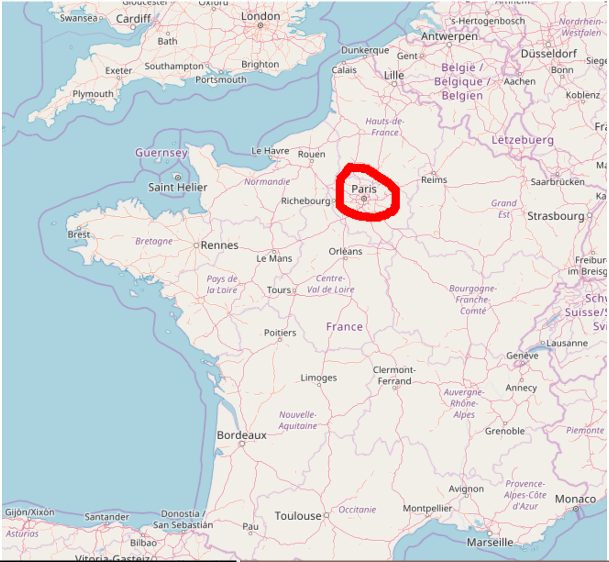
Model-Driven Engineering for Safety, Security and Performance: SysML-Sec

Ludovic APVRILLE
ludovic.apvrille@telecom-paristech.fr

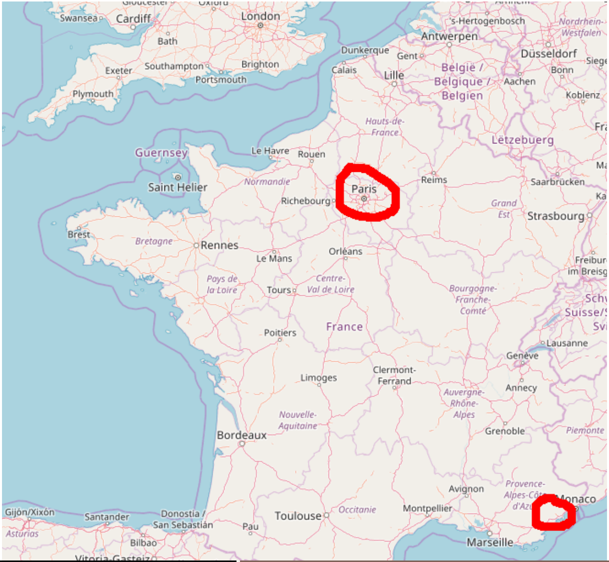
Invited talk - ISAE Supaero

Telecom ParisTech

Telecom ParisTech



Telecom ParisTech



Research Work: Overview

Modeling and Verification for Safe and Secure Embedded Systems
Security Analysis of Embedded Systems

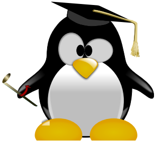
Application domains

IoT, Transports (automotive, drones), telecommunication systems (5G, Software-Defined Radio)

Contracts&projects

Local (Labex, FUI, . . .), national (Nokia lab, . . .), international (H2020 "AQUAS", . . .)

Research Work: Ph.D. Positions



- Definition of safe and secure functional strategies for autonomous vehicles using critical scenarios
- Model-based Joint analysis of Safety and Security

Outline

Examples of Threats

Transport systems

- Use of exploits in Flight Management System (FMS) to control ADS-B/ACARS [Teso 2013]
- Remote control of a car through Wifi [Miller 2015] [Tecent 2017]

Medical appliances

- Infusion pump vulnerability, April 2015. http://www.scip.ch/en/?vuldb.75158

(C) Wired - ABC News

(C) Hospira

Examples of Threats (Cont.)

Internet of Things

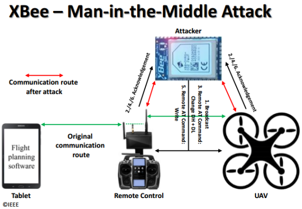
- Proof of concept of attack on IZON camera [Stanislav 2013]

- Vulnerability on fitbit [Apvrille 2015]

A. Apvrille, Hack.lu'2015

(C) beforeitnews

- Hacking a professional drone [Rodday 2016]

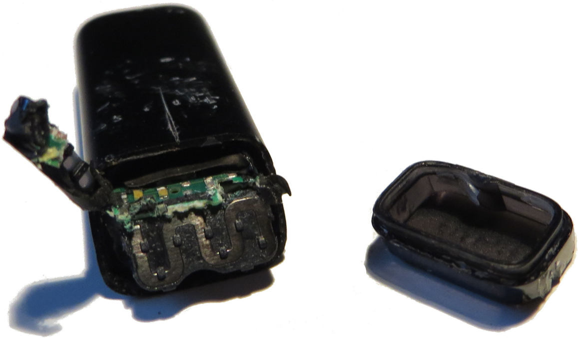


N. Rodday, BlackHat Asia'2016

Finding Vulnerabilities on IoTs

What's inside? Let's look together!

Inside a Fitbit

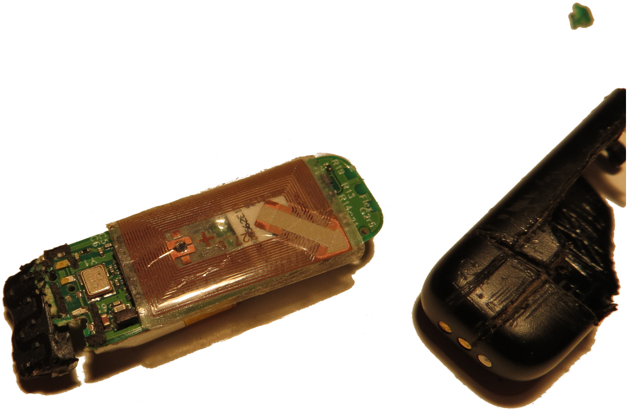


Don't try this at home!

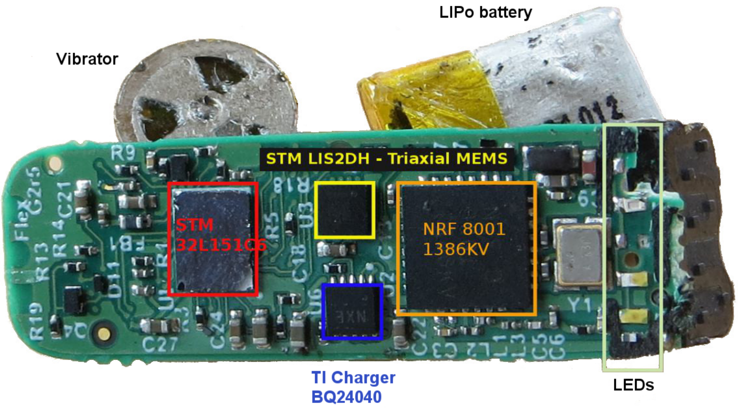
Inside a Fitbit (Cont.)

Again: don't try this at home!

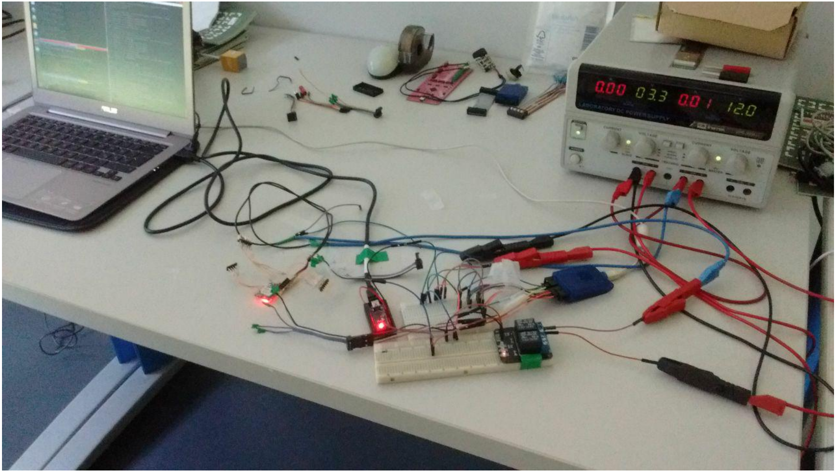
Inside a Fitbit (Cont.)



Fitbit: Hardware Components



Firmware Dumping



Then, How to Identify Vulnerabilities?

Investigations

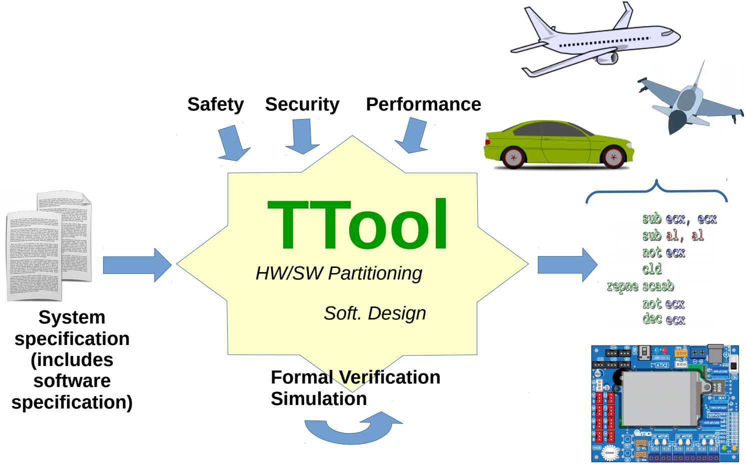
- Testing ports (JTAG interface, UART, ...)
- Firmware analysis
- Memory dump
- Side-channel analysis (e.g. power consumption, electromagnetic waves)
- Fault injection
- ...

Secure your systems!

Develop your system with security in mind from the very beginning

Our solution: SysML-Sec, supported by TTool

Goal: Designing Safe and Secure Embedded Systems



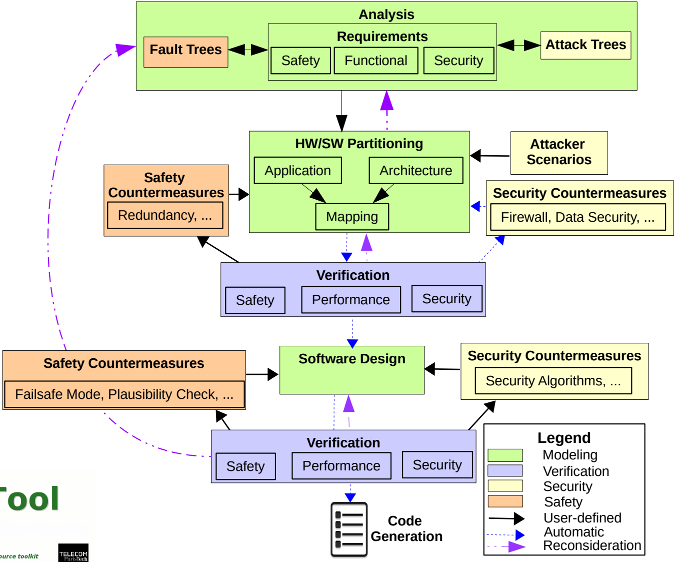
TTool: Key Features

- Model-Driven Engineering tool
- Free and Open-Source
  - Plug-in can be used to insert private/commercial features
- Easy to use
- **Focus on safety, security and performance**
- **Formal verification at the push of a button**

SysML-Sec

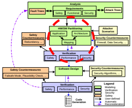
Common issues (addressed by SysML-Sec):

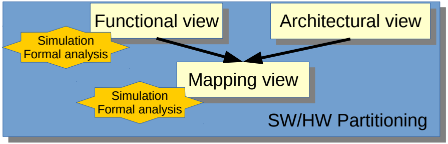
- Adverse effects of security over safety/real-time/performance properties
  - Commonly: only the design of security mechanisms
- Hardware/Software partitioning
  - Commonly: no support for this in tools/approaches in MDE and security approaches



Fully supported by TTool

Partitioning





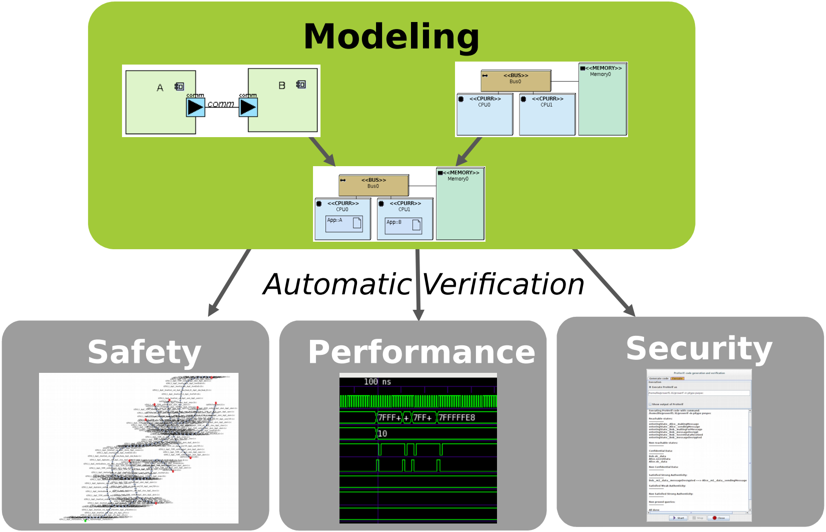
Before mapping

- Security mechanisms can be captured but not verified

After mapping

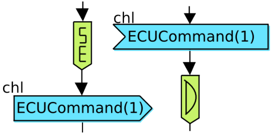
- Verify security (confidentiality, authenticity) according to attacker capabilities
  - Whether different HW elements are or not on the same die
  - Where are stored the cryptographic materials (keys)
  - Where are performed encrypt/decrypt operations
- Impact of security mechanisms on performance and safety
  - e.g. increased latency when inserting security mechanisms

Partitioning Verification



Safety and Security Mechanisms

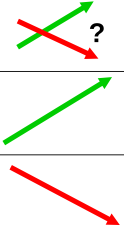
Data Encryption/ Authentication



Safety

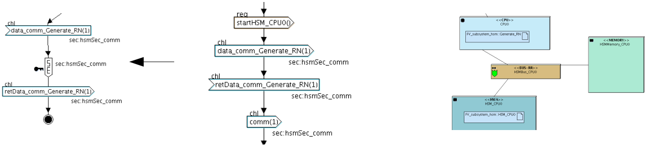
Security

Performance



Safety and Security Mechanisms (Cont.)

Data Security with Hardware Security Module

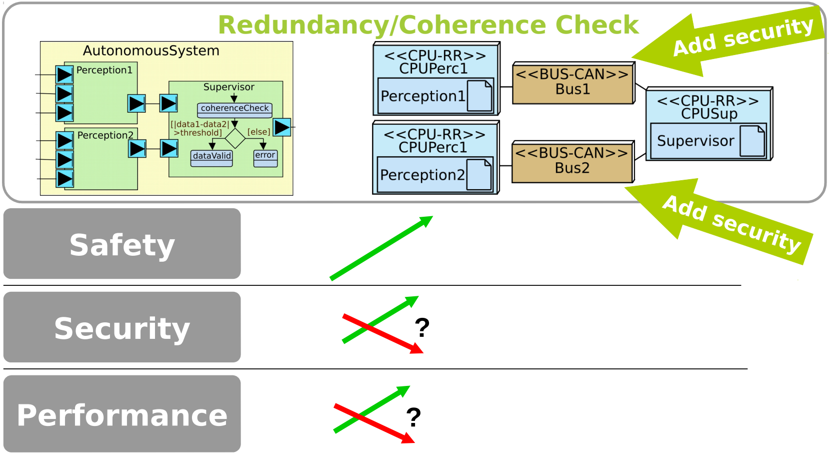


Safety ?

Security

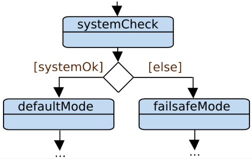
Performance

Safety and Security Mechanisms (Cont.)



Safety and Security Mechanisms

Failsafe mode

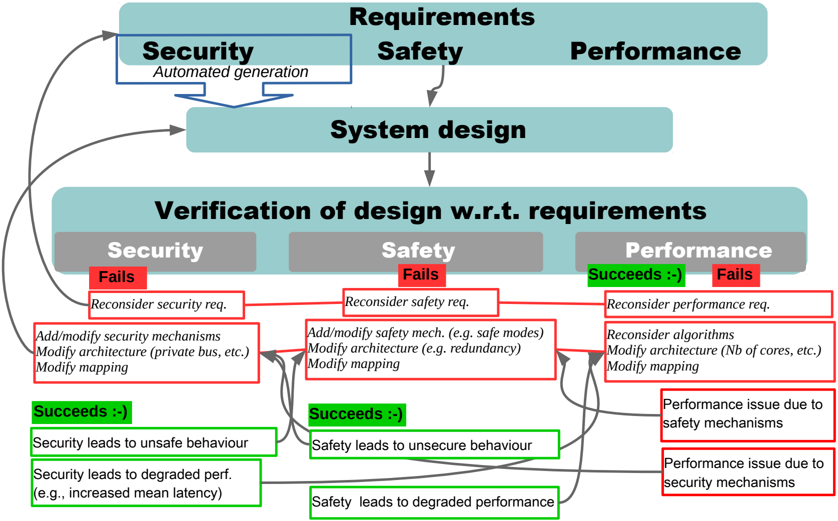


Safety

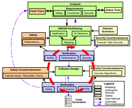
Security ?

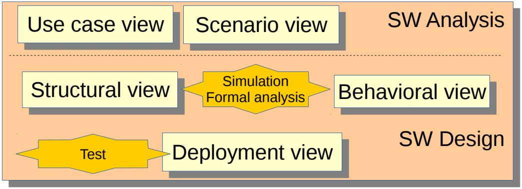
Performance ?

Safety/Security/Performance



SysML-Sec: SW Design





- Precise model of security mechanisms (security protocols)
- Proof of security properties : confidentiality, authenticity
- Channels between software blocks can be defined as private or public
  - This should be defined according to the hardware support defined during the partitioning phase

Case Studies

Cyber security of connected vehicles

- Safety/Security/Performance
- EVITA FP7 Partners: Continental, BMW, Bosch, . . .
- VEDECOM

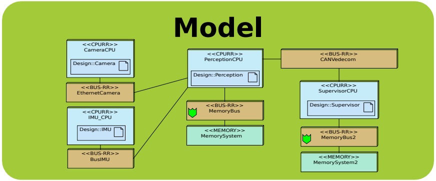
H2020 AQUAS

- Automated train sub-systems (ClearSy): Safety/Security/Performance
- Industrial Drives (Siemens): Safety/Security/Performance

Nokia

- Digital architectures for 5G networks (Safety/Performance)

Case Study: VEDECOM Autonomous Vehicle



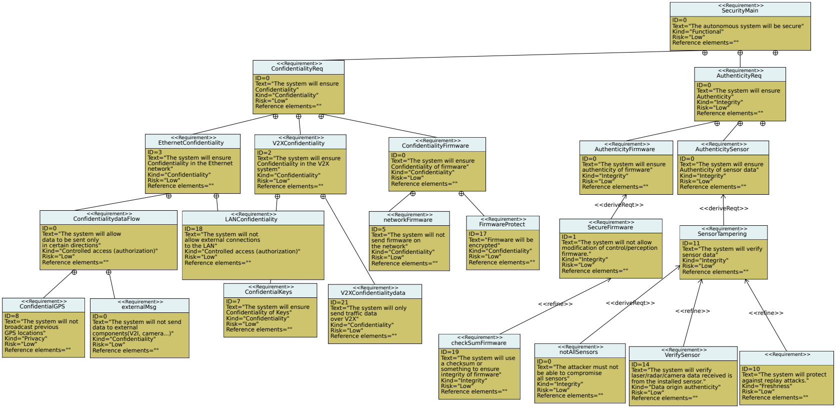
Verification

Tests

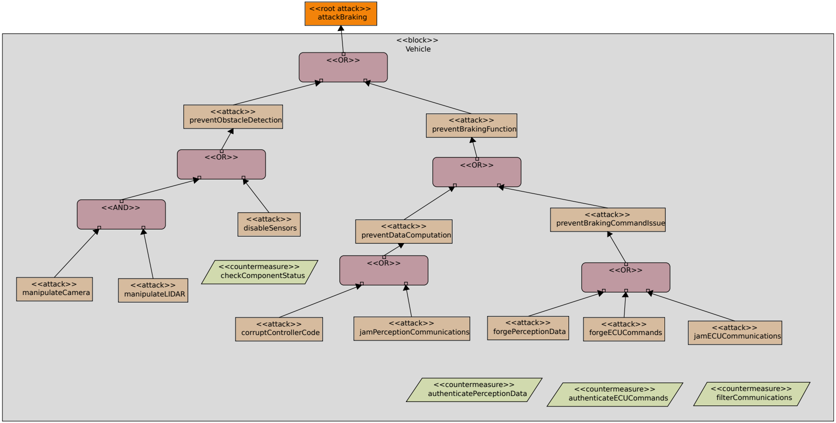
Constraints

- Standard: ISO26262
  - SOTIF: Safety Of The Intended Function
- Security: impact of potential attacks on safety

Requirements

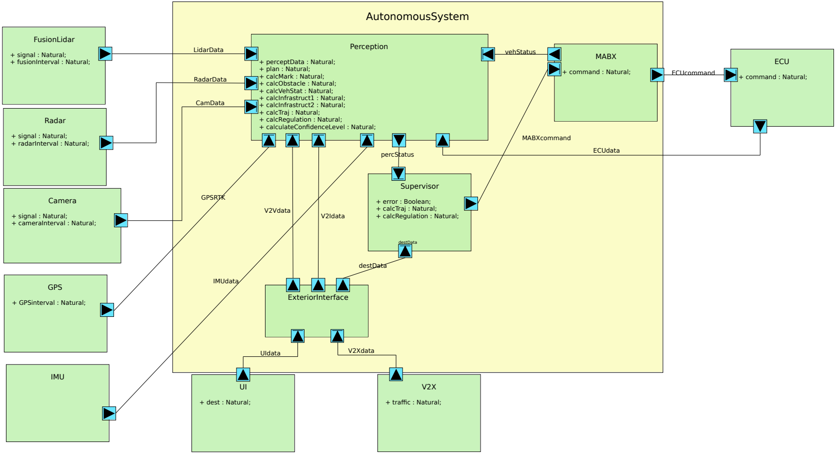


Attacks



<<root attack>> attackBraking
<<block>> Vehicle
<<OR>>
<<attack>> preventObstacleDetection
<<attack>> preventBrakingFunction
<<OR>>
<<OR>>
<<AND>>
<<attack>> disableSensors
<<attack>> preventDataComputation
<<attack>> preventBrakingCommandIssue
<<countermeasure>> checkComponentStatus
<<OR>>
<<OR>>
<<attack>> manipulateCamera
<<attack>> manipulateLIDAR
<<attack>> corruptControllerCode
<<attack>> jamPerceptionCommunications
<<attack>> forgePerceptionData
<<attack>> forgeECUCommands
<<attack>> jamECUCommunications
<<countermeasure>> authenticatePerceptionData
<<countermeasure>> authenticateECUCommands
<<countermeasure>> filterCommunications

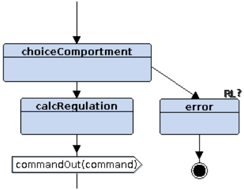
Functional View



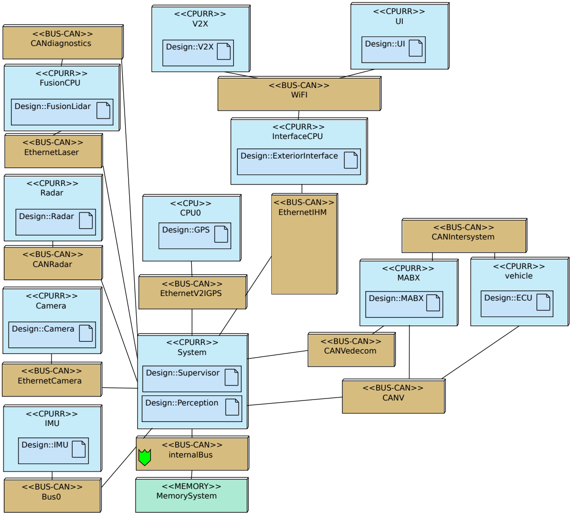
Safety Verification (Before Mapping)

Reachability/Liveness

Queries

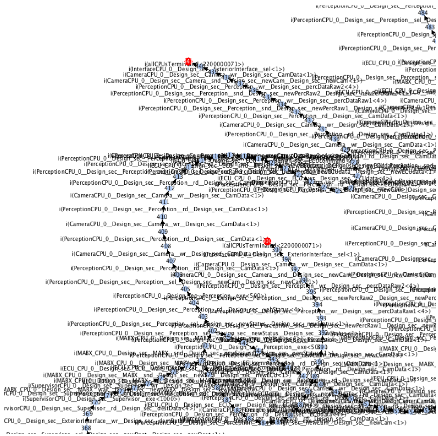


Architecture and Mapping Views

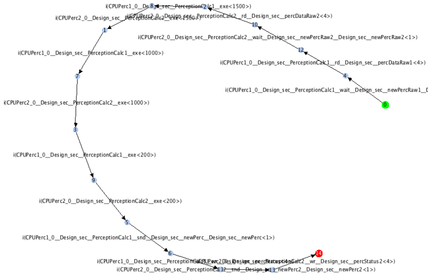


Safety Verification (After Mapping)

Reachability Graph

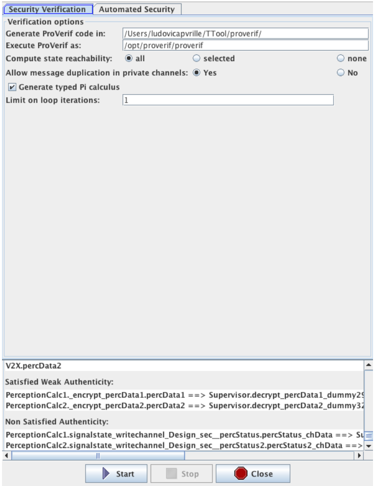


Minimized RG

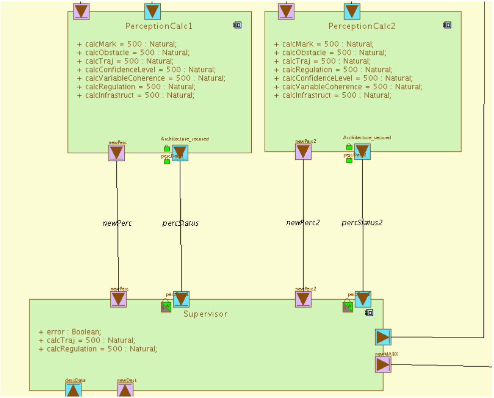


Security Verification

Dialog window



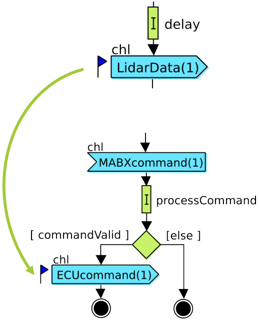
Backtracing

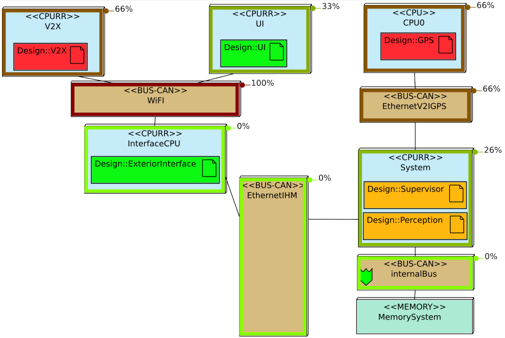


TELECOM ParisTech

Performance Verification

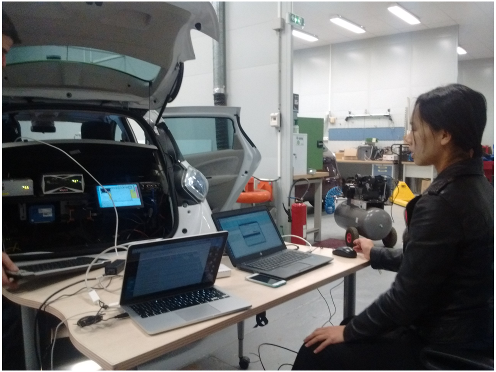
Latency	Bus/CPU Load





SW Design, Code generation, Test

- First SW model from mapping models
- SW model refinement
- SW model verification (safety, security)
- Code generation
  - (Virtual) Prototyping, test



Demo: SmartCard

- Main functions of the system
- Safety of the system (before mapping, after mapping)
- Performance
- Model enhanced with Security
- Impact on performance

Conclusion and Future Work

Achievements: SysML-Sec

- Methodology for designing safe and secure embedded systems
- Fully supported by TTool
- Applied to different domains, e.g., automotive systems, IoTs, malware

Future work

- Security risk assistance and backtracing
- Assistance to handle conflicts between security/safety/performance
  - Design space exploration

To Go Further ...

Web sites

- https://sysml-sec.telecom-paristech.fr
- https://ttool.telecom-paristech.fr

References

- Ludovic Apvrille, Yves Roudier, "SysML-Sec: A SysML Environment for the Design and Development of Secure Embedded Systems", Proceedings of the INCOSE/APCOSEC 2013 Conference on system engineering, Yokohama, Japan, September 8-11, 2013.
- Ludovic Apvrille, Yves Roudier, "Designing Safe and Secure Embedded and Cyber-Physical Systems with SysML-Sec", Chapter in Model-Driven Engineering and Software Development, p293–308, Springer International Publishing, 2015