Model-based approaches to FDIR

Alessandro Cimatti

Fondazione Bruno Kessler

cimatti@fbk.eu

Model-based design and MBSA

- Model based methods are prominent for development of control systems
  - Models to represent high level views of the system
  - Requirements precisely captured
- Model based methods are increasingly applied for the safety assessment of systems under faulty conditions

What can MBSA do?

- From nominal models to extended models
  - Fault extension
- Automated generation of
  - Fault trees
  - FMEA tables
  - Reliability measures

- Tools for MBSA
  - FSAP and XSAP (http://xsap.fbk.eu/)
  - COMPASS (see workshop tomorrow)
- The IMBSA’17 conference @ Trento

Formal Verification, Validation, and Safety Assessment

Model Checking

$$\mathcal{M} \models \varphi$$

Verification & Validation **Safety Assessment**

Formal Verification, Validation, and Safety Assessment

Model Checking

$$\mathcal{M} \models \varphi$$

Fault Injection

$$\mathcal{M} \Longrightarrow \mathcal{M}_{[\mathcal{F}]}$$

Verification & Validation

Safety Assessment

Formal Verification, Validation, and Safety Assessment

Model Checking

$$\mathcal{M} \models \varphi$$

Fault Injection

$$\mathcal{M} \Longrightarrow \mathcal{M}_{[\mathcal{F}]}$$

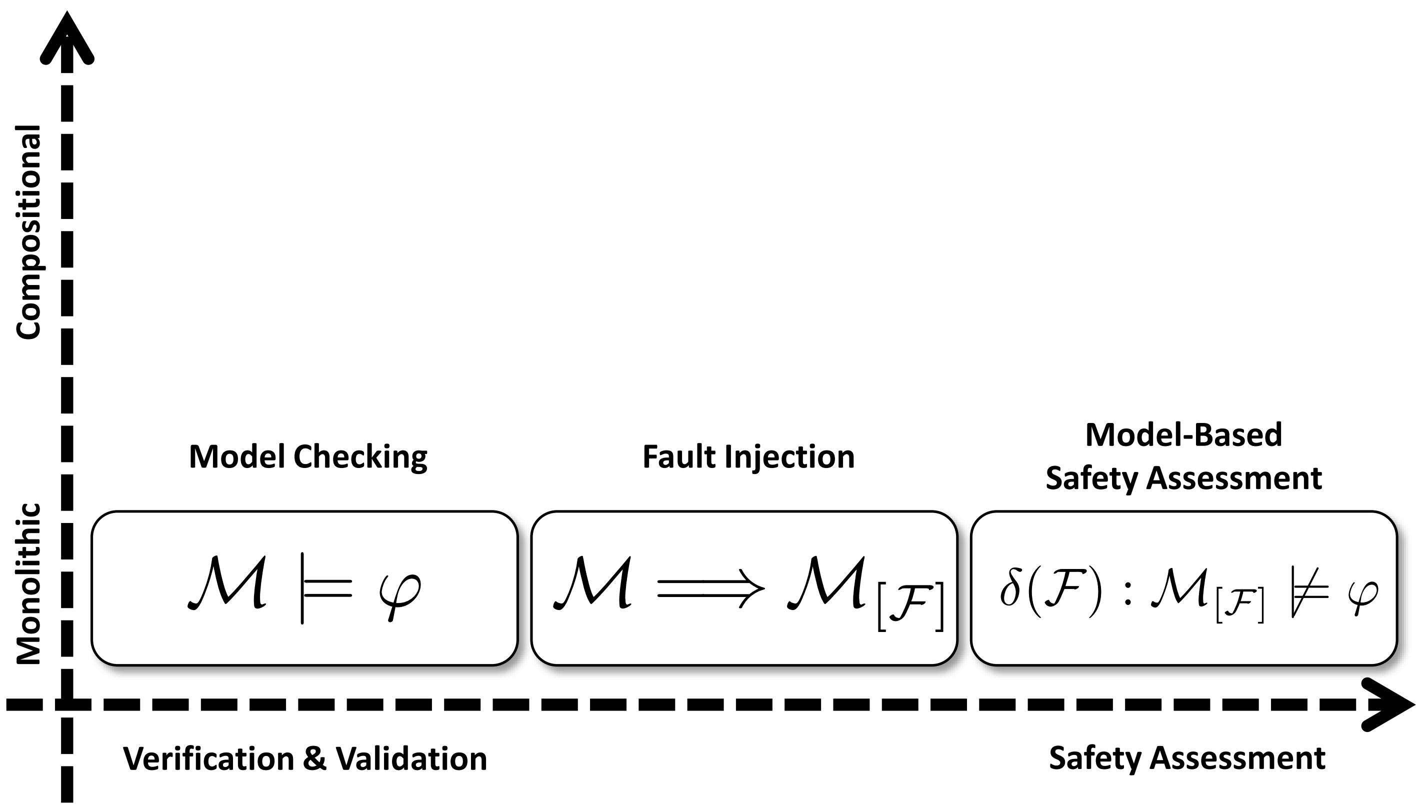
Model-Based Safety Assessment

$$\delta(\mathcal{F}) : \mathcal{M}_{[\mathcal{F}]} \not\models \varphi$$

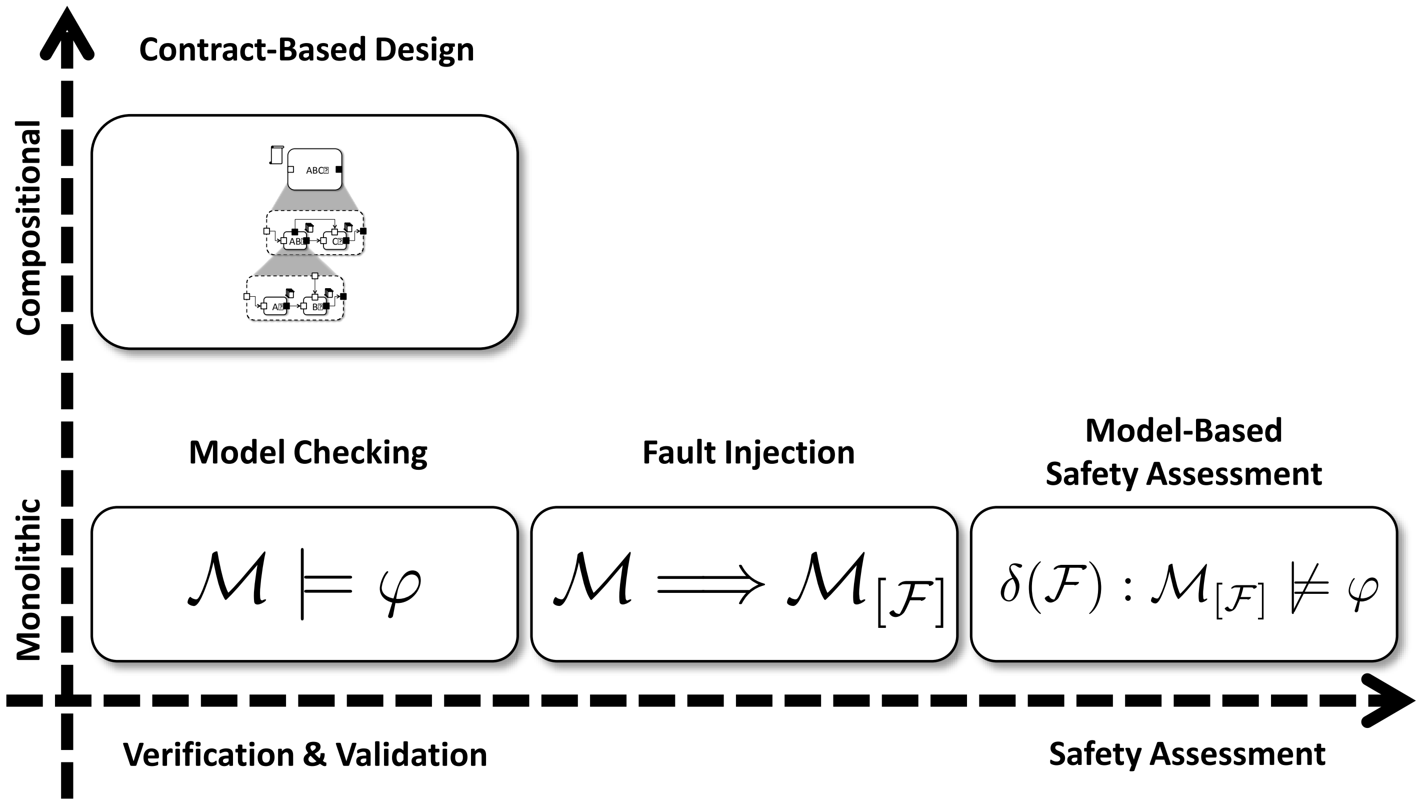
Verification & Validation

Safety Assessment

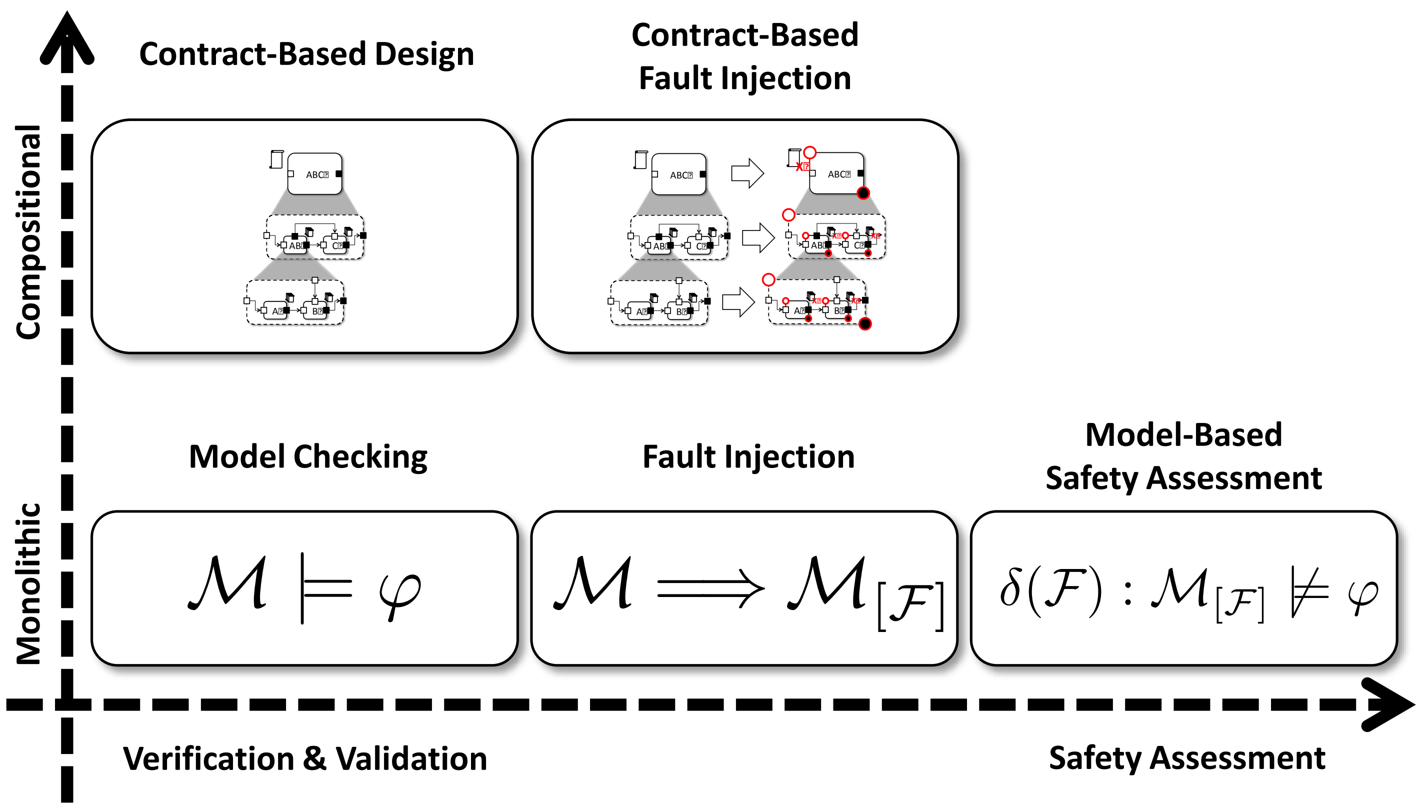
Formal Verification, Validation, and Safety Assessment



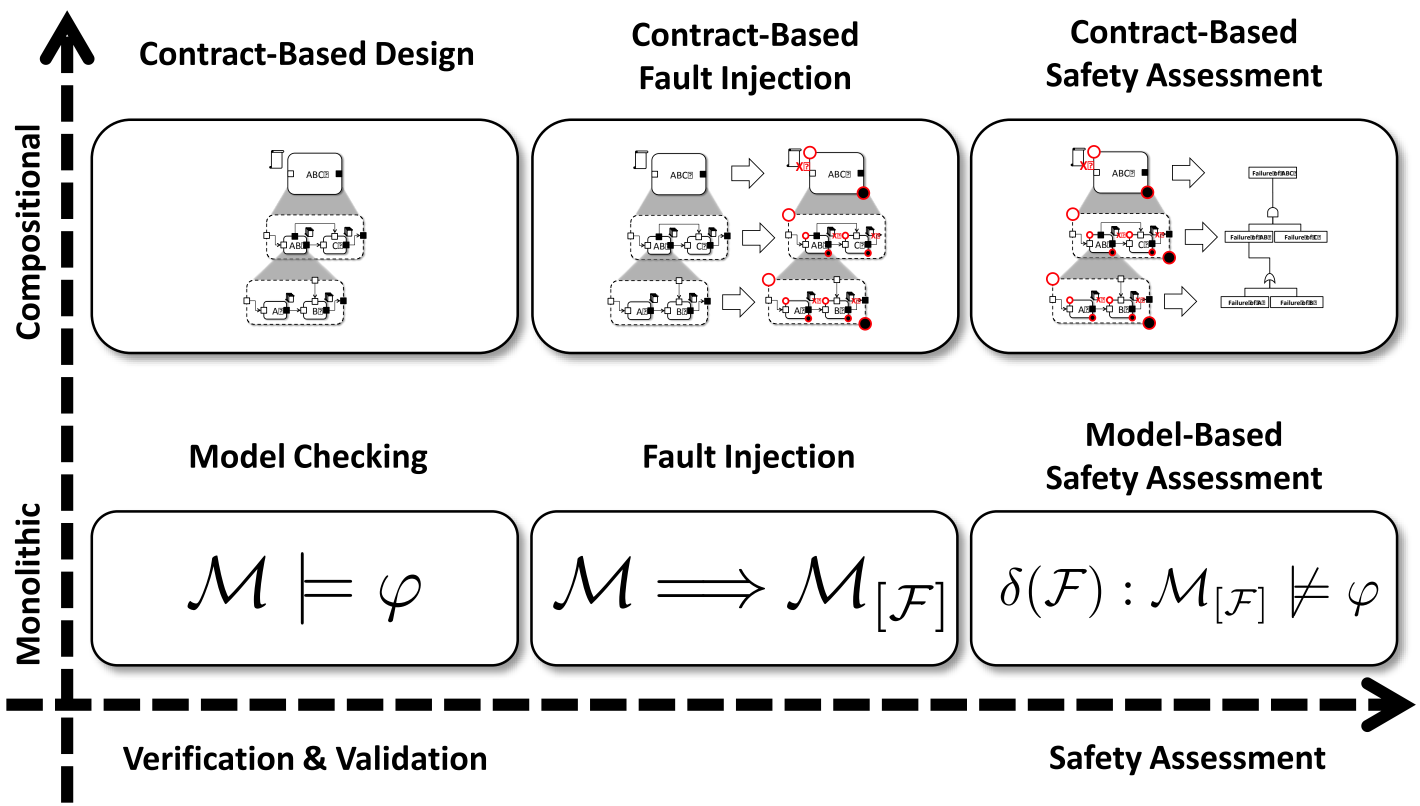
Formal Verification, Validation, and Safety Assessment



Formal Verification, Validation, and Safety Assessment



Formal Verification, Validation, and Safety Assessment



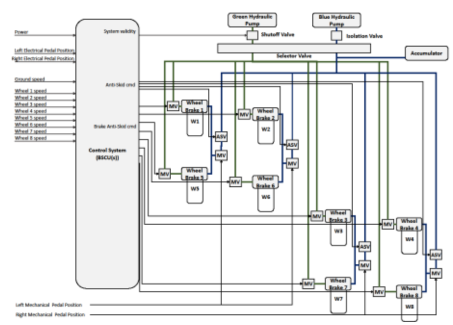
Tool chain

- Infinite-state transition systems
  - The **OCRA** tool for contract-based design
    - http://ocra.fbk.eu/
  - The **nuXmv** model checker
    - http://nuxmv.fbk.eu/
  - The **xSAP** platform for safety analysis
    - http://nuxmv.fbk.eu/

- Hybrid systems
  - **HyCOMP** as a model checker
    - http://hycomp.fbk.eu/

A Wheel Brake System

- Control brake for aircraft wheels
- Redundancy
  - Multiple BCSU
  - Hydraulic plants
- Functions
  - Asymmetrical braking
  - Antiskid
    - Single wheel/coupled
    - depending on control mode

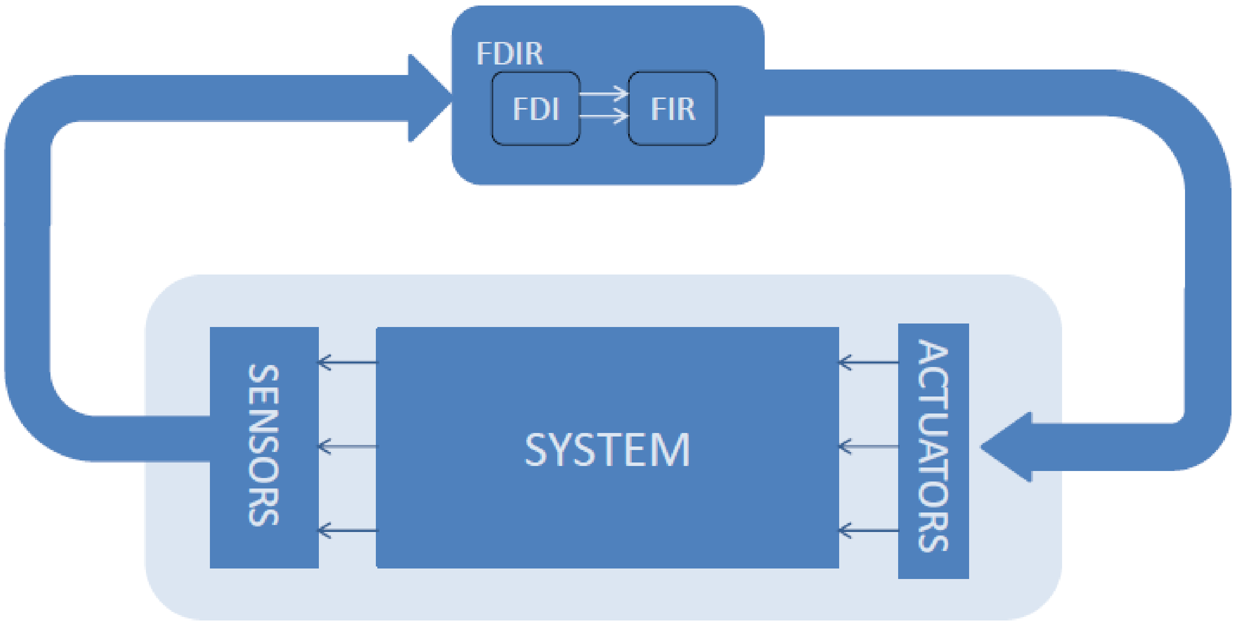


Applications

- Joint project with Boeing on MBSA
  - Formal Design and Safety Analysis of AIR6110 Wheel Brake System [CAV'15]
- Adopted in NASA project on analysis of NextGen
  - Comparing Different Functional Allocations in Automated Air Traffic Control Design [FMCAD'15]
- The COMPASS tool chain
  - AADL modeling language
  - Several projects funded by the European Space Agency
  - Specific design technique for FDIR

From MBSA
to model-based FDIR

Fault Detection, Identification and Recovery (FDIR)



FDIR …

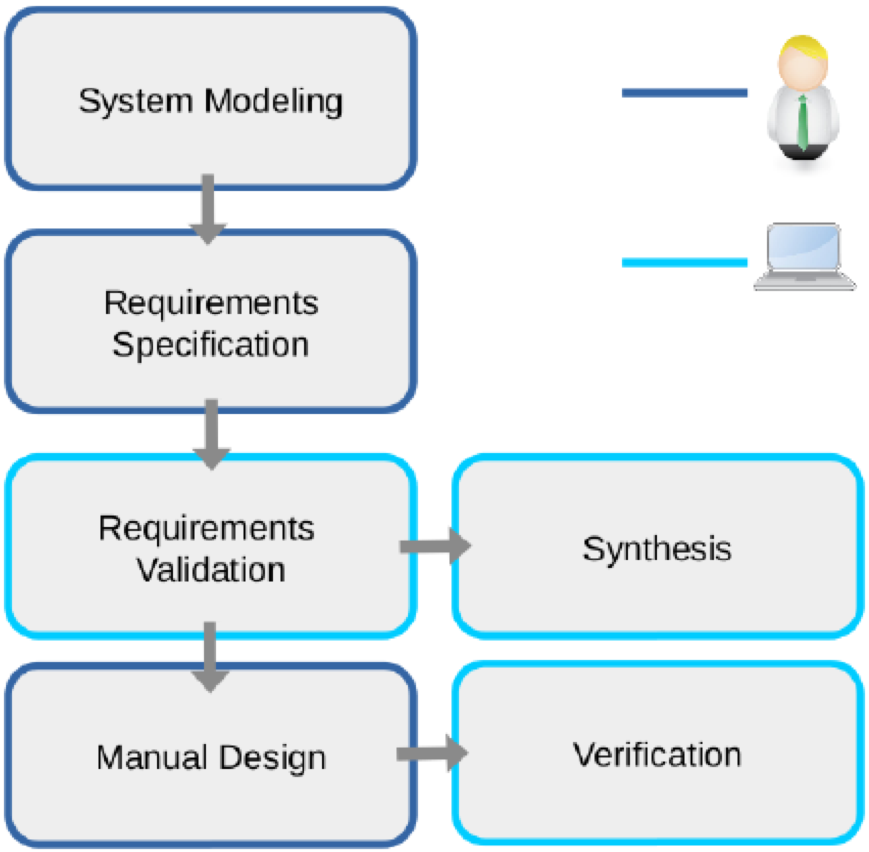
- The development of Fault Detection, Isolation and Recovery is poses additional challenges, due to the partial observability of the system state that must be dealt with at run-time, and can only partly be tackled by sensors.
- Need for
  - Early validation of the FDIR design
  - Simplification of certification process
  - Higher dependability of the system
  - Reduction of costs in terms of design effort, implementation, and possible reuse of FDI components

Focus on FDI

How to design an FDI component?

- Partial observability
- Subtle interaction of faults and nominal events

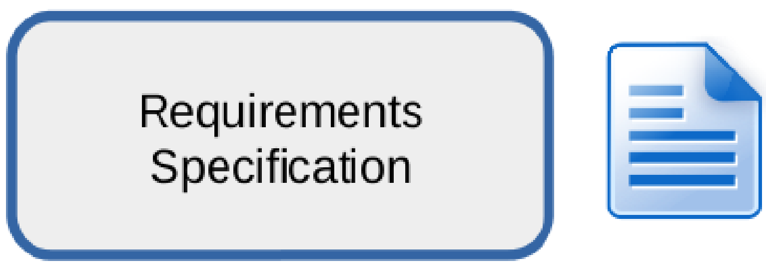
Phases for FDIR design



System Modeling

- How does the system work?
- What are the faults?
- What sensors are available?

How to specify an FDI component?

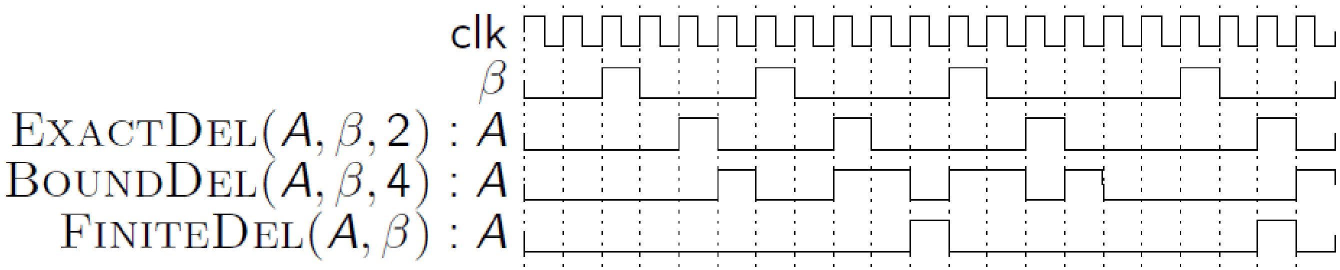


- What alarms should the FDI provide?
- What are acceptable delays?
- When should an alarm be ignored?

Patterns for FDIR specification

Delay between the diagnosis condition β and the alarm A

Whenever the fuel valve gets stuck-closed, the FDI should raise the alarm within 4 time-units (BoundDel)



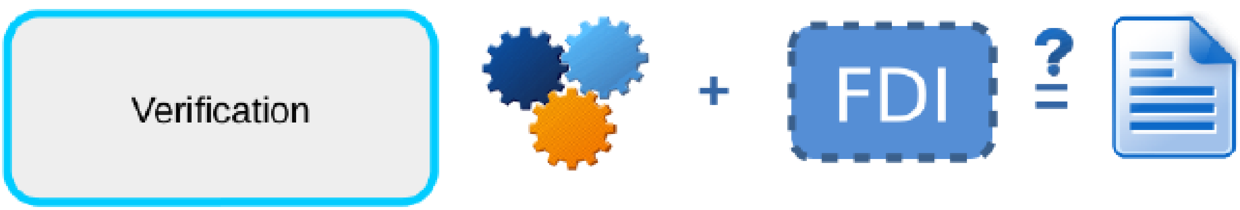
FDIR requirements validation

By themselves:

- Inconsistencies: they cannot be all satisfied simultaneously
- Over-constraining: A good behavior is not possible
- Under-constraining: A bad behavior is possible

Against the System: e.g., diagnosability

FDI verification

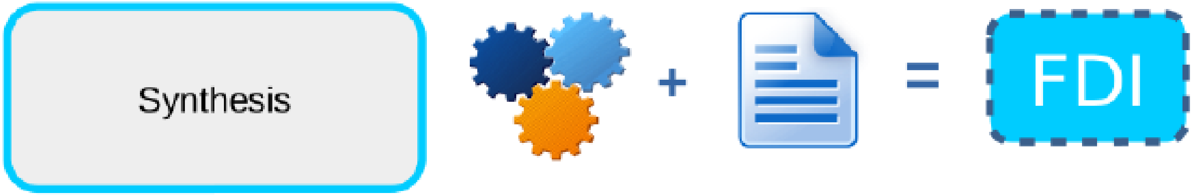


Testing:

- Incomplete analysis
- Requires a detailed implementation

Formal Model + Formal Requirements ⇒ Model-Checking

FDI synthesis



Advantages:

- Proof of realizability.
- Quick way to obtain a prototype.

Challenges:

- Computationally hard (sometimes undecidable).
- Hard to understand for humans.

Safety condition defining a (set of) configurations of the system.

- Bad configuration of the system:

  Both engines are off

- Fault has occurred:

  The fuel valve is stuck-closed

- Conditions on the evolution of the system:

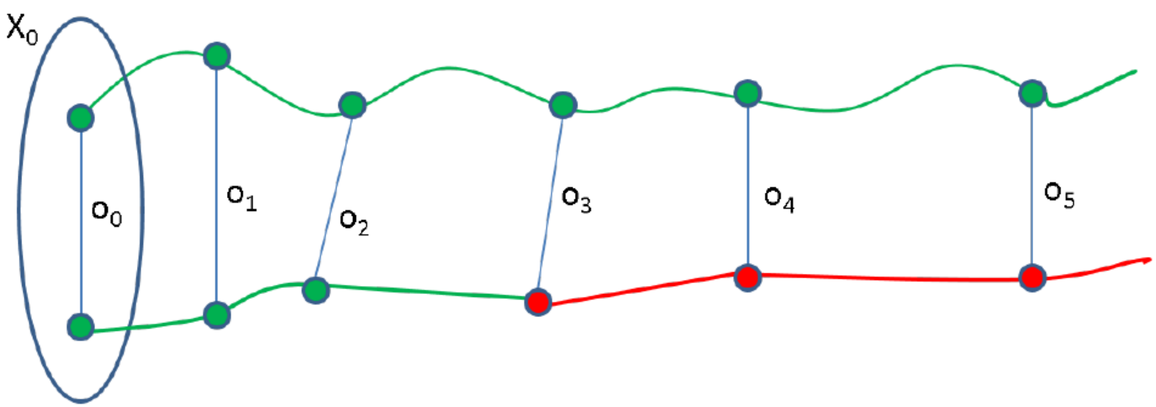
  The fuel valve has been stuck-closed for at least 3 time-units and the engines are currently on

Diagnosability: global vs local

Always possible to satisfy an Alarm Condition?

No! Observations might not be sufficient to disambiguate:

Critical Pair



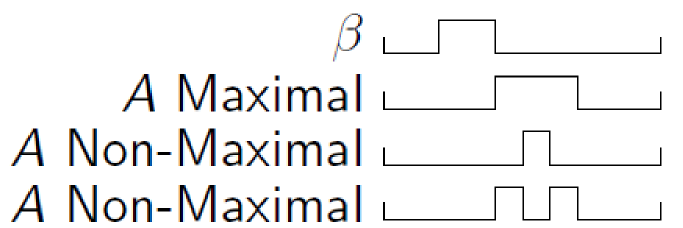
- No critical pair = System diagnosable [Sampath]
- Too coarse-grained? E.g., 1 critical pair?

$\Rightarrow$ Trace Diagnosability: Diagnose as much as possible

Maximality

- The alarm should go up as soon and for as long as possible

$$\textsc{BoundDel}(A, \beta, 4)$$



Maximality removes ambiguity

Sensor synthesis

- Synthesis of observability requirements
  - Have we got enough sensors?
  - Design space exploration
- Find the sets of sensors that guarantee diagnosability
- Reduced to a parameterized model checking problem

Fault Recovery and Recoverability

- Recovery strategy as a mapping between alarms and suitable recovery actions
- Effectiveness of recovery:
  - Is the execution of the recovery strategy sufficient to restore a functionality or achieve a desired effect?

- Recoverability – existence of suitable recovery strategy
  - Tackled with planning techniques
  - Needs to take into account adversarial environment

Conclusions and Challenges

- Formal account of FDIR
- Supported by formal tools
- The need for a structured process (see next talk)

- Challenges:
  - FDI-system connection models
    - Synchronous vs asynchronous composition
    - Cycle-based vs event based
  - Centralized vs distributed approach
  - Temporal epistemic logic as back-end
  - Fault-tolerance evaluation of redundancy architectures

Some (mildly) provocative statements

- Need for case studies
  - Fundamental step for FDIR community building
- Modeling language should not be a blocking issue
  - The techniques are largely independent
  - Towards COMPASS-STAR?
- Lack of GUI should not be a blocking issue
  - Textual language + artifacts viewers
    profitably applied in industrial settings
- Is there a need for a change?
  - "Existing process is good enough"
- "excel is not enough for designing spacecrafts"
  [BB, 2015, personal communication]