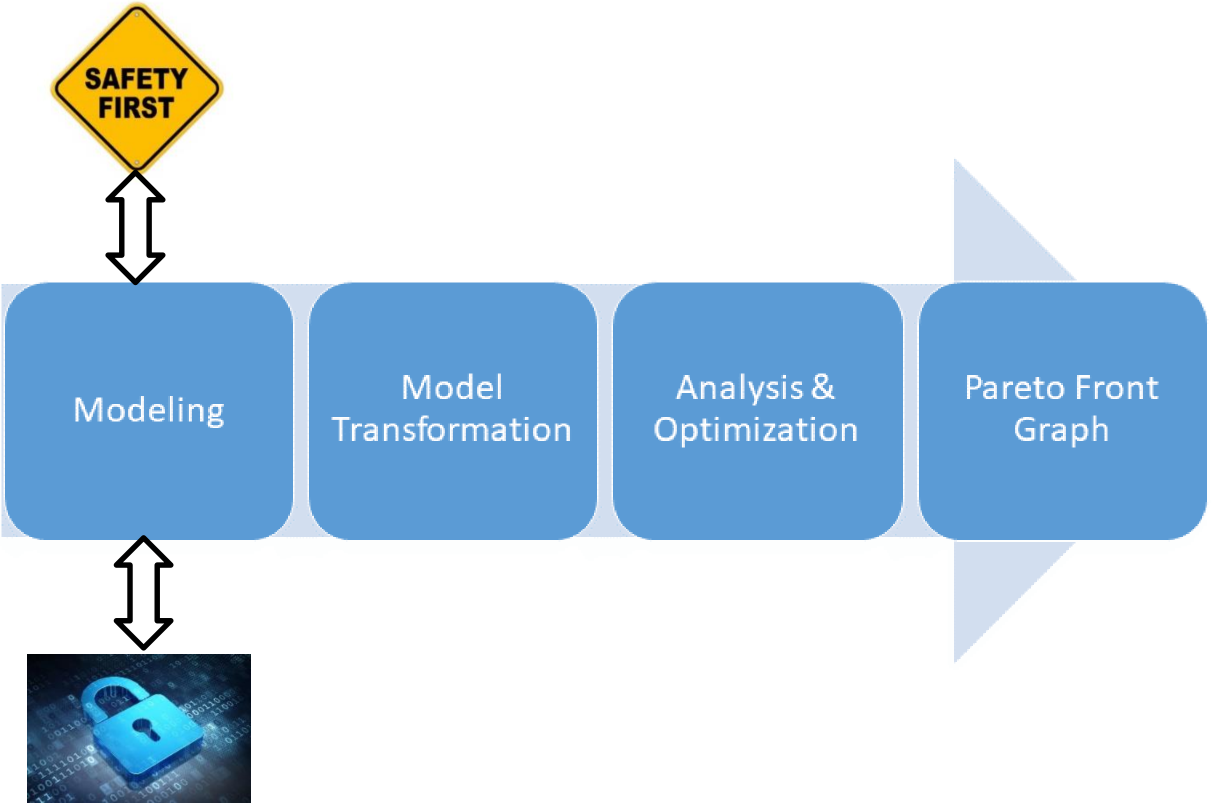
Pareto Front



Outline

- Introduction
- Research Context & Motivation
- Contribution: Model-based Co-optimization of Objectives
- **Experimentation**
- Conclusion & Perspectives

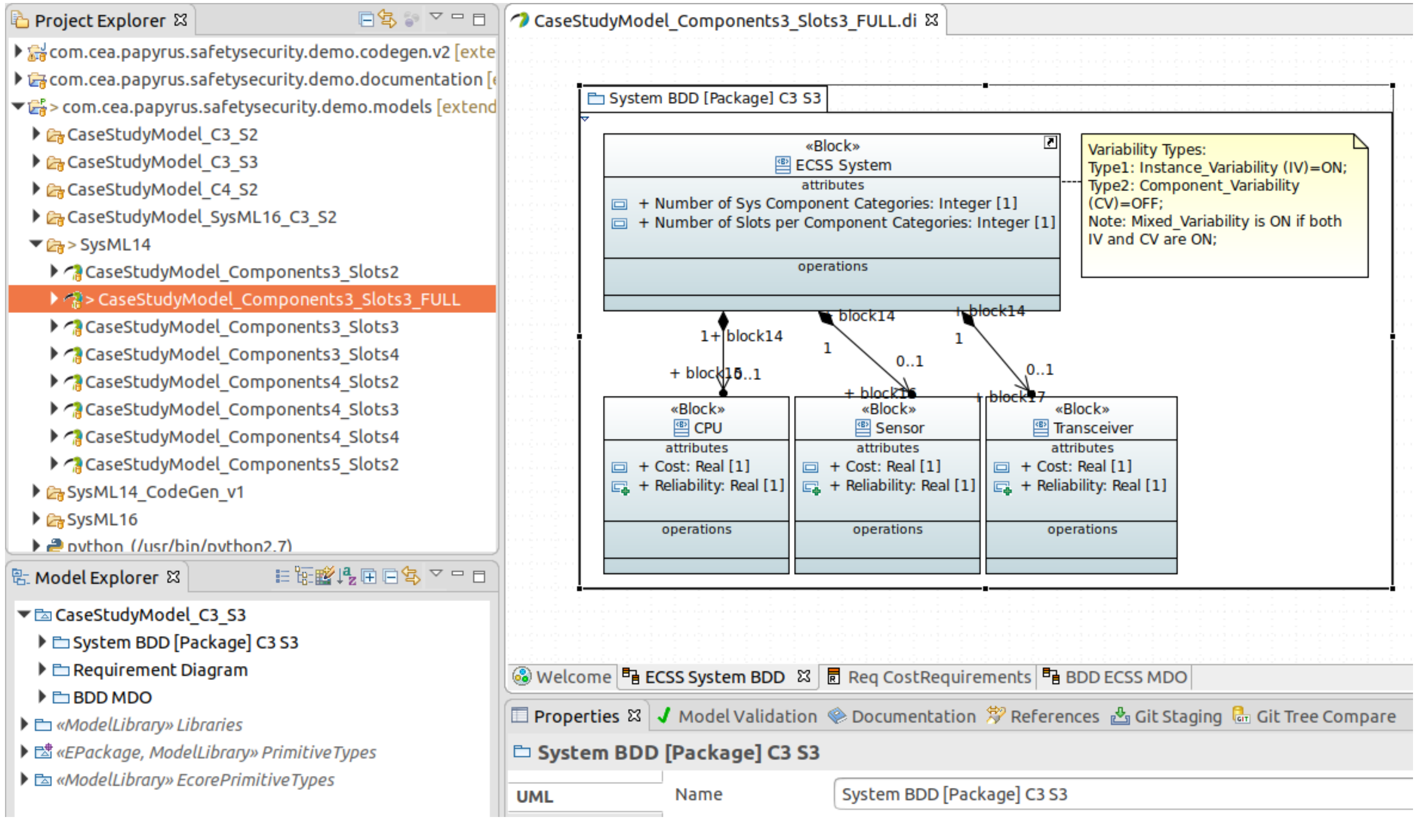
Experimentation (1/4)

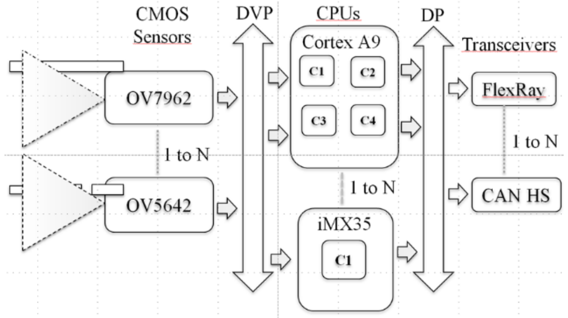


Experimentation Setup

- Eclipse Papyrus Framework
  - SysML Modeling
  - Code generation [Papyrus Designer]
- Python constrain solver

Experimentation (2/4)

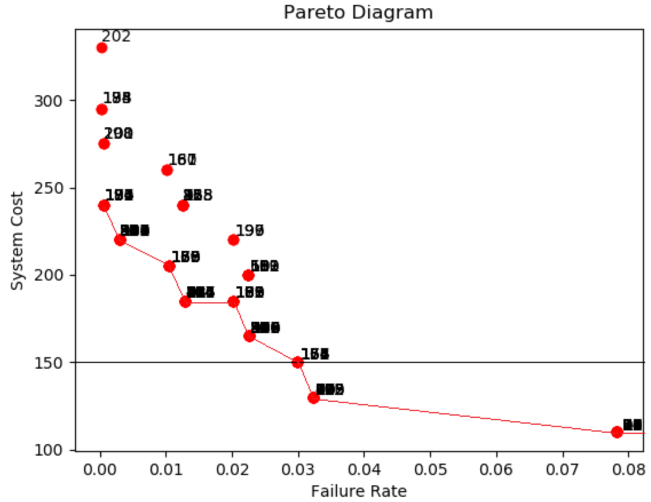




Code Generation (Model to Text)

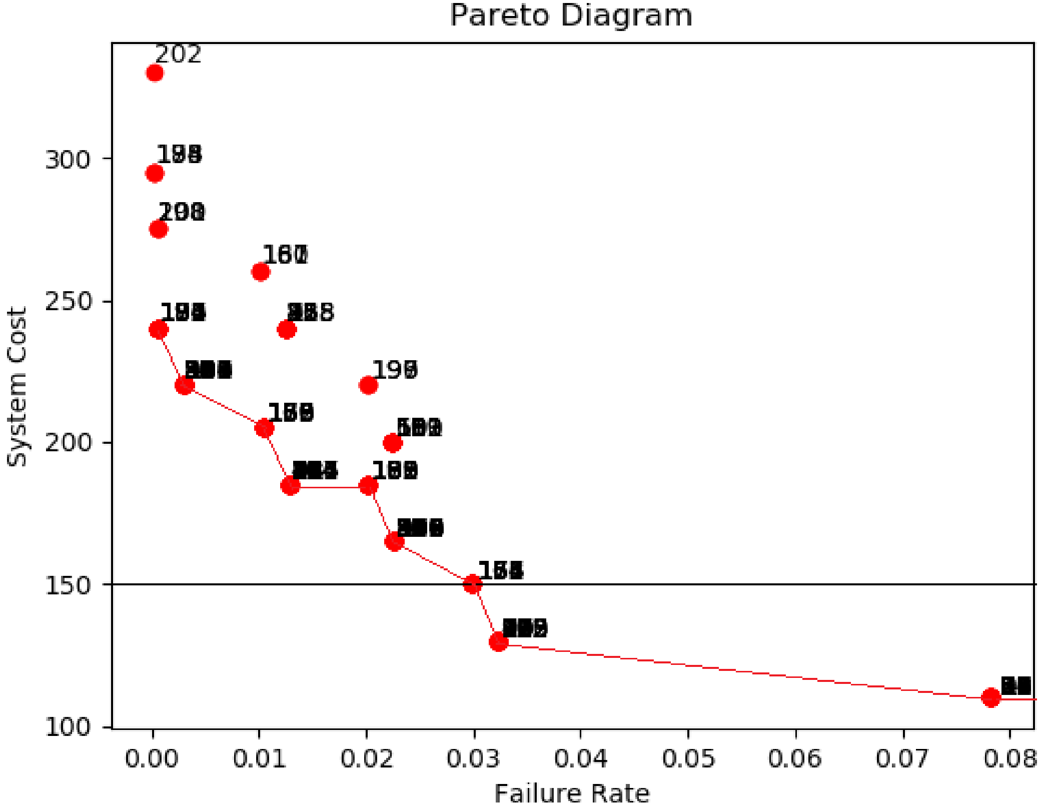
Set of generated files in Python

CSP Solver

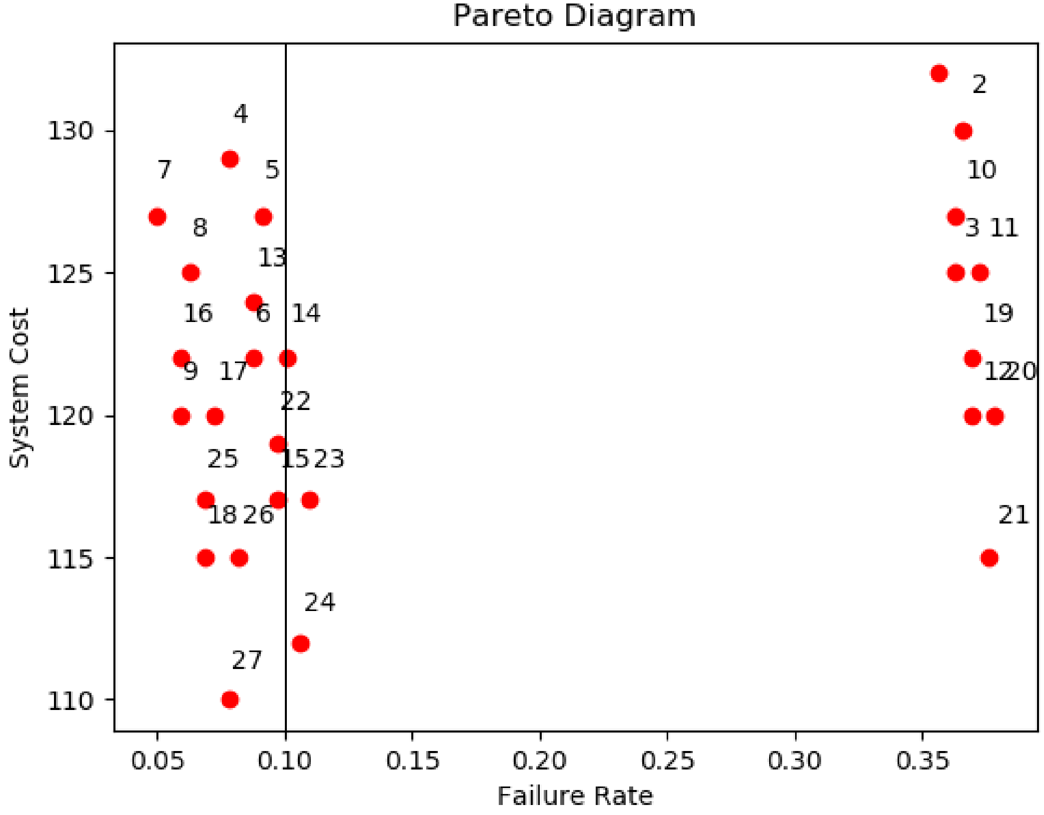


Experimentation (3/4)

Pareto Front



Component Variability

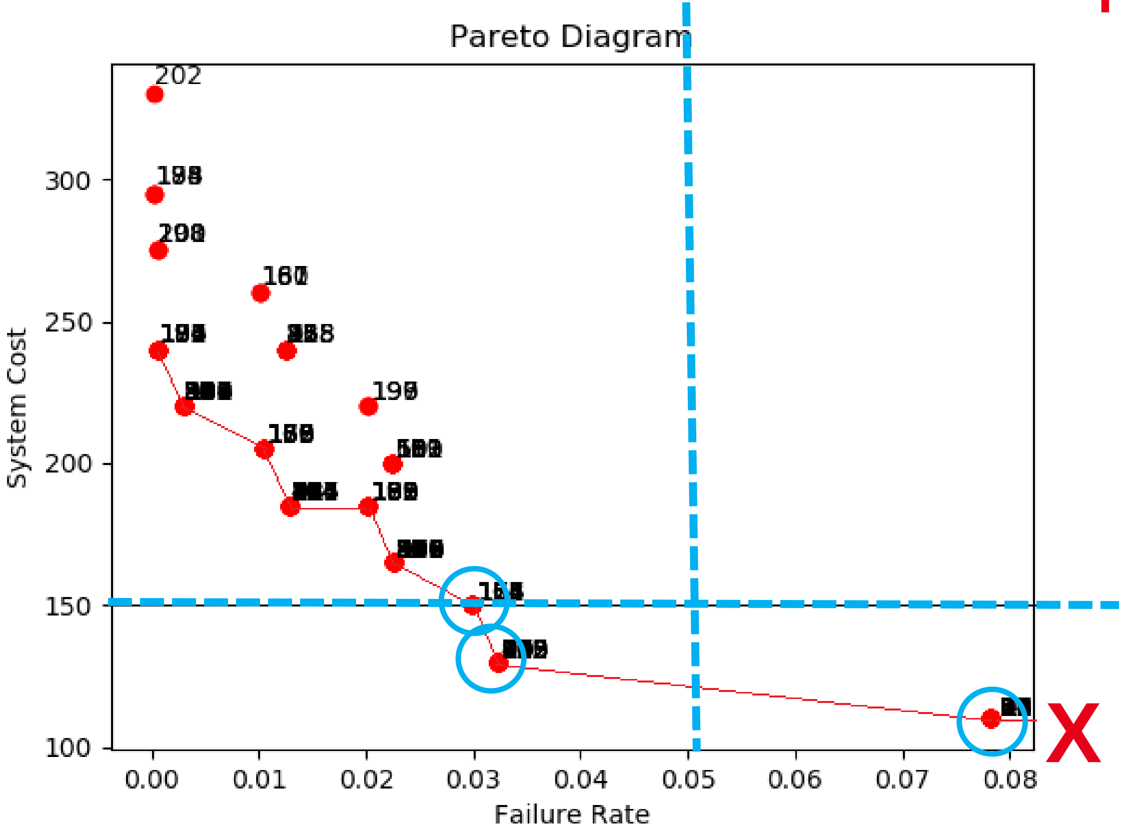


Instance Variability

High-level requirement trade-offs are represented via Pareto front

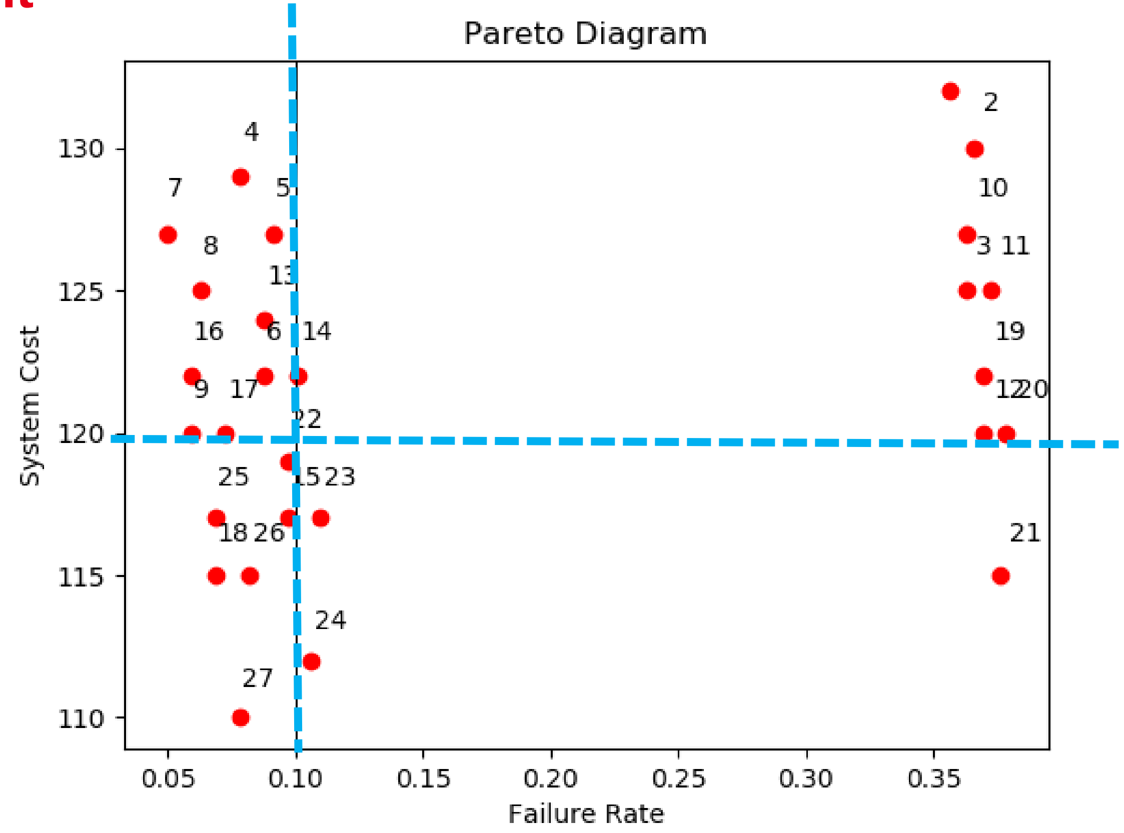
Experimentation (4/4)

Pareto Front



Component Variability

Cost ≤ **150** && Failure rate ≤ **0.05**



Instance Variability

Cost ≤ **120** && Failure rate ≤ **0.10**

Outline

- Introduction
- Research Context & Motivation
- Contribution: Model-based Co-optimization of Objectives
- Experimentation
- **Conclusion & Perspectives**

Conclusion & Perspectives

- We proposed a method and a tool to perform co-optimization of safety & security objective using MDE
  - Method involves SysML based modelling, model transformation and use of constraint solvers to provide (quasi) optimal solutions

- We integrated safety + security features via the common objective function (HW cost), which is optimized w.r.t failure rate to evaluate the impact and interplay of both concerns
  - This work will support engineers to analyze & visualize different concerns (or objectives) along with the possible set of solutions, all in the same interface

- As perspectives:
  - To extend this framework with more objective functions (both common and specific to safety & security)
  - To extend and test the scalability of the approach with larger and complex case studies

Questions?

Kunal Suri, PhD
kunal.suri@cea.fr

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

Établissement public à caractère industriel et commercial | RCS Paris B 775 685 019