An overview of the Embedded Systems unit at Fondazione Bruno Kessler

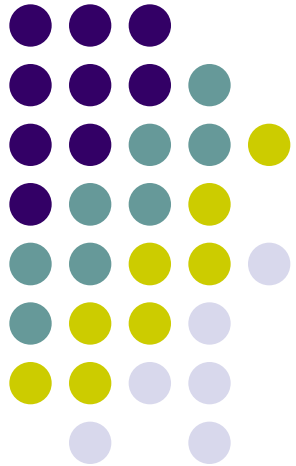
Alessandro Cimatti

Fondazione Bruno Kessler

Center for Information Technology – IRST

Head of Embedded Systems Unit

cimatti@fbk.eu



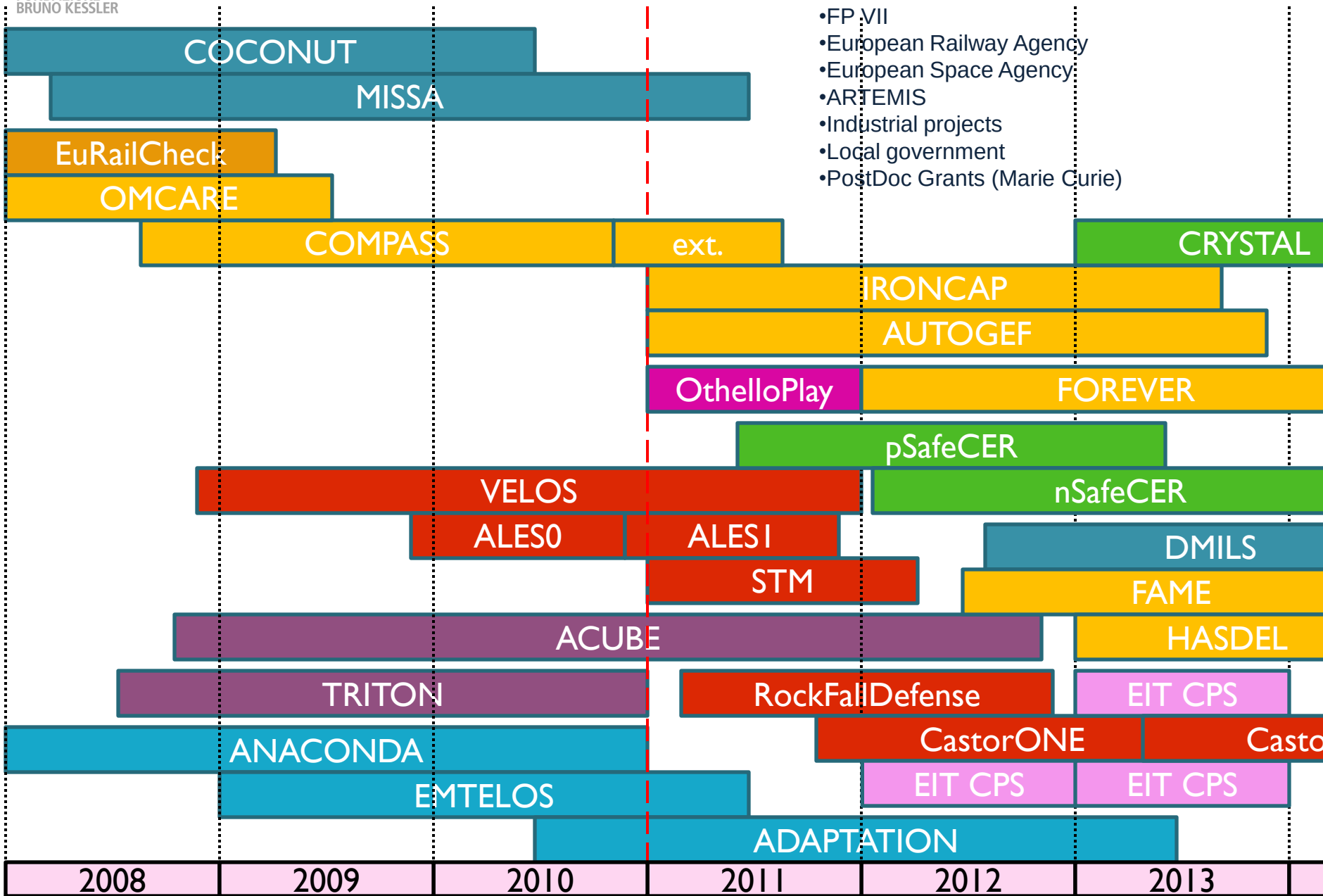
FONDAZIONE BRUNO KESSLER

Overview

- The Embedded Systems Unit
  - 28 people
  - 7 research staff, 7 postdocs, 8 programmers, 6 ph.d. students
  - Open call for more ph.d. students and postdocs!
  - H-indices: 39, 27, 25, 20, 11, 9

- Strategy: tight integration of
  - Basic research
  - Tool development
  - Technology transfer

Projects/funding history (since 2008)



FONDAZIONE BRUNO KESSLER

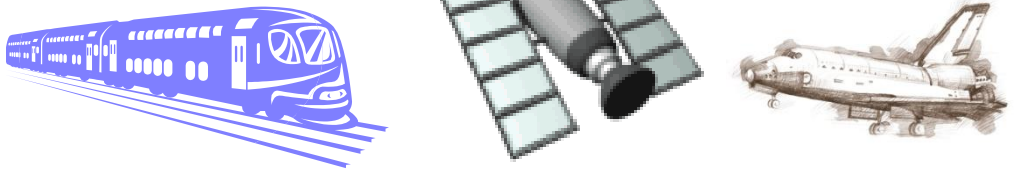
Our activities

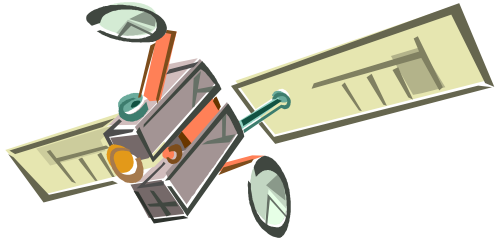
- Support Design automation with formal methods
  - Find more bugs, earlier in design flow, certify correctness
  - Areas
    - Functional verification (traditional)
    - Dependability (FTA, FMEA) assessment
    - Requirements analysis
- Model based autonomous reasoning
  - Planning and scheduling
  - Execution monitoring
  - Fault detection, identification and recovery (FDIR)
- Distributed Wireless Sensor Networks
  - middleware for advanced programming
- Complex Embedded Intelligence Systems
  - Monitoring of social environments

Better Embedded Systems

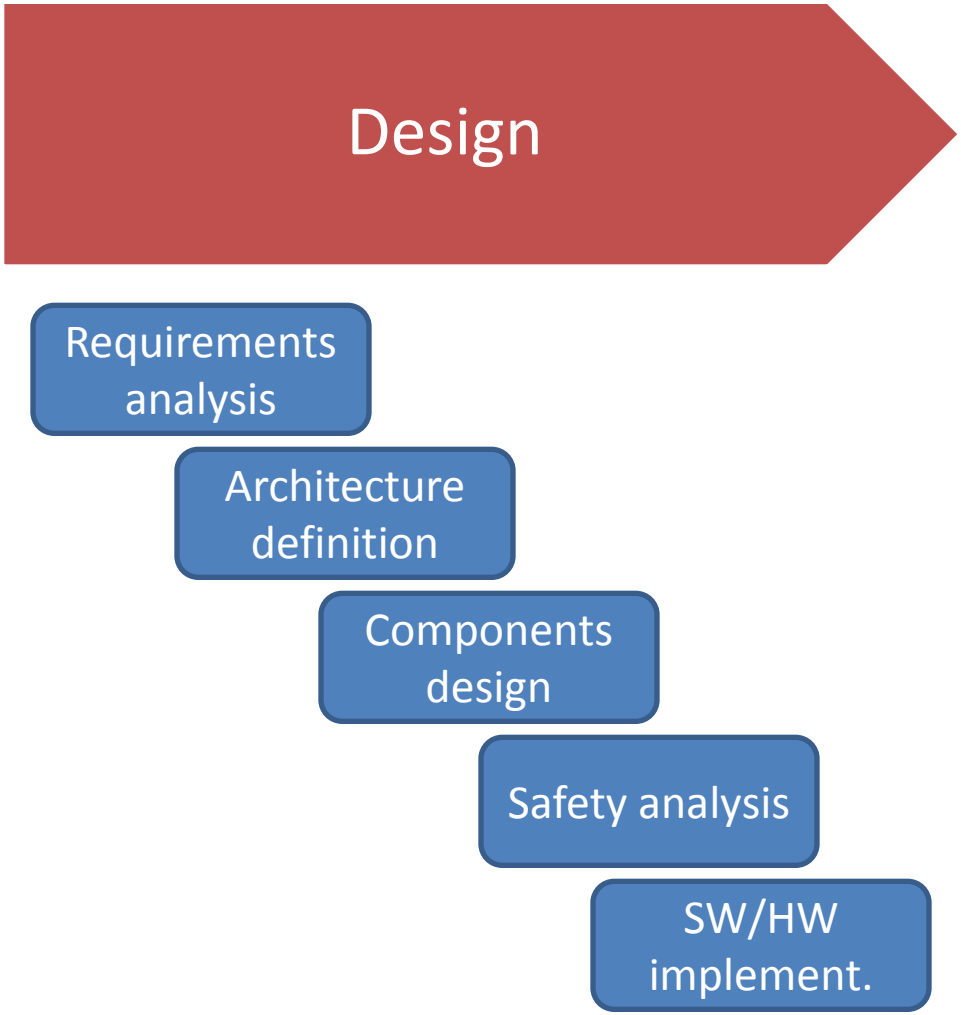
- Embedded systems
  - Over 98% of all computing chips are *embedded*
    - Hidden in all sorts of things that do not even look like computers
  - Embedded processors top 10 billion units in 2008
  - Average annual growth rates of 6.4%.in 2008-2013

- Some application domains
  - automotive:
    - ABS, drive-by-wire, airbags, traction control, fuel injection
  - railways:
    - control of the trackside (e.g. switches, semaphores)
    - onboard breaking control
  - avionics:
    - fly-by-wire
  - hardware
    - microprocessors, ASIC, Systems on Chip
  - space:
    - satellites, rovers
  - industrial climatisation
    - production temperature control
  - environment monitoring
    - prevention/detection of avalanches
  - domotics
    - sensor networks, intelligent sensing
  - biomedical devices
    - radiation control





Life Cycle of Complex Systems

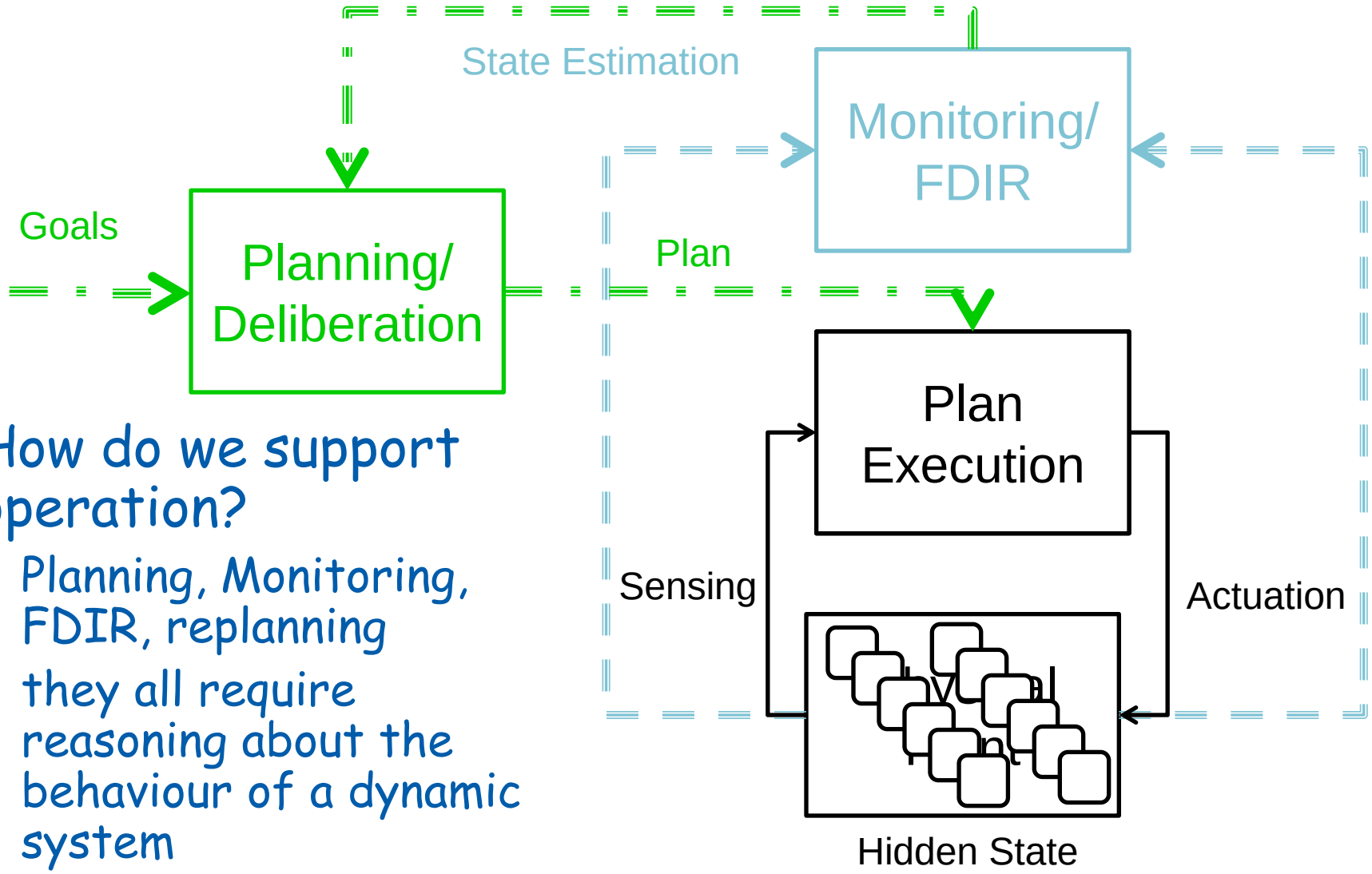


- How do we support the design?
- Requirements validation:
  - Are the requirements flawed?
- Functional correctness
  - Does the system satisfy the requirements?
- Safety assessment
  - Is the system able to deal with faults?

From design to operation…

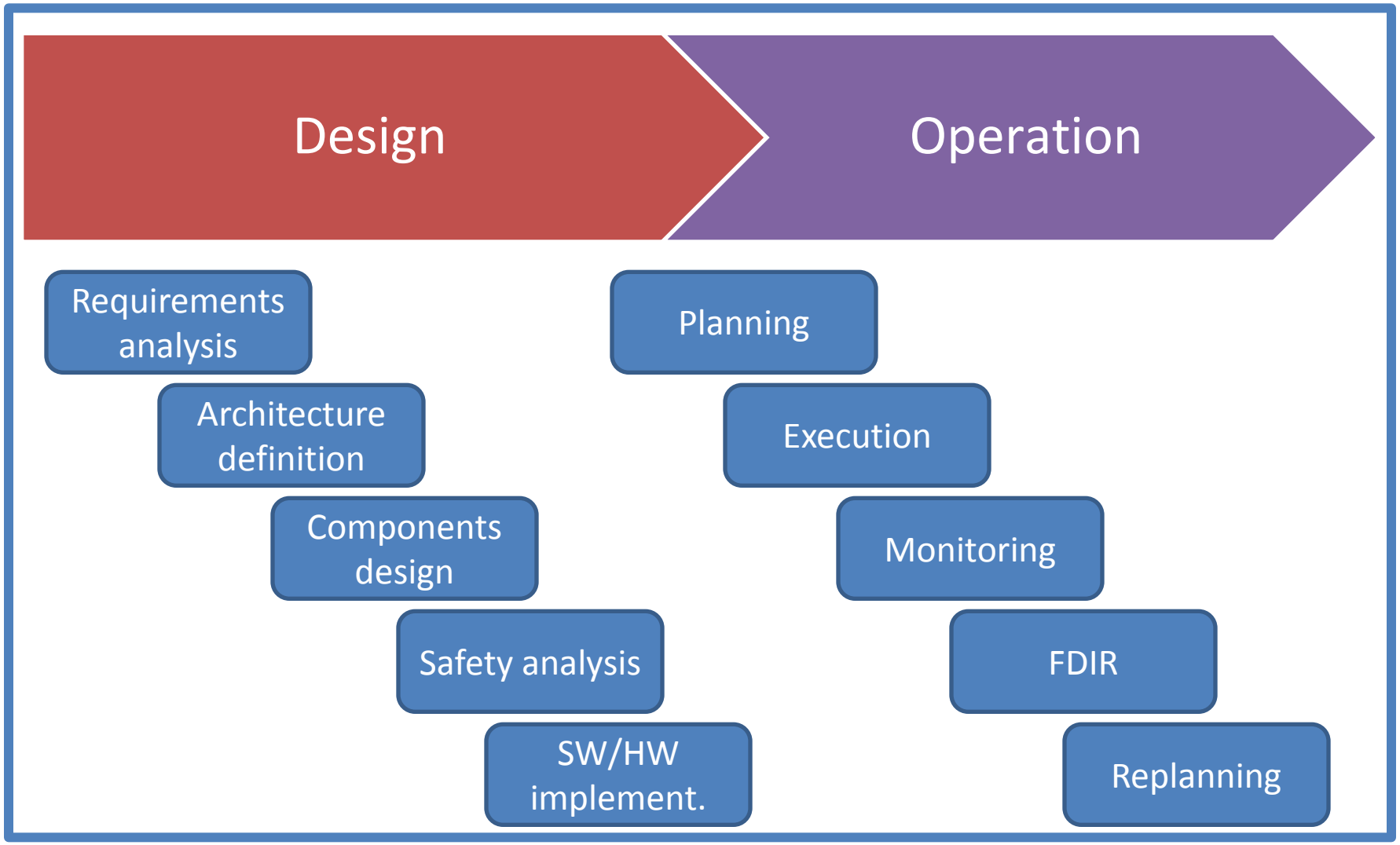
- Planning
  - plan how to achieve desired "firing" sequence
  - retrieve pipes from holds, pre-weld, send to firing line, final weld
- Execution Monitoring
  - welding may fail, activities can take more time than expected
  - plant may fail
- Fault Detection, Fault Identification/Isolation
  - is there a problem? where is it?
- Fault Recovery
  - put off-line problematic equipment
- Replanning
  - identify alternative course of actions, e.g. reroute pipes

Complex systems operation



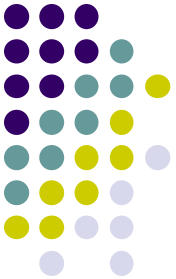
- How do we support operation?
  - Planning, Monitoring, FDIR, replanning
  - they all require reasoning about the behaviour of a dynamic system

Life Cycle of Complex Systems



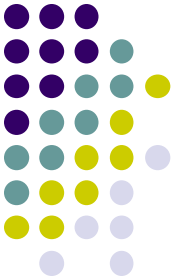
Projects in Requirements Analysis

The ETCS project



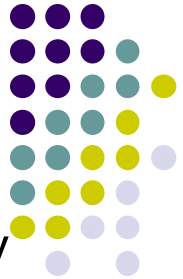
- Invitation To Tender by European Railway Agency
- Proposal with RINA and Dr. Graband
- Two cooperating UdRs
  - Embedded Systems
  - Software Engineering
- Selected out of a dozen proposals
- Tight timing constraints

Focus: requirements, not model



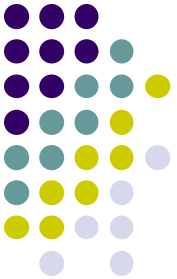
- In traditional formal verification
  - the *design* is under analysis
  - the requirements are taken as "golden"
  - verification means checking compliance
- Here the goal is to
  - enhance quality of *requirements*
- A much harder task!
  - from informal to formal

Why is it so hard?



- Requirements analysis is a pervasive problem in nowadays industry
  - In hardware design, standards for languages to represent properties and design intent are emerging (e.g. PLS, SVA)
- Problem 1: Natural language
  - ambiguous
  - hard to process automatically with NLP
  - requires background information
- Problem 2: when are my requirements good?
  - are they too strict? Are some required behaviours being (wrongly) disallowed?
  - are they too weak? Are some undesirable behaviours being (wrongly) allowed?
- The source of the matter is that what is being modeled is informal
  - the design intent that must be captured by the specification is in the head of the specifier

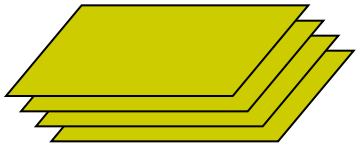
Issues of interest in this project



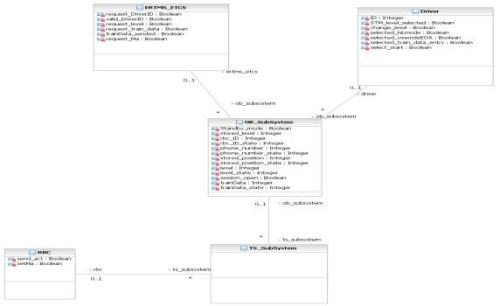
- PB1: Bridging the gap between natural language and formal analysis
- PB2: providing methods for pinpointing flaws in requirements
- And also (as usual) …
  - Integration within requirements engineering flow
  - Usability
    - Avoid intricate formalisms
    - Hide formal methods with semiformal representations
  - Automation of the verification process
    - Model checking

From Informal to Formal

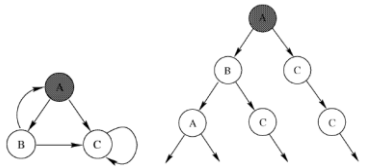
NATURAL LANGUAGE

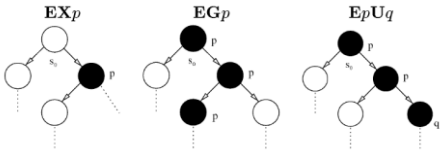


SEMIFORMAL LANGUAGE



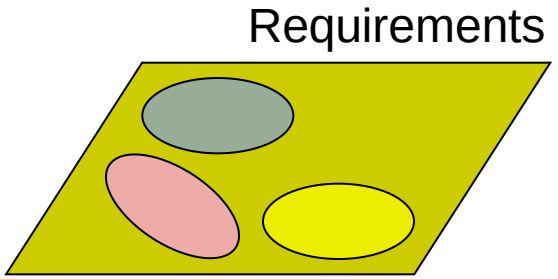
FORMAL LANGUAGE

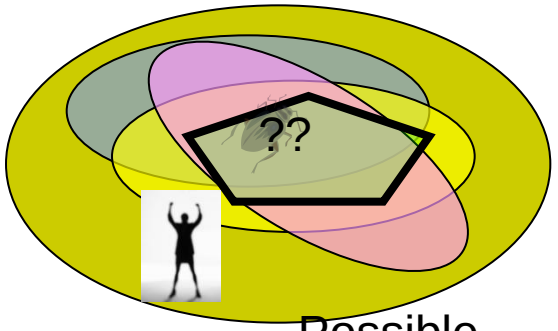




Which flaws in requirements?

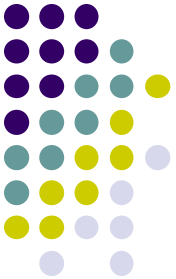
- A set of requirement is a set of constraints over possible evolutions of the entities in the domain
- Possible questions
  - Are my requirements too strict?
  - Are my requirements too weak?
- Possible checks
  - Consistency check (too strict?)
    - is there at least one admissible behaviour?
  - Possibility check (too strict?)
    - is a given desirable behaviour admissible?
  - Assertion check (too weak?)
    - is a given undesirable behaviour excluded?





- Warning: no way to formalize design intent!

ETCS
Results and next phases

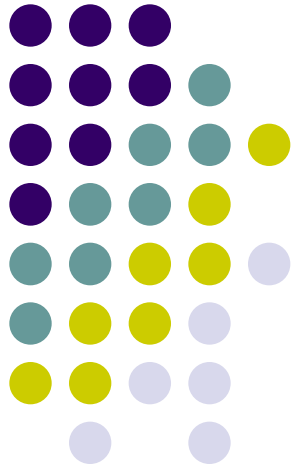


- Three main outputs
  - A requirements analysis methodology
    - integrating informal and formal techniques
    - hiding formal techniques as much as possible
  - A support toolset
    - based on standard commercial tools
    - integrating state-of-the-art verification engine
  - Formalization of substantial fragments of ETCS specifications

- Additional insights
  - 2-days theoretical training (25 people)
  - 3 x 5-days hands-on training classes
  - Distribution/maintenance of ERA tool by FBK

- Ongoing activities
  - Dissemination (under ERA auspices)
  - Investigation of NLP techniques

Projects in Design Verification



FONDAZIONE BRUNO KESSLER

NuSMV: an open model checker

- Started in 1997
  - Joint project with CMU, UniTN, UniGE
- Traditional hardware verification
  - Finite state models
  - Temporal logic model checking
- Symbolic methods
  - BDD-based, SAT-based
- Widely used
  - 20K downloads (> 500 cit.)
  - Backend for design environments
    - Fujitsu, Rockwell-Collins, …
    - Communications of the ACM 53(2): *Software Model Checking Takes Off*

FONDAZIONE BRUNO KESSLER

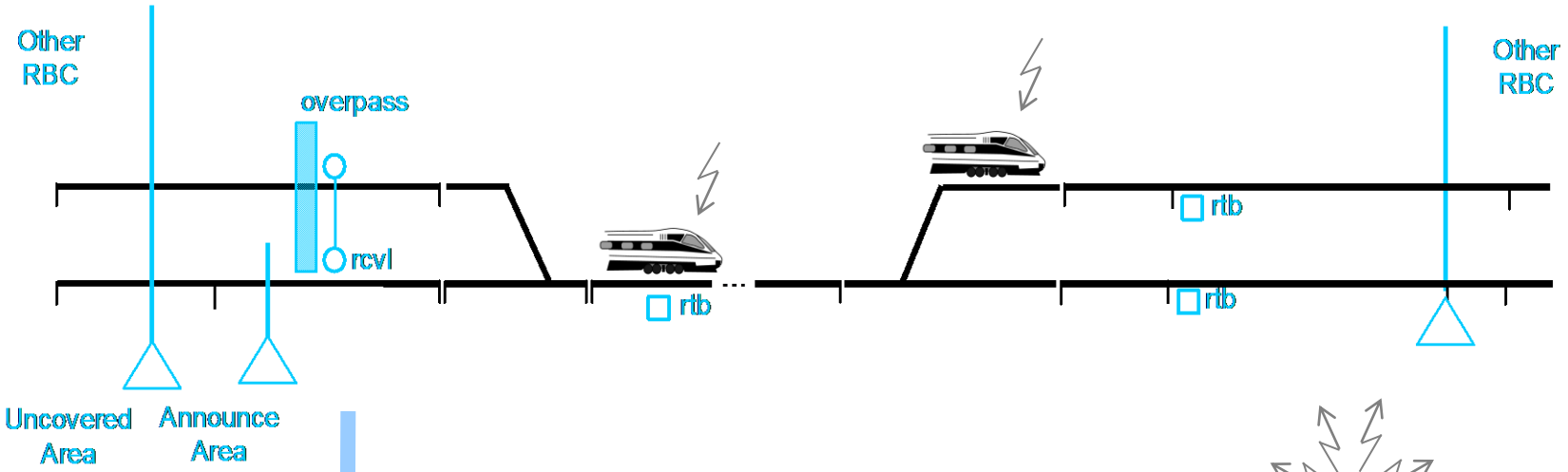
The new generation…

- The nuXmv model checker
  - Discrete case: more powerful verification engines
    - Extended verification algorithms
    - Compositional reasoning
  - Infinite case: real and integer variables
    - Can model Real-time, resources, speed, power, …
    - Discrete+continuous dynamics
    - Tight integration with MathSAT5
  - More functions in a unique integrated framework
    - requirements analysis
    - functional verification
    - dependability assessment

Railways interlocking

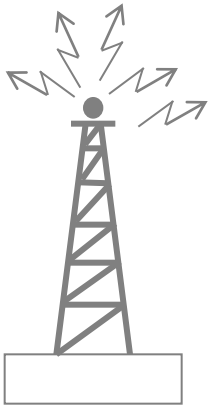
- From relay circuits to software
- Key problems
  - Correctness
  - Time to delivery
  - Certification costs

Train to Track traffic control

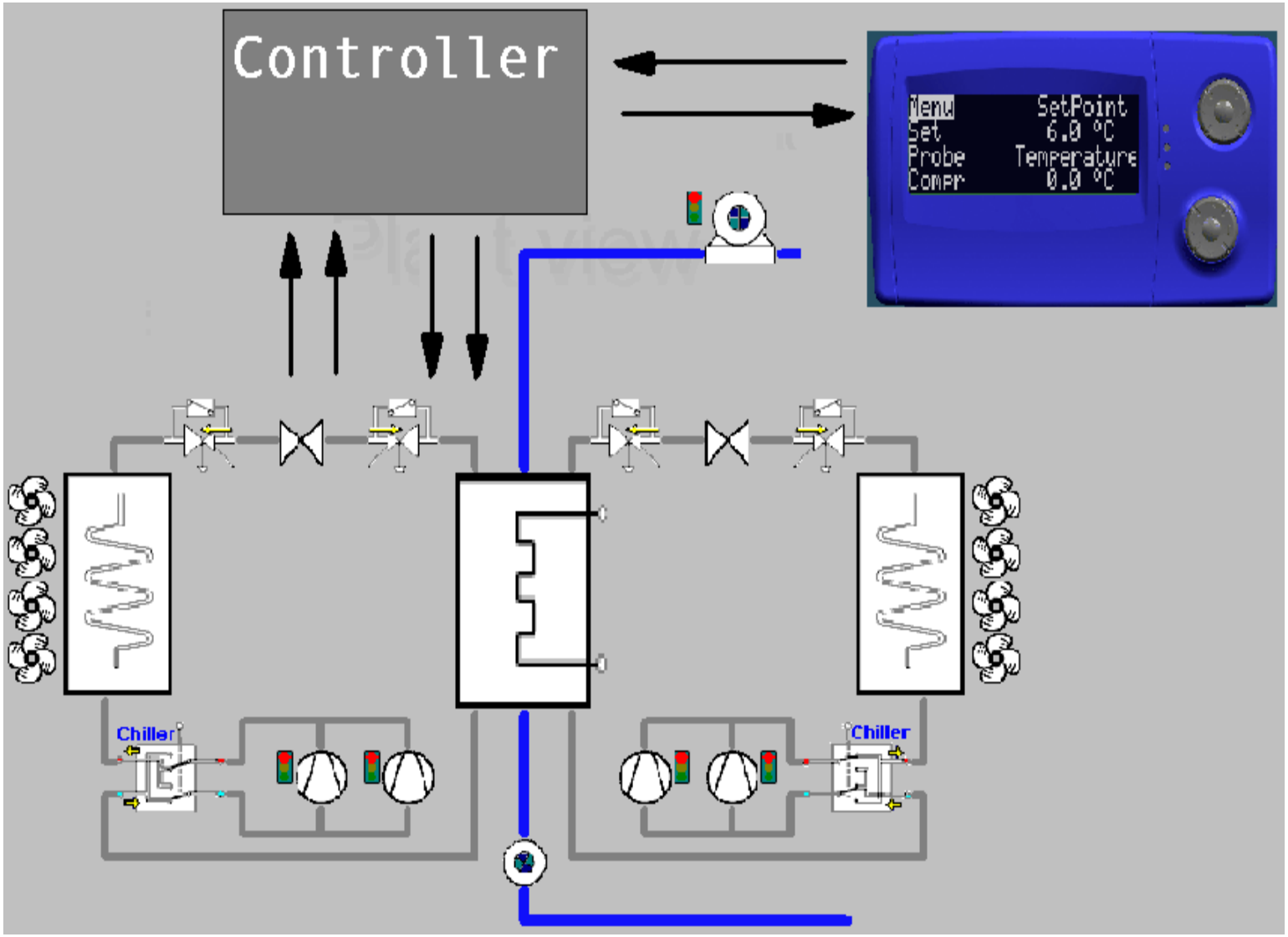


Interlocking

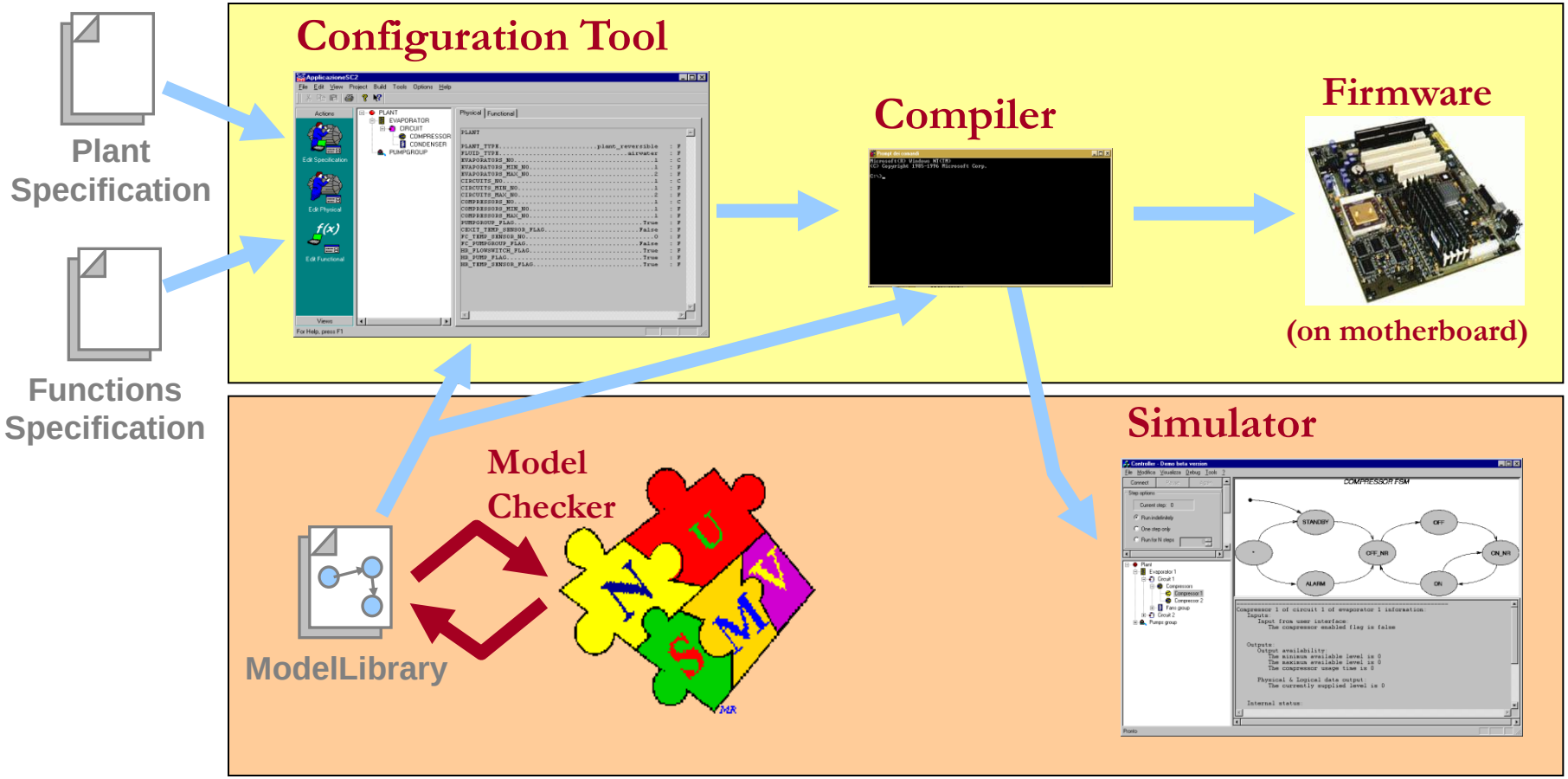
Radio Block Centre



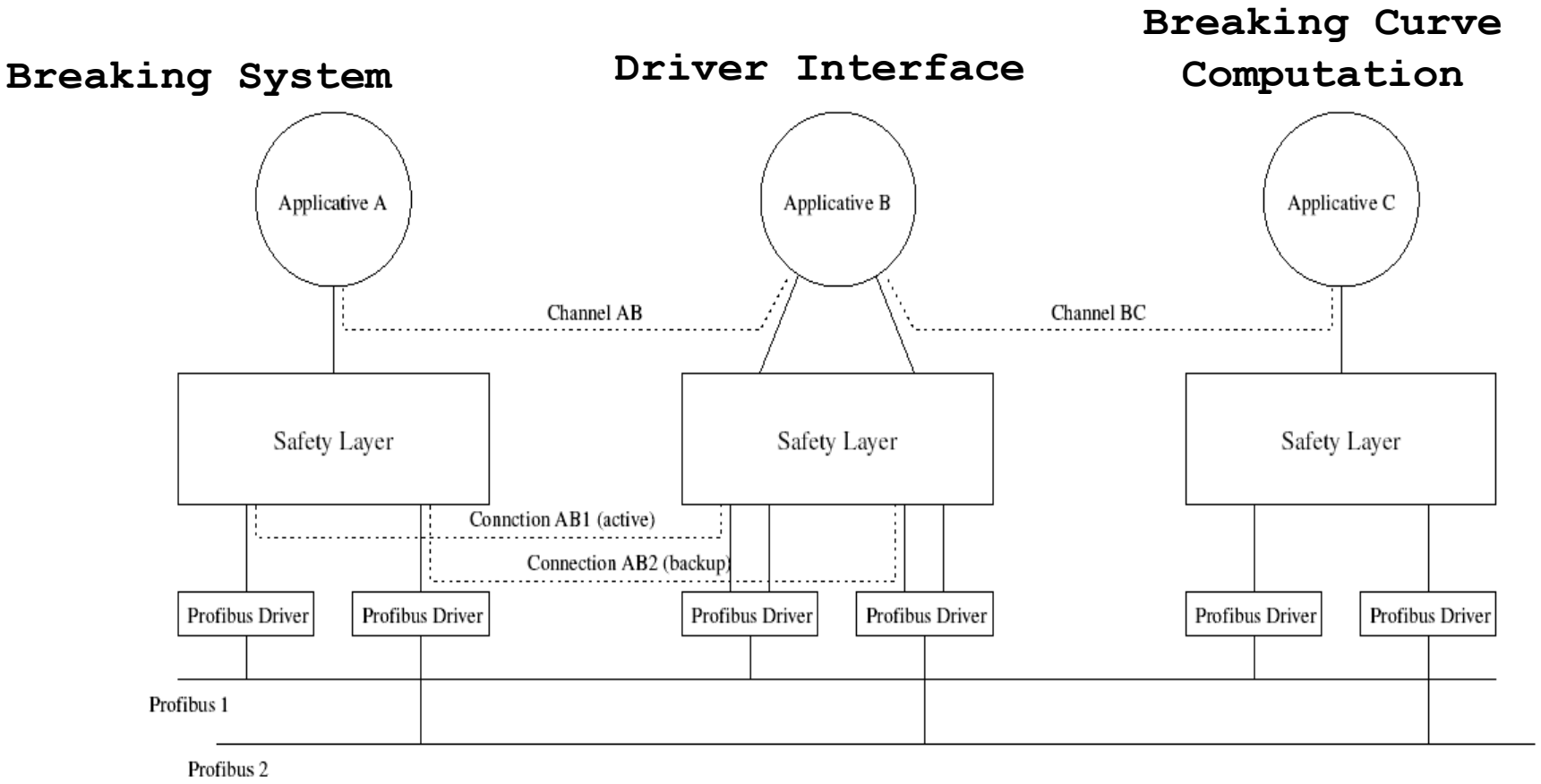
Controls for Air Conditioning



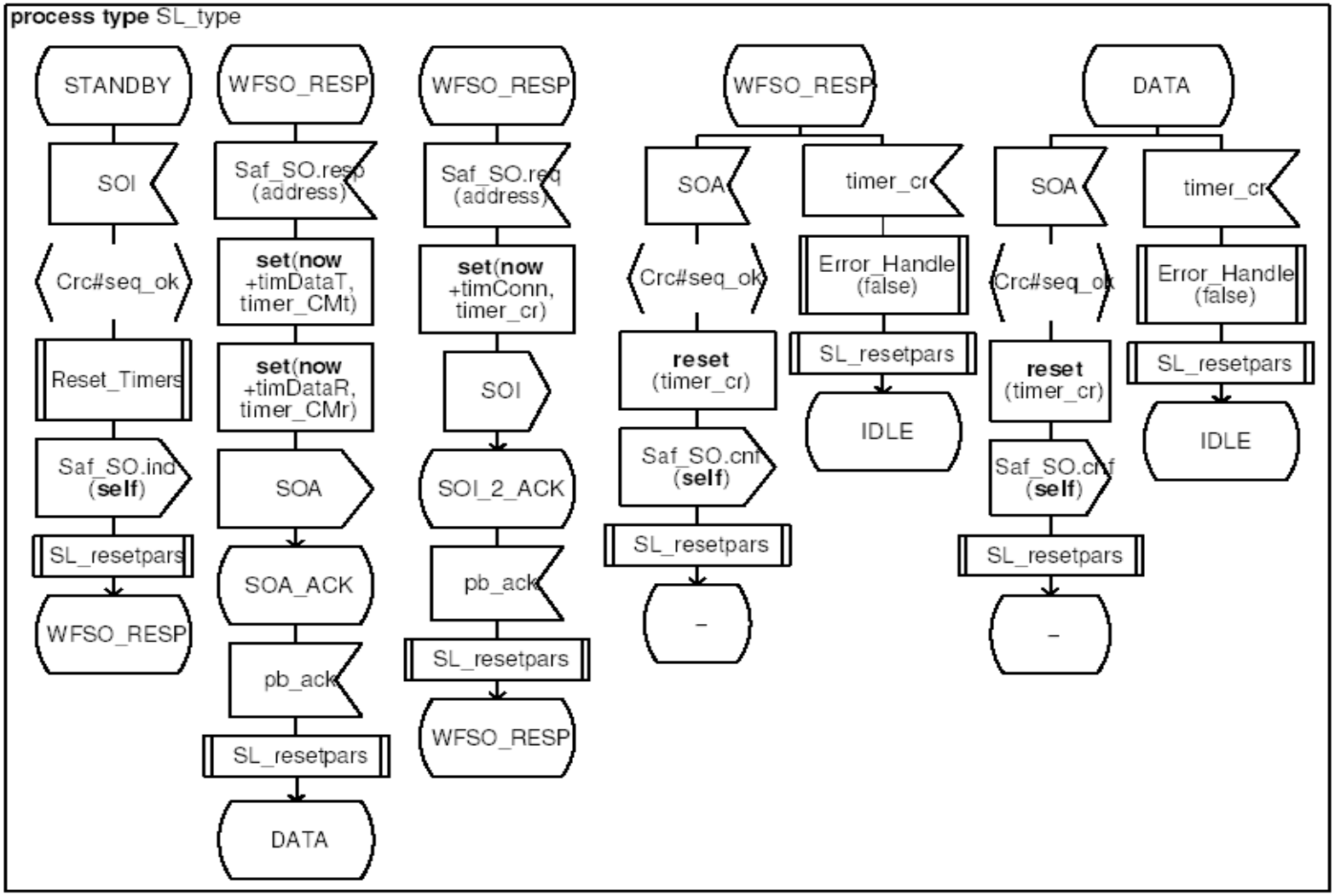
Air Conditioning Control



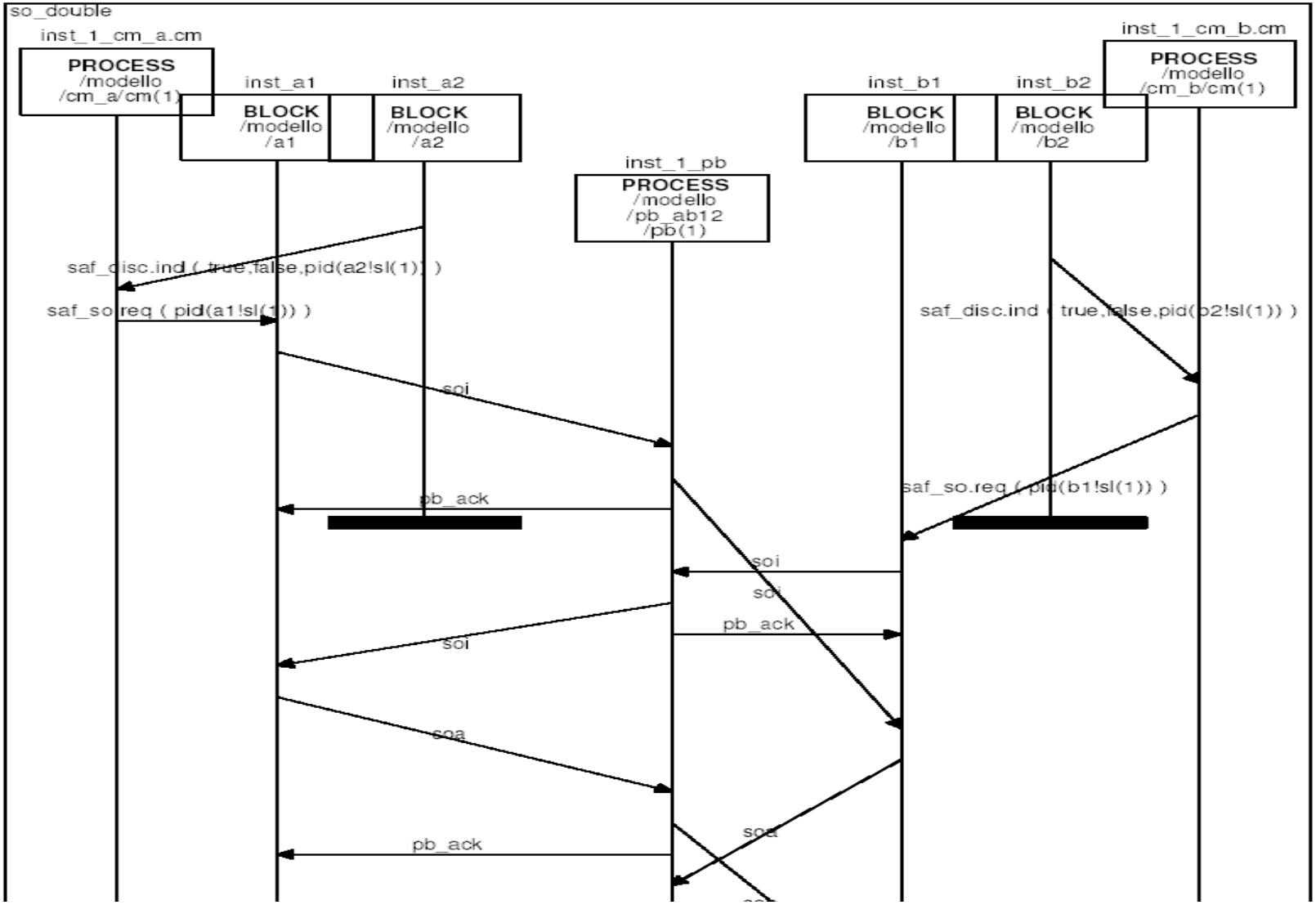
Formal Design of Vital Protocol



Formal Design of Vital Protocol



Formal Design of Vital Protocol

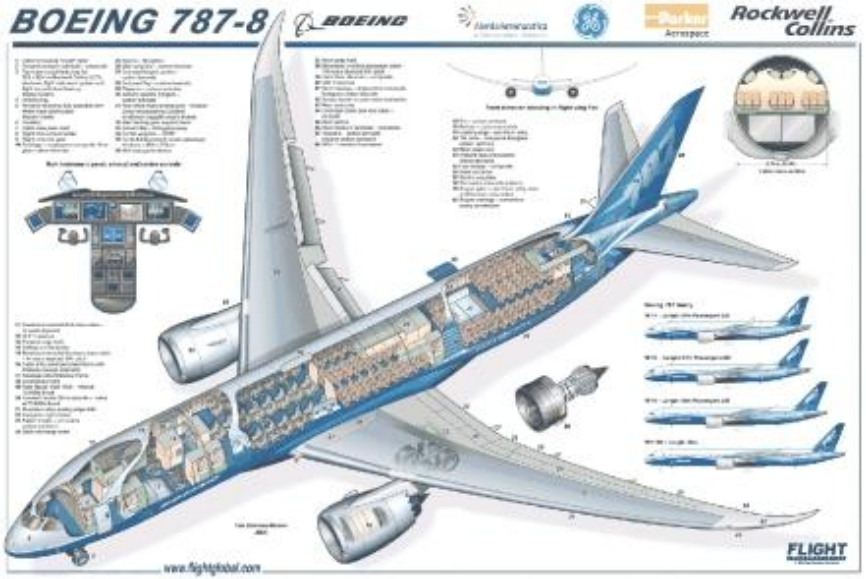


Verification of Avionic Software

- Power transfer system
  - dynamic reconfiguration for on-board power generation
  - Boeing 787, classified content
  - centralized control algorithm: very complex
  - in practice: 6 CU, complex network topology, high degree of parameterization, distributed control
- Design flow
  - Matlab/Simulink/StateFlow design
  - automated code generation
- Key problem: testing is unfeasible!
  - execution of up to 50 test cases on ground simulated (split aircraft)
  - thousands of years necessary for required coverage

- Countermeasure: formal verification
- NuSMV in design flow
  - Matlab to NuSMV translator
  - Custom NuSMV functionalities based on design style

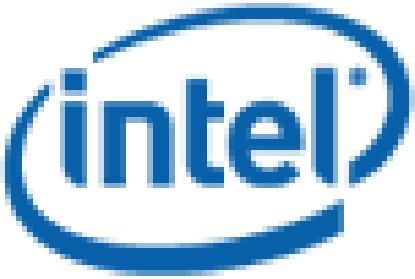
- Target commercialization of new verification flow
  - address 178B certification issues
    - automated test pattern generation
    - support to inspection
  - expand to other application domains
    - automotive, space



FONDAZIONE BRUNO KESSLER

MicroCode Verification

- SMT: next generation constraint solvers
  - boolean reasoning (a la SAT)
  - extended with constraint solving
  - reals, integers, EUF, bit vectors, arrays, …
- The MathSAT SMT solver
  - Backend for NuSMV3, software model checking
- Industrial impact
  - Bit-vector solver to verify microcode
  - Integration in Intel design flow
    - Combinational equivalence checking
    - Sequential equivalence checking
    - Property verification
  - Significant scalability advances
    - Hours vs seconds
  - Best paper award at FMCAD10



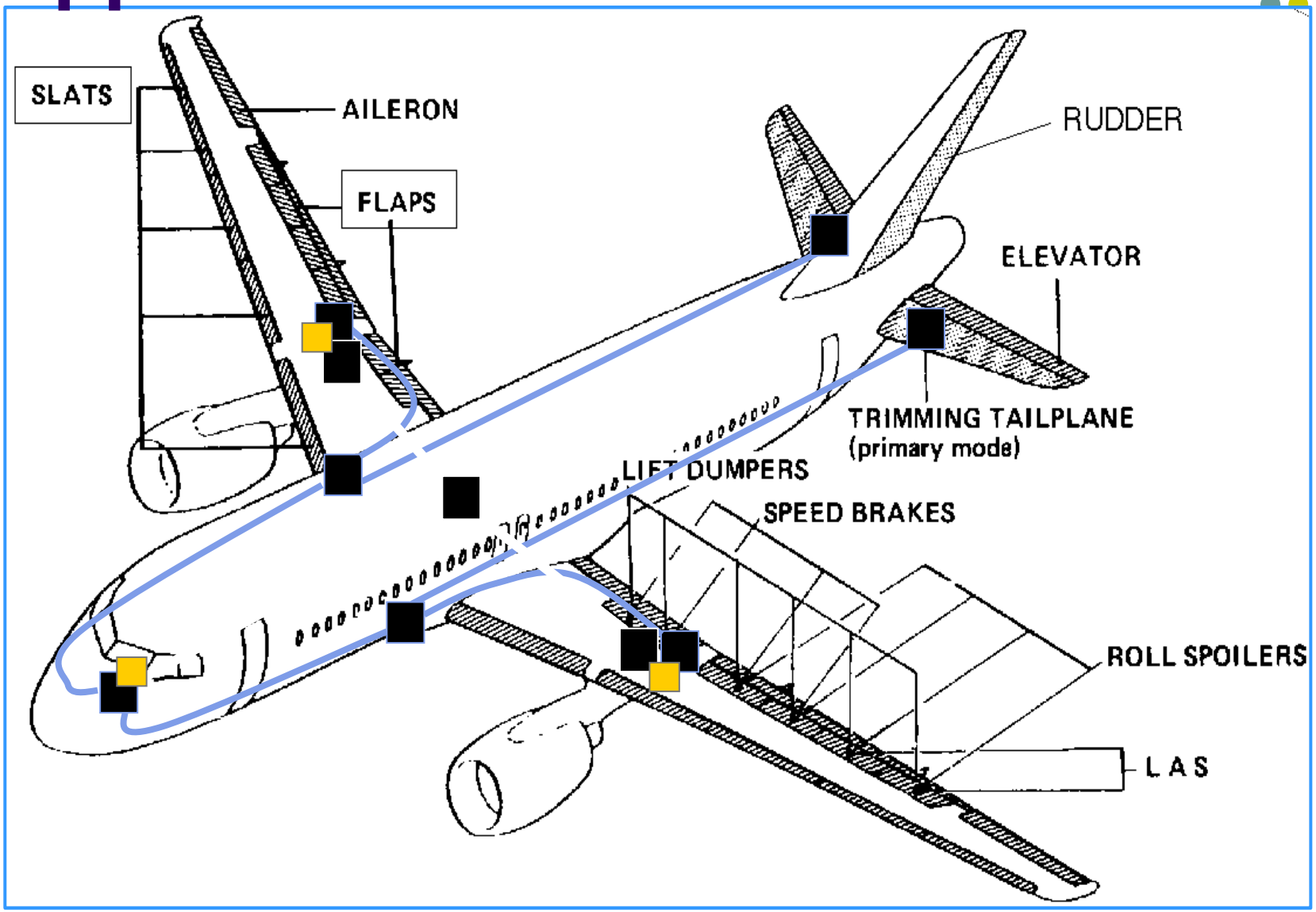
FONDAZIONE BRUNO KESSLER

Software Model Checking Railways Interlocking

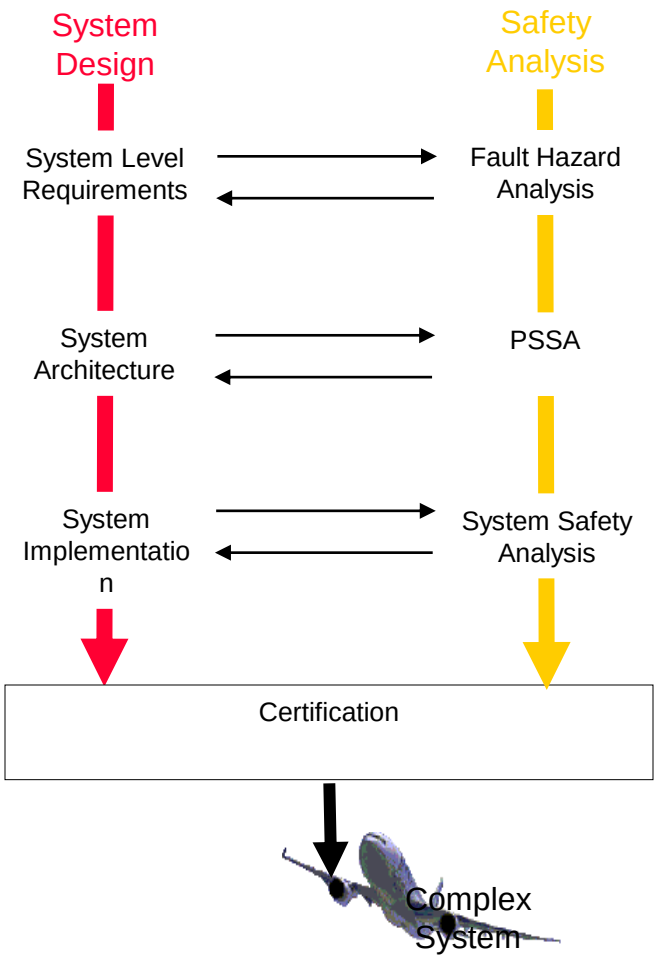
- Software Model Checking
  - Software as input to the model checker
  - Search based on automated abstraction/refinement
  - Increasingly accurate logical models of effect of transitions on memory state
  - Formal verification for automated migration of banking software
- Architecture-aware search
  - A scheduler that sequentializes cooperative threads
  - Many “products” based on same scheduler, threads change
  - Exploit nature of scheduler for specialized search
- The KRATOS software model checker
  - Front-end for threaded C language
  - Based on NuSMV3 and MathSAT
  - SystemC, PLC, Interlocking control, AADL
- Verification of Ansaldo Industrial Interlocking Systems
  - Control logic specified in high level proprietary imperative software
  - Execution controlled by application-specific scheduler
  - Working on integration within industrial verification flow

Projects in Safety Analysis, Fault Tree Analysis, and FMEA

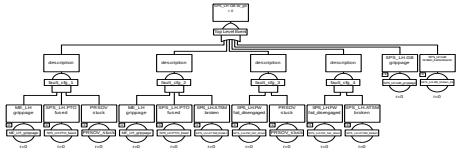
Application Domain



Safety Analysis in Avionics



Fault Tree Analysis



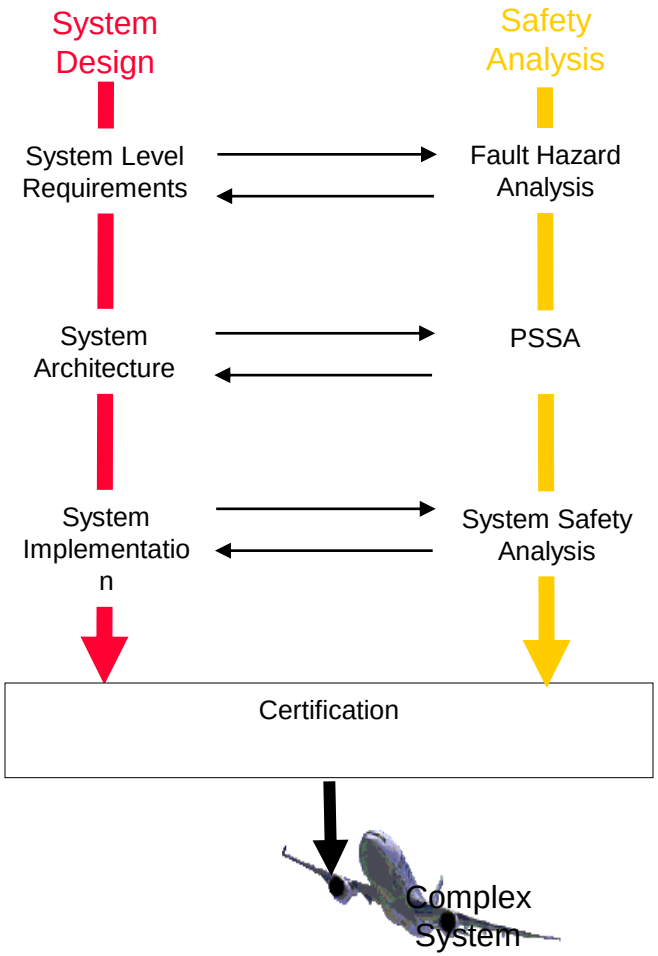
FMEA

Fault	Probability	Intermediate Effect	Final Effect	Severity

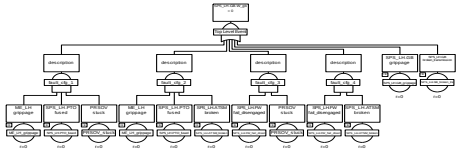
Critical Points

- **Link between System Design and Safety Analysis.**
- **Growing complexity of systems.**

Safety Analysis in Avionics



Fault Tree Analysis



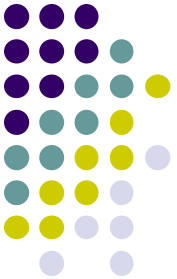
FMEA

Fault	Probability	Intermediate Effect	Final Effect	Severity

Critical Points

- **Link between System Design and Safety Analysis.**
- **Growing complexity of systems.**

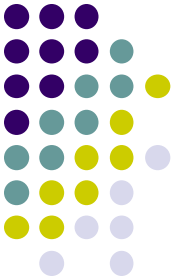
Industrial Partners



- Airbus UK Ltd
- Airbus Deutschland
- Alenia Aeronautica S.p.A.
- Dassault Aviation
- EADS Apsys
- High Integrity Solutions
- Thales

COMPASS

Correctness, Modeling, and Performance of Aerospace Systems



- **Development of an integrated environment featuring the following functionalities**
  - **Functional Correctness**
  - **Safety Analysis**
  - **Performability Analysis**
  - **Diagnosability Analysis**
  - **Requirements Validation**

ThalesAlenia Space

esa

RWTH AACHEN UNIVERSITY

FONDAZIONE BRUNO KESSLER

Contact information

- Alessandro Cimatti
  - Head of Embedded Systems Unit
  - Center for Information Technologies – IRST
  - Fondazione Bruno Kessler, Trento, Italy
- +39 0461 314320
- cimatti@fbk.eu