HASDEL

Hardware Software Dependability for Launchers

Final Presentation Days

Tuesday, 09 December 2014

Presented by David LESENS & Marco BOZZANO

esa

AIRBUS DEFENCE & SPACE

FONDAZIONE BRUNO KESSLER

RWTH AACHEN UNIVERSITY

Agenda

- Introduction – Objectives of the HASDEL project
- The HASDEL approach
- Use cases
  - Equipment reintegration
  - ATV data handling system architecture
- Demonstration
- Conclusion

Objectives of the HASDEL project

Objectives

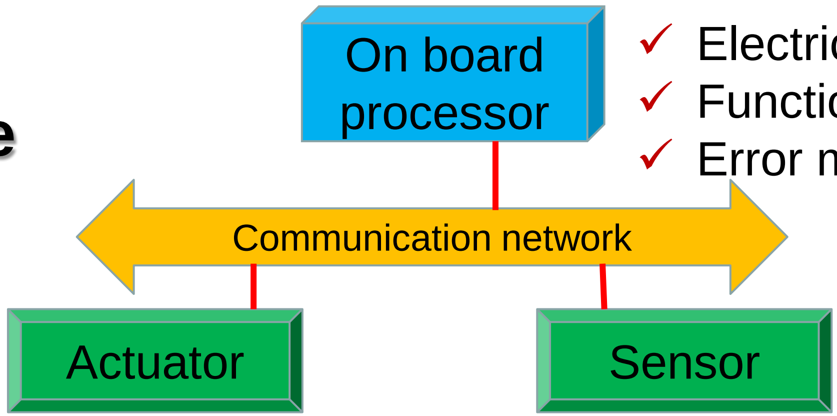
- Analysing the specific needs of launcher systems in the domain of RAMS (*Reliability, Availability, Maintainability and Safety*) analysis
- Extending the COMPASS toolset with these specific needs

COMPASS
Correctness, Modeling and Performance of Aerospace Systems

Launchers and space transportation vehicles specificities

- High level of criticality
- Hard real time requirements
- Functional complexity
- RAMS requirement complexity (e.g. management of redundancies)

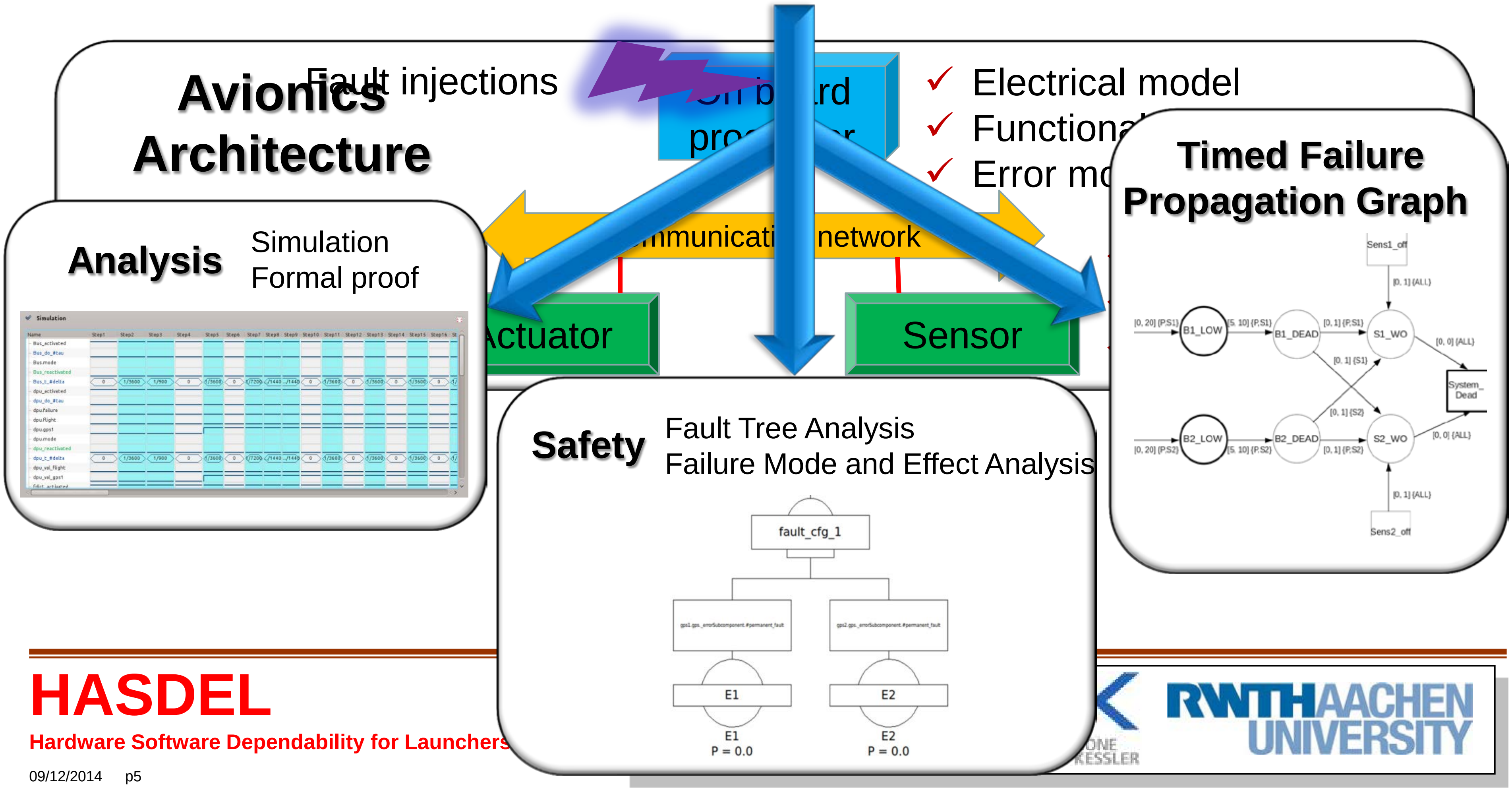
Avionics Architecture



On board processor:
- ✓ Electrical model
- ✓ Functional model
- ✓ Error model

Actuator:
- ✓ Electrical model
- ✓ Functional model
- ✓ Error model

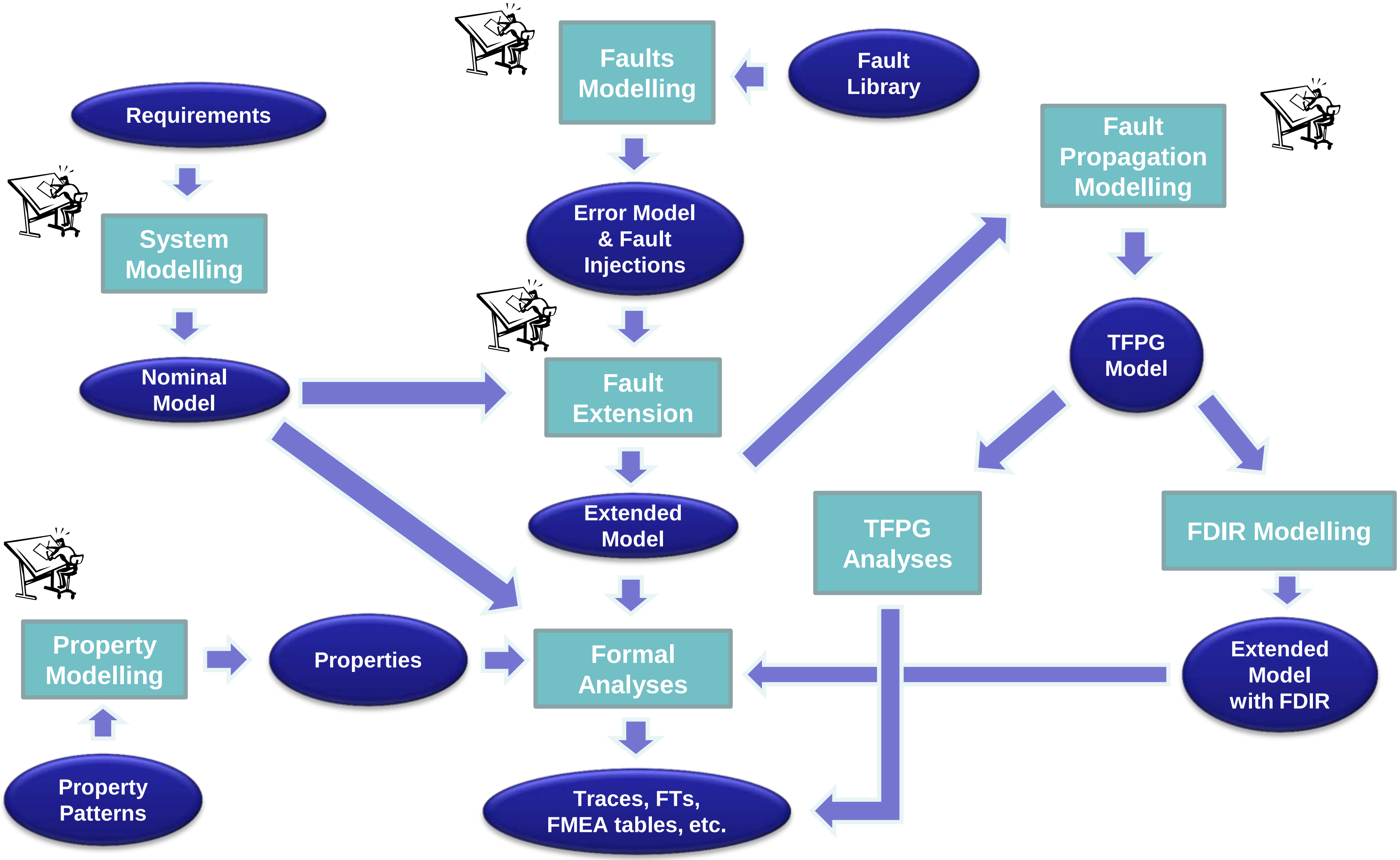
Sensor:
- ✓ Electrical model
- ✓ Functional model
- ✓ Error model


Avionics Architecture
Fault injections
Electrical model
Actuator
Sensor
Analysis
Simulation
Formal proof
Safety
Fault Tree Analysis
Failure Mode and Effect Analysis
fault_cfg_1
E1
E1
P = 0.0
E2
E2
P = 0.0
Timed Failure Propagation Graph
Sens1_off
B1_LOW
B1_DEAD
S1_WO
System_Dead
B2_LOW
B2_DEAD
S2_WO
Sens2_off
RWTH AACHEN UNIVERSITY

Agenda

- Introduction – Objectives of the HASDEL project
- The HASDEL approach
- Use cases
    - Equipment reintegration
    - ATV data handling system architecture
- Demonstration
- Conclusion

The HASDEL Approach: Flow

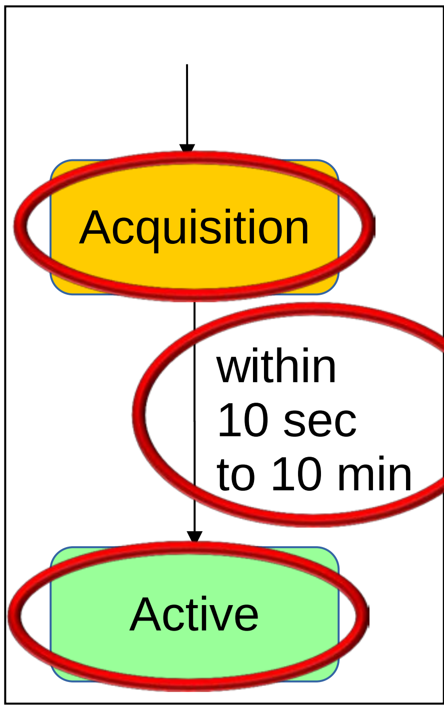


The HASDEL Toolset

- Comprehensive toolset
  - Modelling in SLIM, a variant of AADL
  - Implementing the V&V flow and analyses illustrated in previous slides
- Based on state-of-the-art model checking tools

SLIM language

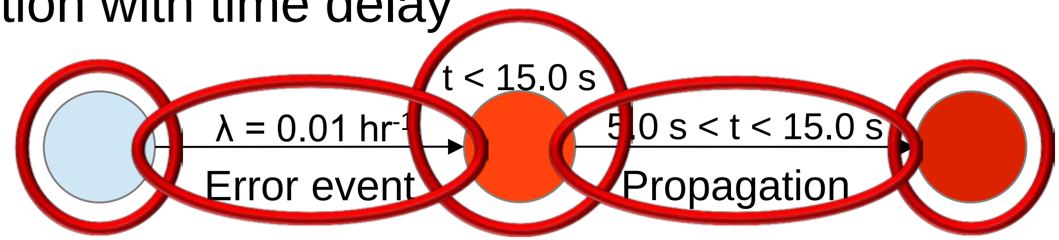
- HASDEL enables modelling of:
  - Behaviour using modes and states
  - Data shared by connections and flows
  - Timed/hybrid dynamics using clocks and continuous variables




```
device gpsDevice
   features
      measurement : out data port bool default false;
end gpsDevice;

device implementation gpsDevice.i
   flows
      measurement := true in modes (active);
   modes
      acquisition        : activation mode urgent in 10 min;
      active : mode;
   transitions
      acquisition -[ within 10 sec to 10 min ]-> active;
end gpsDevice.i;
```

SLIM: Timed Failure models

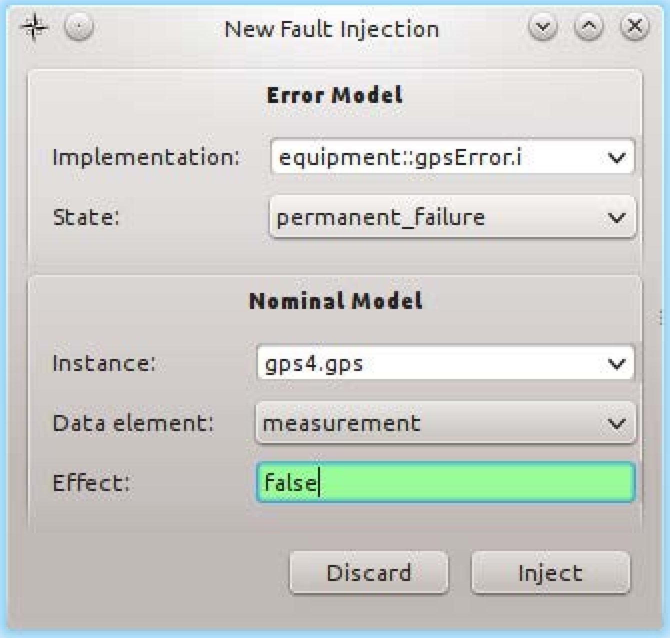
- An example: modelling error propagation
  - First transition with probabilistic rate
  - Next transition with time delay




```
error model SubError
features
   err_prop : out error propagation;
end SubError;

error model implementation SubError.Impl
events
   err_evt : error event occurrence poisson 0.01 per hour;
states
   e_nominal : initial state;
   e_triggered : error state urgent in 15 sec;
   e_propagated : error state;
transitions
   e_nominal -[err_evt]-> e_triggered;
   e_triggered -[err_prop between 5 sec and 15 sec]-> e_propagated;
end SubError.Impl;
```

Fault injections

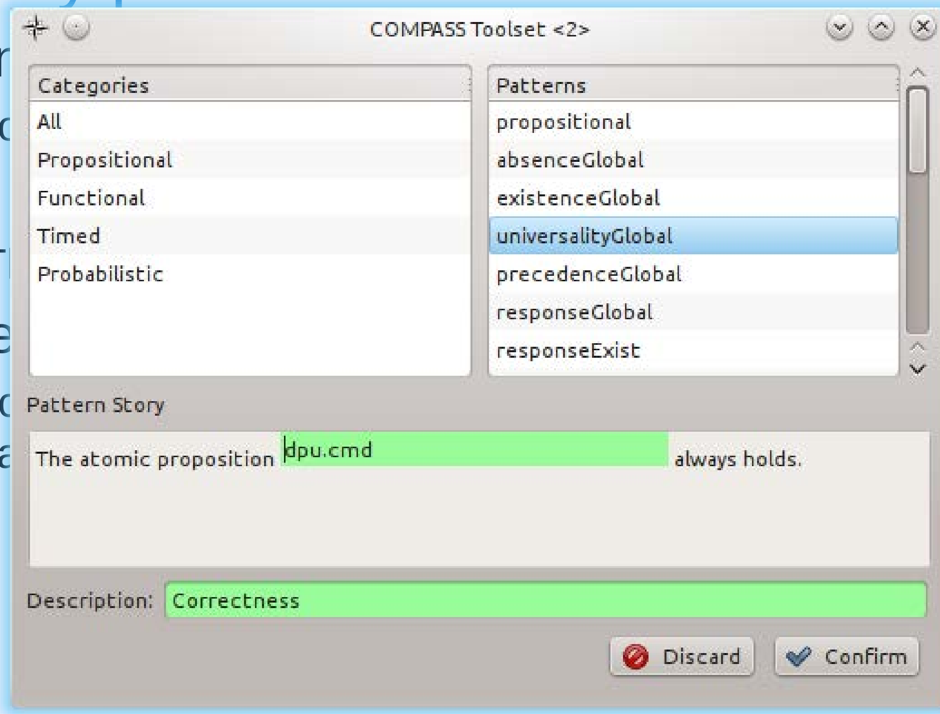
- Nominal and failure models are coupled by fault injections
- Example:



- "When the error state is permanent_failure, gps.measurement becomes false"

Timed Property Patterns

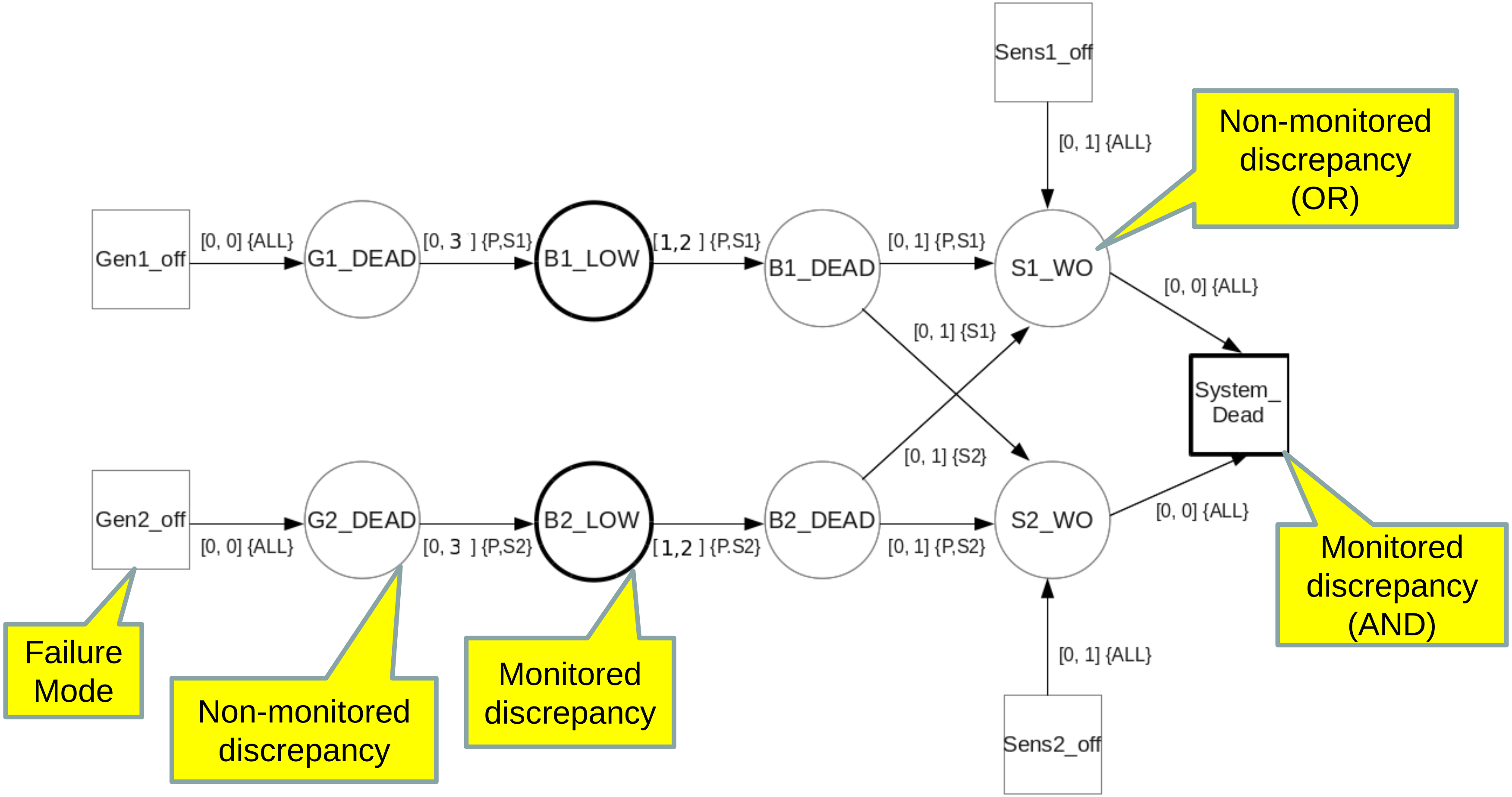
- HASDEL enables modelling of properties via instantiation of property patterns
- Classes of property patterns
  - Functional patterr
    - E.g., absenceGlo
  - Timed patterns
    - E.g. absence: "T ... me units"
  - Probabilistic patte
    - E.g., probabilistic ... etween Time1 and Time2 with proba



Timed Failure Propagation Graphs (TFPGs)

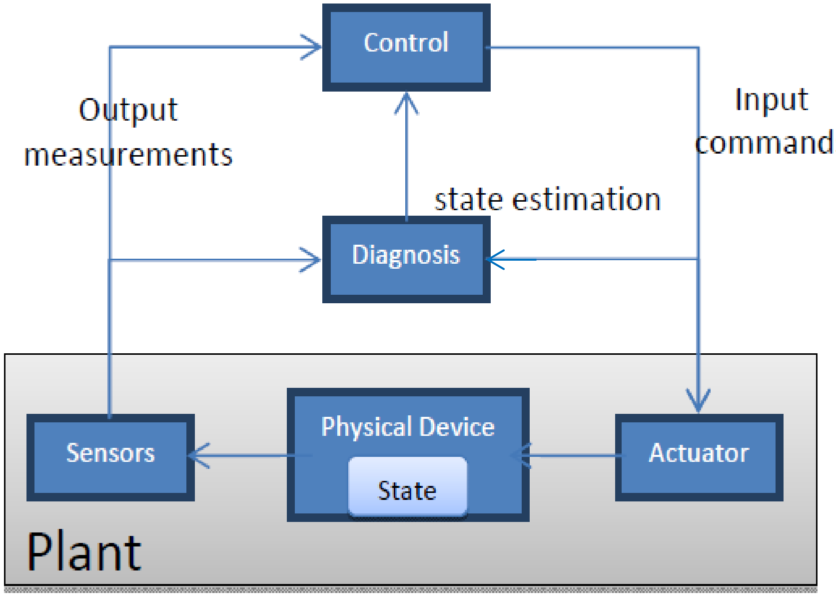
- Graph-like formalism to describe failure propagation
  - Faults
  - Interaction between different faults (AND/OR semantics)
  - Propagation delays (time intervals)
  - Context information (system modes)
  - Effects of fault propagation (discrepancies)
  - Observability (monitored and non-monitored discrepancies)
- TFPGs can be used for diagnosis and prognosis
- TFPG analyses supported by HASDEL
  - Validation of a TFPG with respect to a system model
  - Validation of TFPG as a model for diagnosis
  - Automatic synthesis of a TFPG from a system model

An Example TFPG



Fault Detection, Isolation and Recovery (FDIR)

- Diagnosis system
  - Plant (Physical Device) in closed loop with a controller
  - Control is responsible for commanding actuators
  - Diagnosis tracks the hidden state of the plant over time



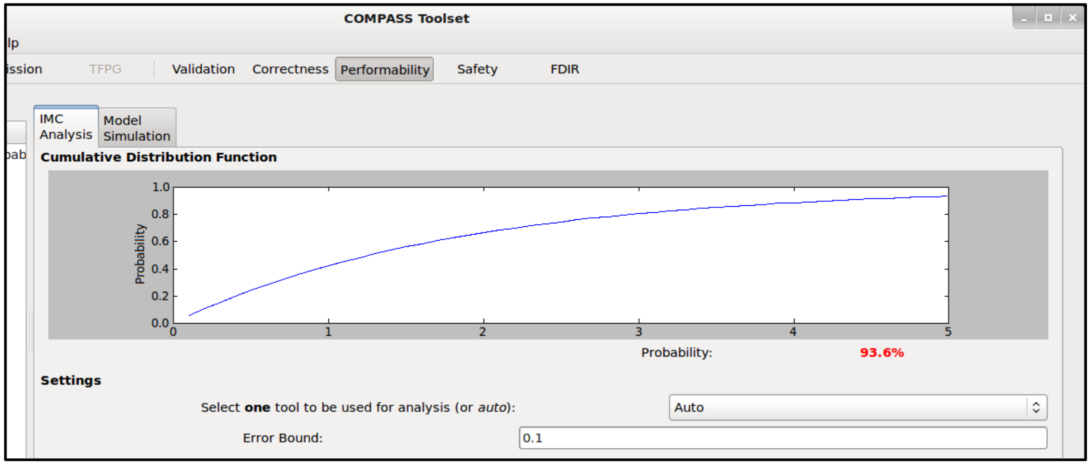
- Partial observability
  - Only a limited number of observables (sensors) are available

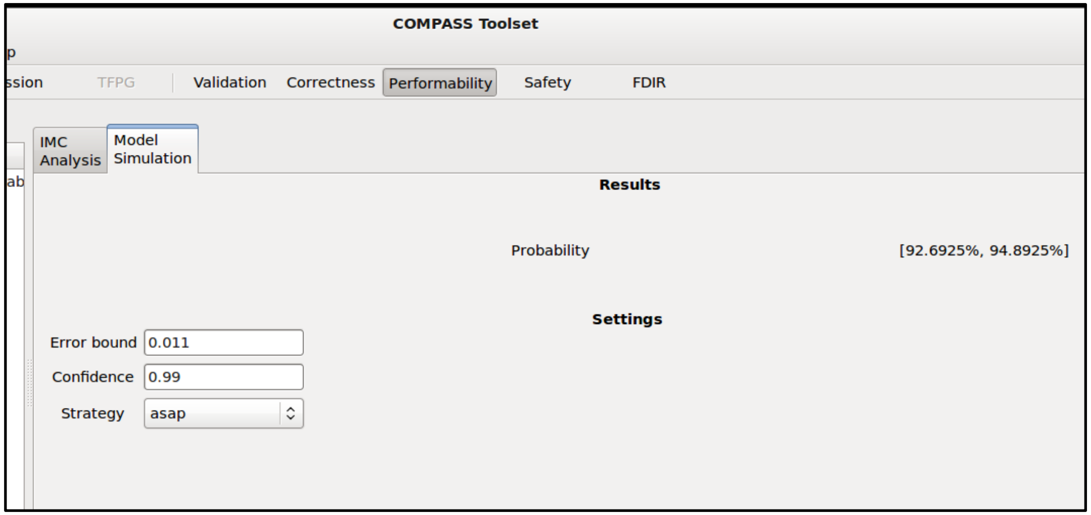
Timed FDIR Analyses

- Timed Diagnosability Analysis
  - Check if there exists a diagnoser that can infer at run-time accurate and sufficient information to diagnose system properties (e.g., occurrence of faults)
  - It helps identifying if enough observables are available for building an FDIR sub-system
  - E.g.: “fault F is diagnosable within T time units”
- Timed FDIR effectiveness analysis
  - Check the effectiveness of an existing FDIR sub-system
  - Fault detection, fault isolation and fault recovery analyses
  - E.g.: “fault F can be detected by the FDIR sub-system within T time units”

Probabilistic risk analysis

- Performability analysis
  - Investigate model reliability
- Numerical analysis
  - Based on Markov Chain model checking

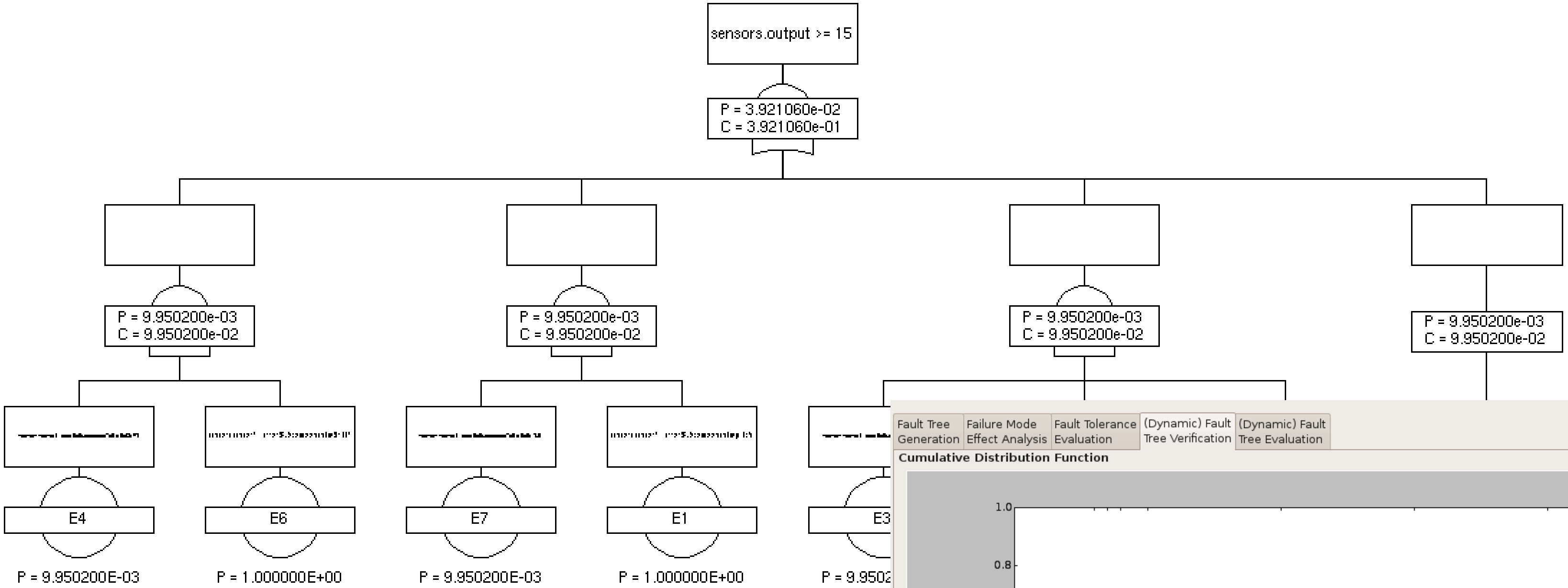




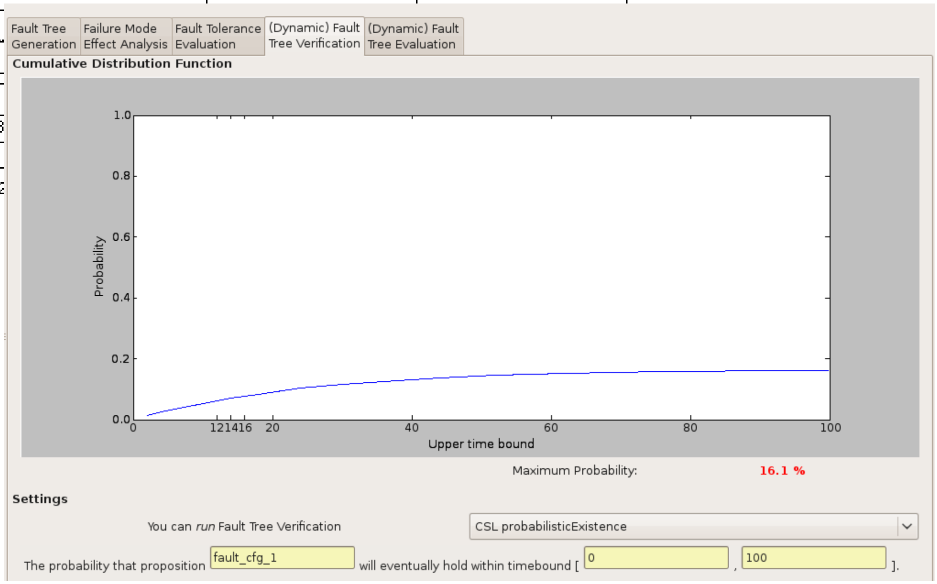
- Statistical analysis
  - Based on Monte Carlo method

Fault Tree analysis

- Supports Fault Tree generation and evaluation



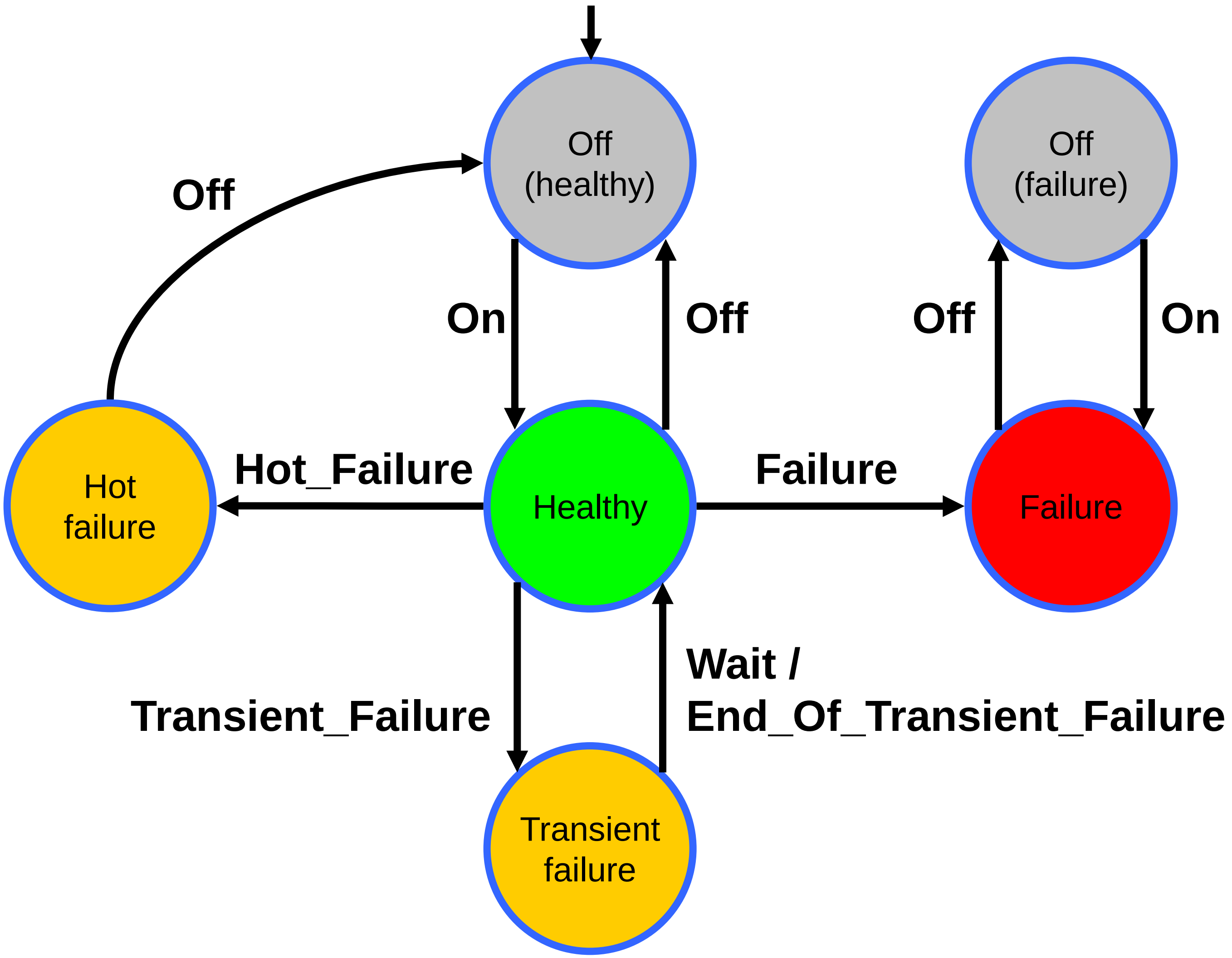
- And Verification



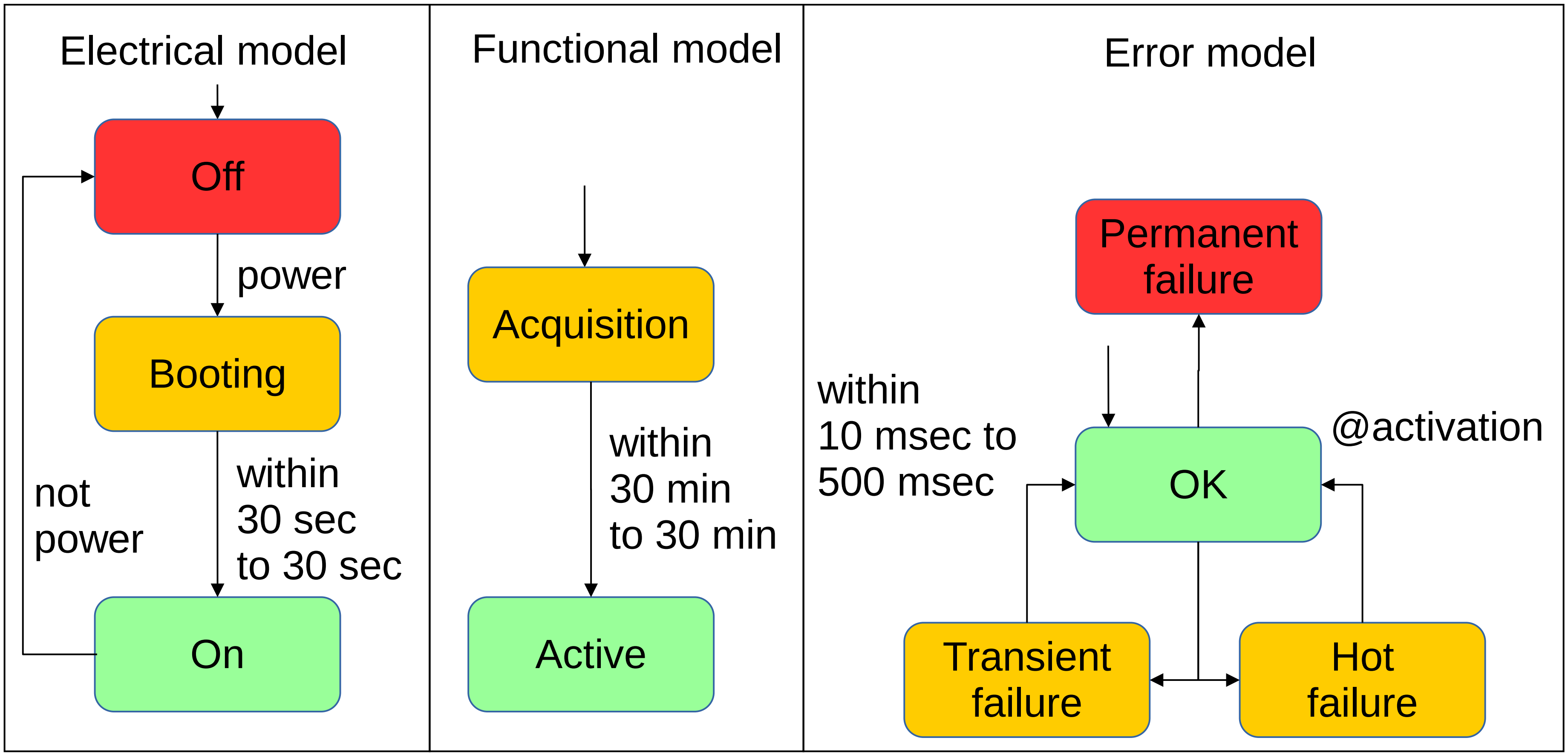
Agenda

- Introduction – Objectives of the HASDEL project
- The HASDEL approach
- Use cases
  - Equipment reintegration
  - ATV data handling system architecture
- Demonstration
- Conclusion

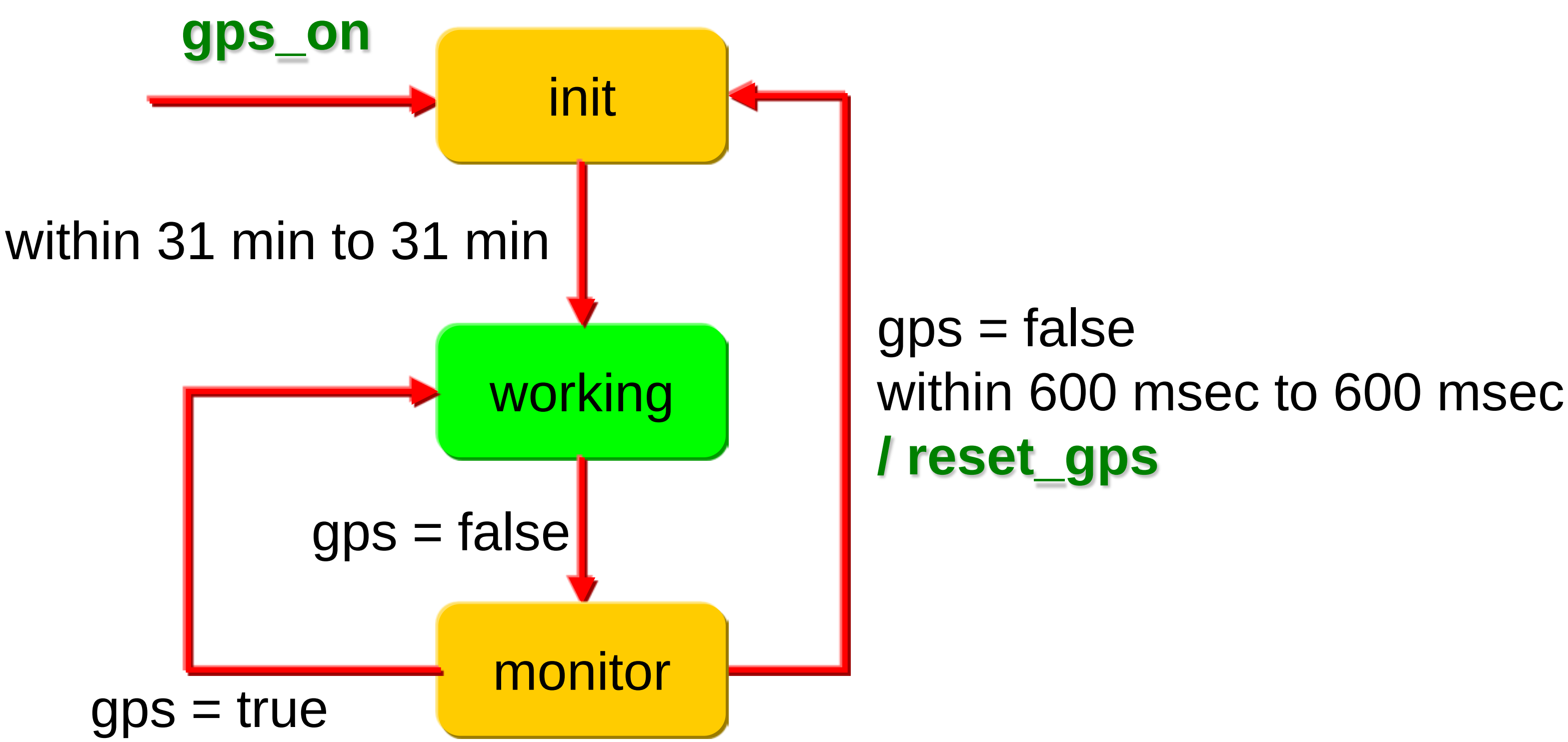
Equipment reintegration



Equipment reintegration modelling principle



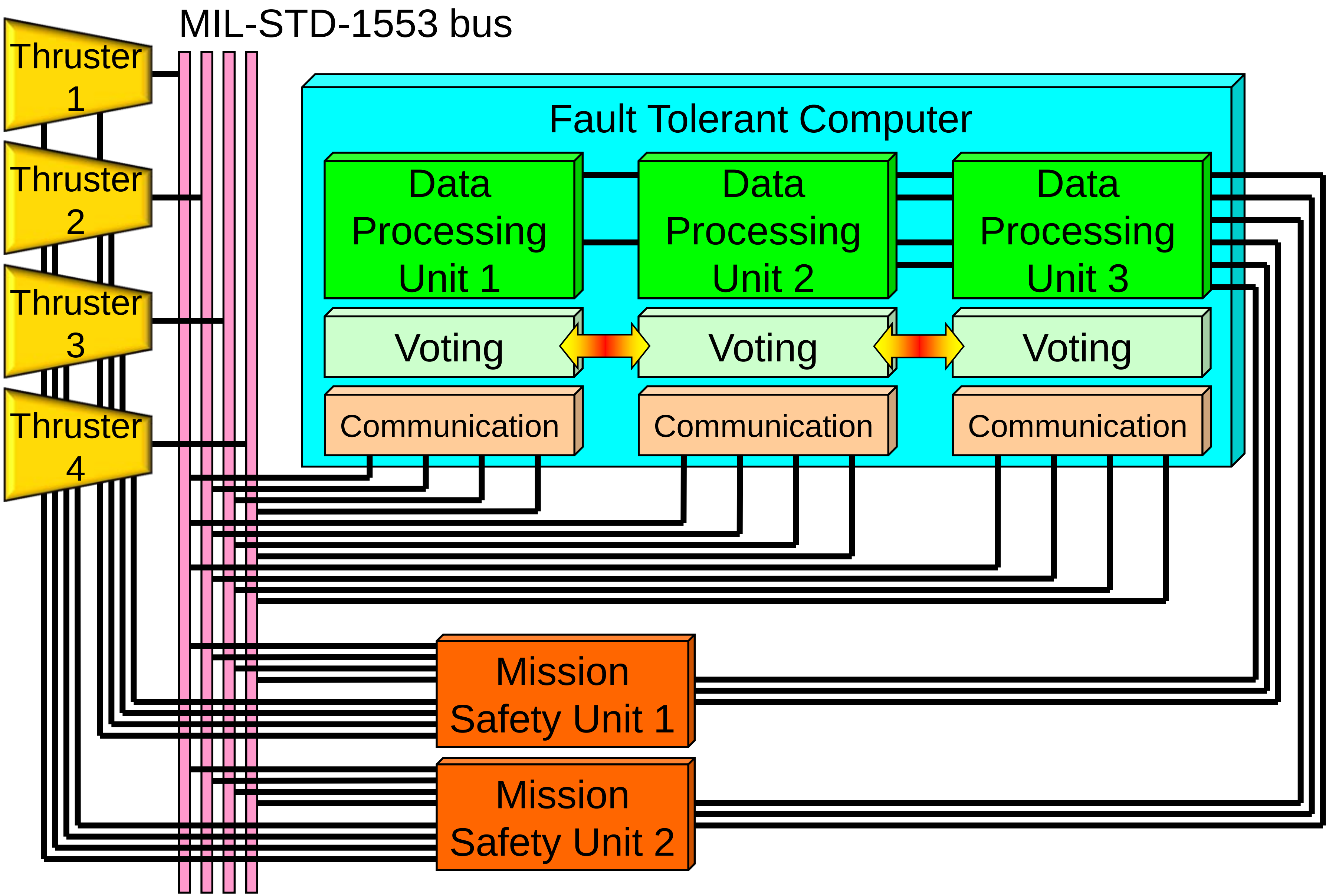
Simplified FDIR



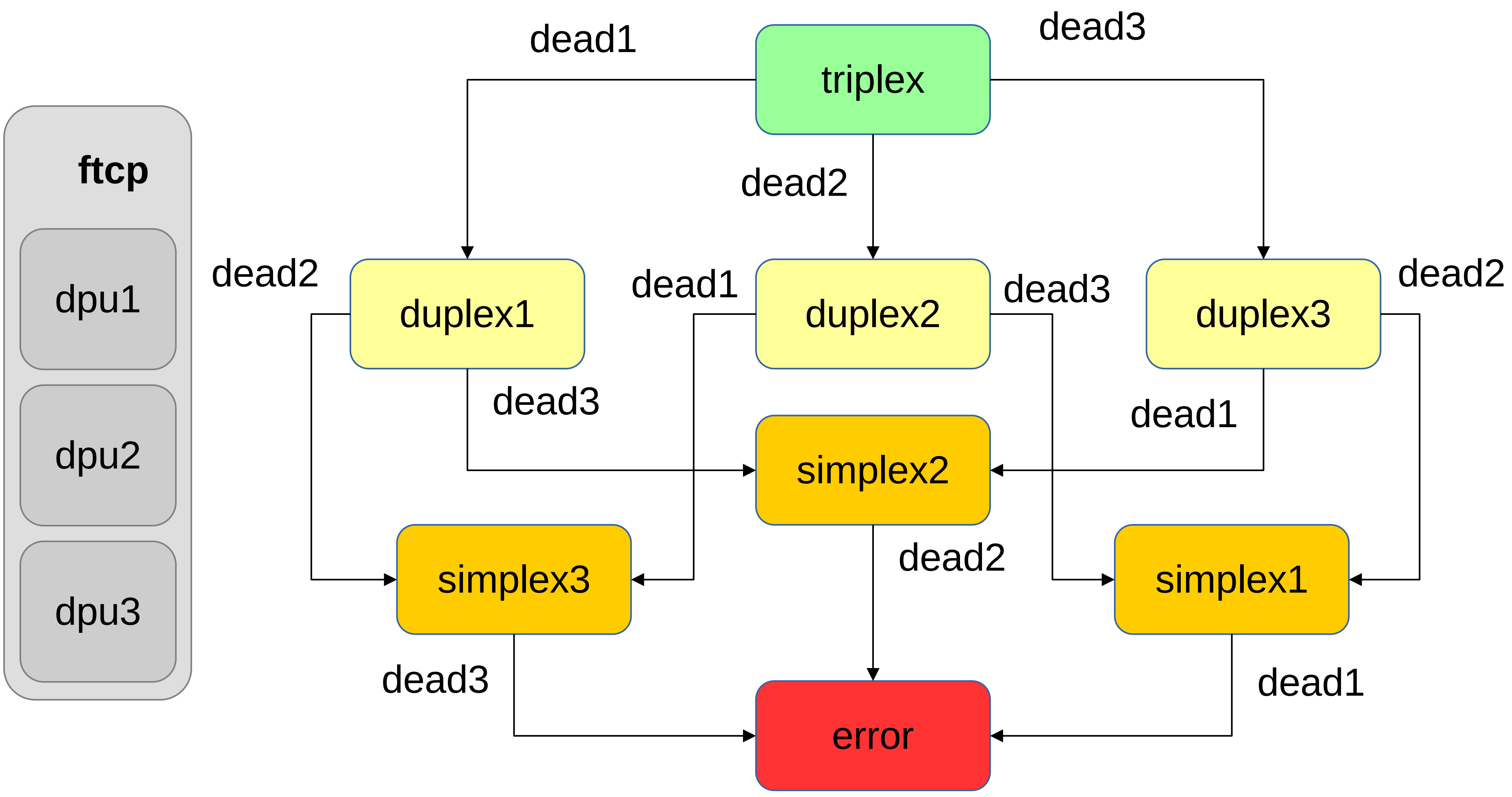
Agenda

- Introduction – Objectives of the HASDEL project
- The HASDEL approach
- Use cases
  - Equipment reintegration
  - ATV data handling system architecture
- Demonstration
- Conclusion

ATV data handling system architecture



Fault Tolerant Computer model



Properties

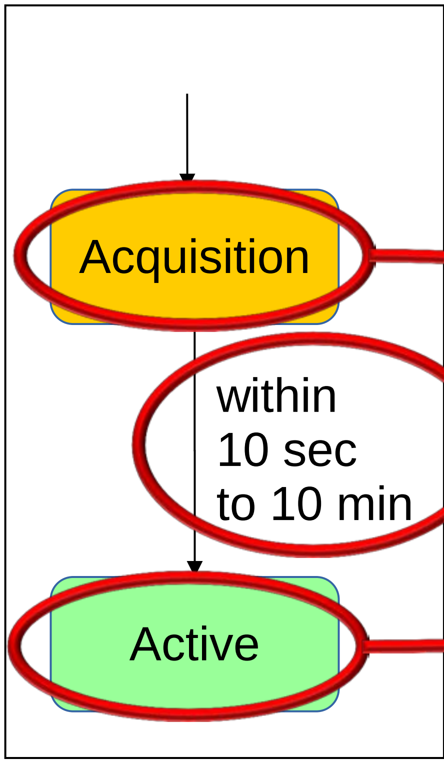
Property type	Property description
expectedTime	"The expected time to reach a state where the proposition *not failure* holds."
longRunAverage	"The long-run average time spent in states where the proposition *not failure* holds."
probabilisticInvariance	"The probability that *not failure* holds continuously within timebound [*0 min* , *2 min*]"

Agenda

- Introduction – Objectives of the HASDEL project
- The HASDEL approach
- Use cases
  - Equipment reintegration
  - ATV data handling system architecture
- Demonstration
- Conclusion

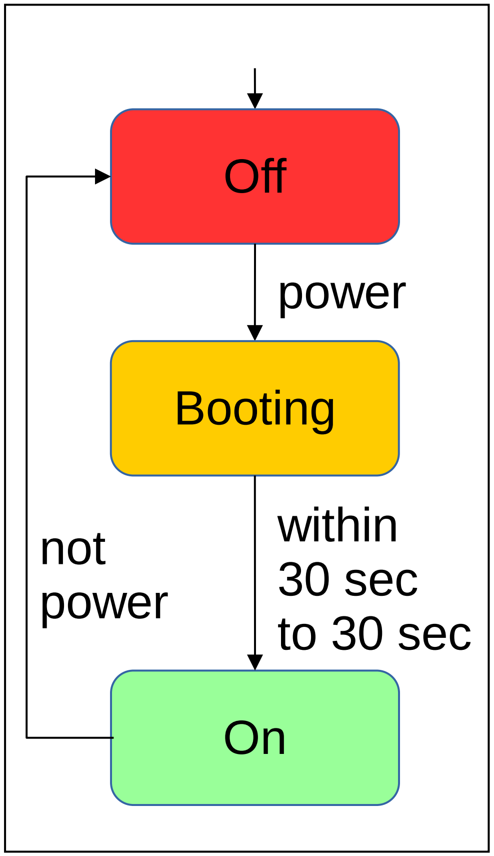
Functional view

The sensor provides correct measurement




```
device gpsDevice
        features
                measurement : out data port bool default false;
end gpsDevice;

device implementation gpsDevice.i
        flows
                measurement := true in modes (active);
        modes
                acquisition : activation mode urgent in 10 min;
                active      : mode;
        transitions
                acquisition -[ within 10 sec to 10 min ]-> active;
end gpsDevice.i;
```

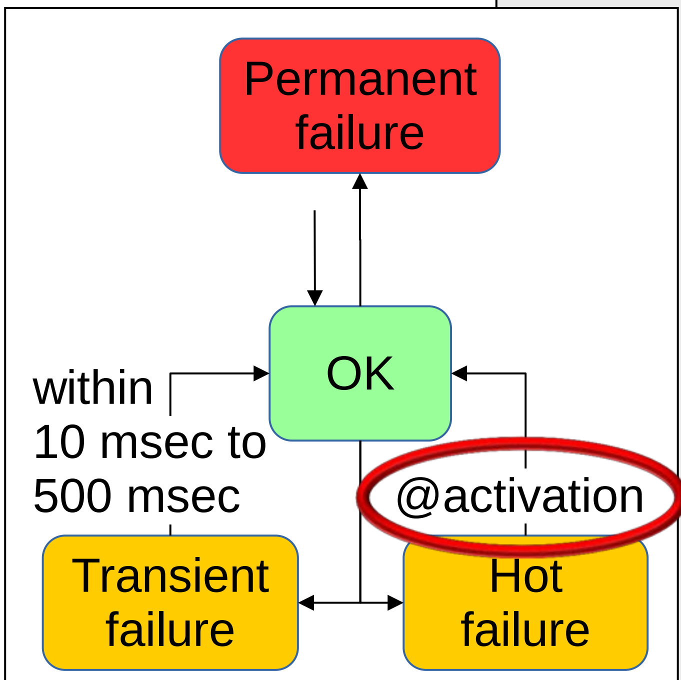
Electrical view




```
system gps
        features
                measurement     : out data port bool default false observable;
                powerOn         : in  event port;
                powerOff        : in  event port;
end gps;

system implementation gps.i
        subcomponents
                gps : device gpsDevice.i in modes (onMode);
        connections
                data port gps.measurement -> measurement in modes (onMode);
        modes
                offMode         : initial mode;
                booting         : mode urgent in 30 sec;
                onMode          : mode;
        transitions
                offMode         -[ powerOn                 ]-> booting;
                booting         -[ within 30 sec to 30 sec]-> onMode;
                booting         -[ powerOff                ]-> offMode;
                onMode          -[ powerOff                ]-> offMode;
                offMode         -[ powerOff                ]-> offMode;
                booting         -[ powerOn                 ]-> booting;
                onMode          -[ powerOn                 ]-> onMode;
end gps.i;
```

Measurement is provided only in onMode

Error view

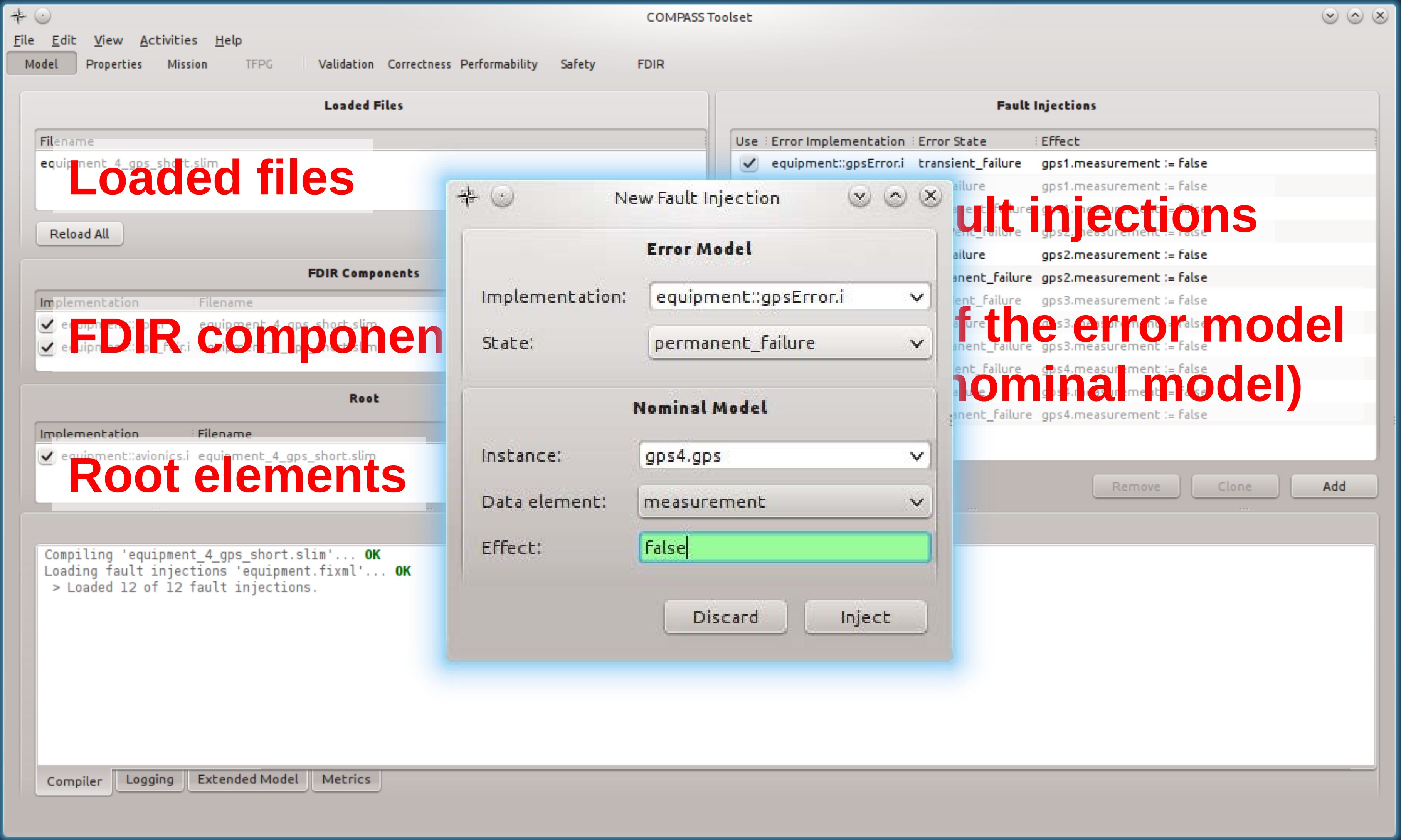


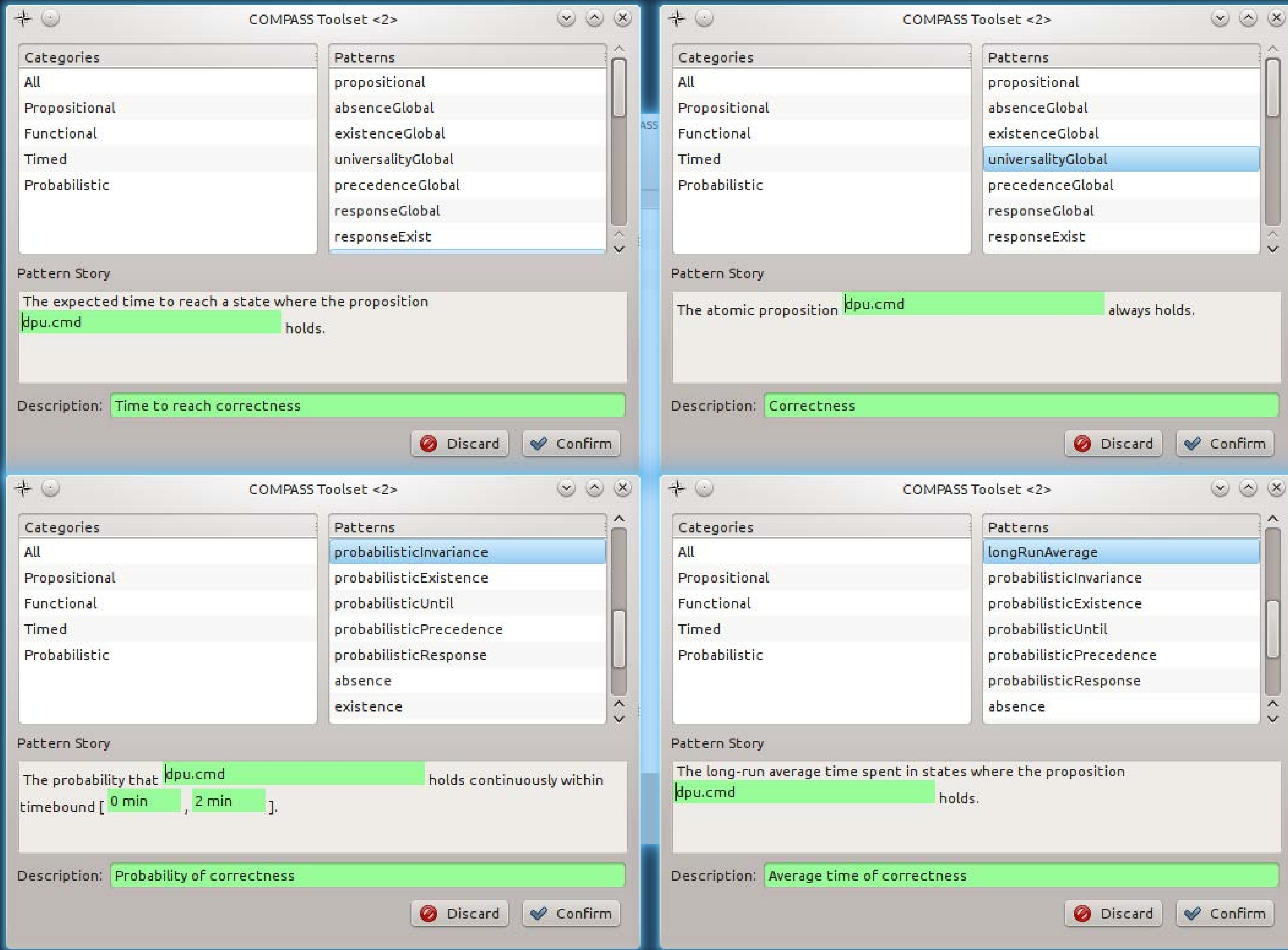
Probability of failure occurrence

Triggered on activation

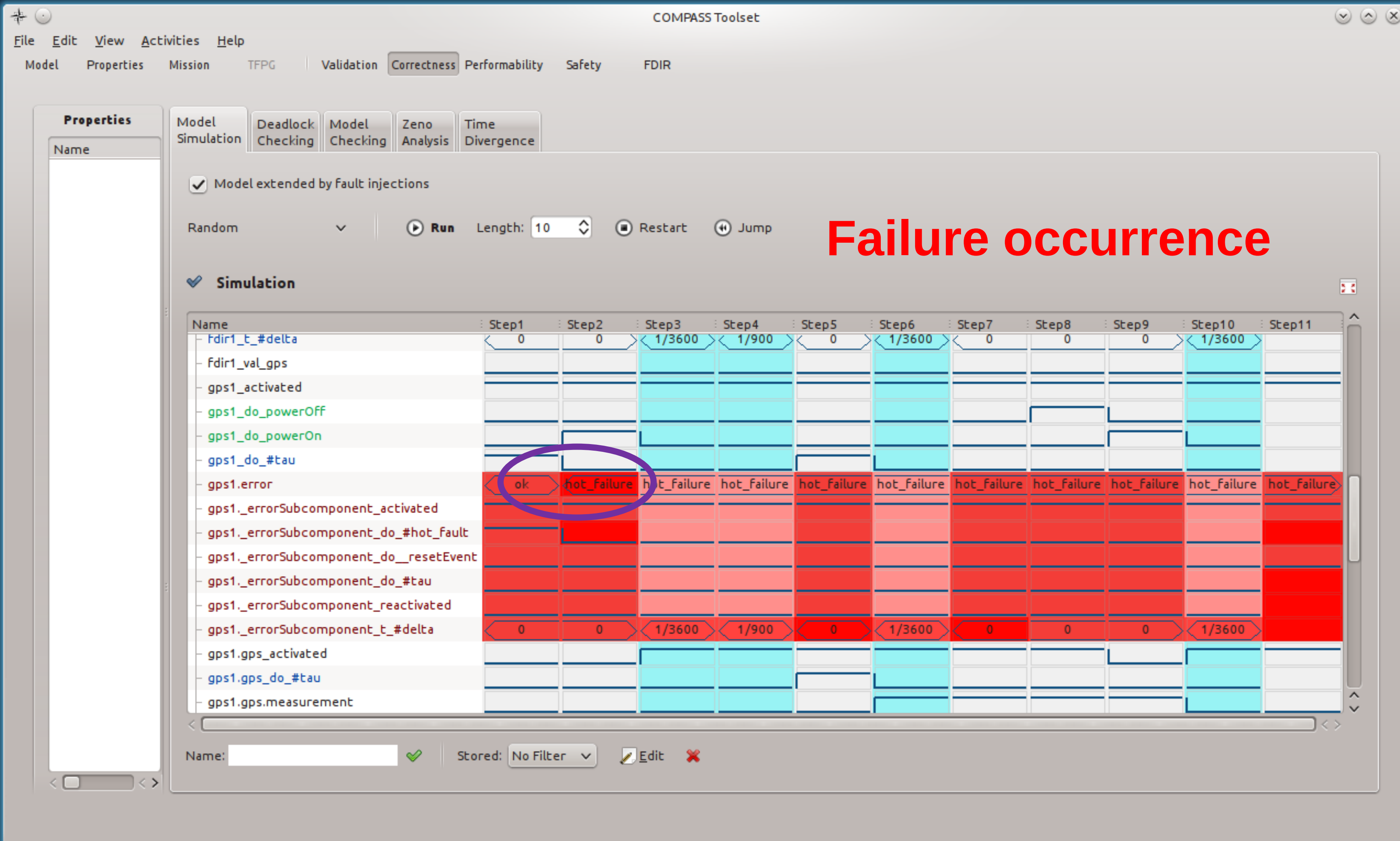
```
error model gpsError
end gpsError;

error model implementation gpsError.i
      events
            transient_fault      : error event occurrence poisson 0.01  per hour;
            hot_fault            : error event occurrence poisson 0.01  per day;
            permanent_fault      : error event occurrence poisson 0.001 per day;
            none                 : error event;
      states
            ok                   : initial state;
            transient_failure    : error state urgent in 500 msec;
            hot_failure          : error state;
            permanent_failure: error state;
      transitions
            ok                   -[ transient_fault        ]-> transient_failure;
            transient_failure    -[ none within 10 msec to 500 msec ]-> ok;
            ok                   -[ hot_fault              ]-> hot_failure;
            hot_failure          -[ @activation            ]-> ok;
            transient_failure    -[ @activation            ]-> ok;
            ok                   -[ permanent_fault        ]-> permanent_failure;
end gpsError.i;
```

IHM

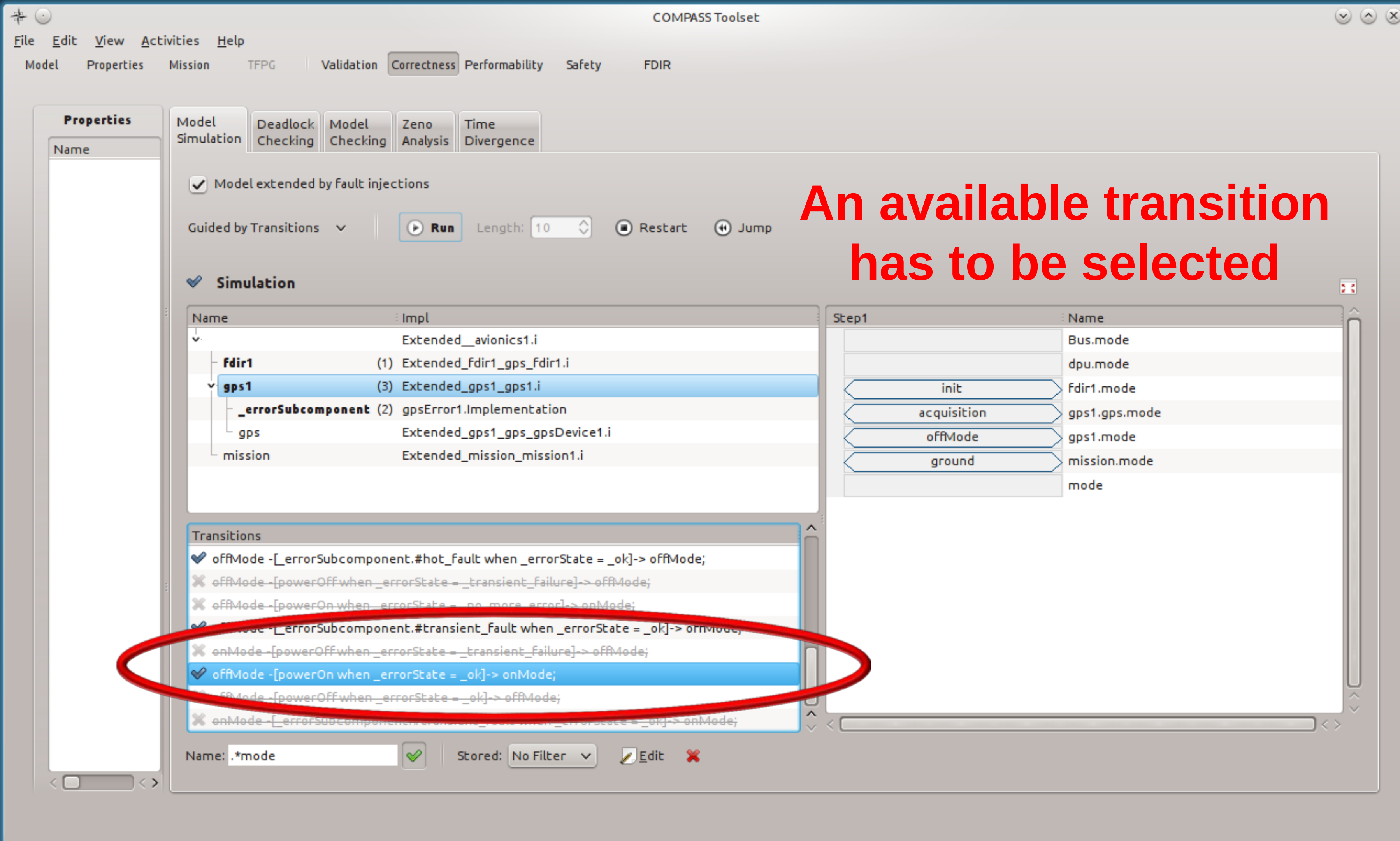

COMPASS Toolset
File Edit View Activities Help
Model Properties Mission TFPG Validation Correctness Performability Safety FDIR
Loaded Files
Filename
equipment_4_gps_short.slim
Loaded files
Reload All
FDIR Components
Implementation Filename
FDIR componen
Root
Implementation Filename
equipment::avionics.i equipment_4_gps_short.slim
Root elements
Fault Injections
Use Error Implementation Error State Effect
equipment::gpsError.i transient_failure gps1.measurement := false
ult injections
f the error model
nominal model)
gps2.measurement := false
gps3.measurement := false
gps4.measurement := false
Remove Clone Add
New Fault Injection
Error Model
Implementation: equipment::gpsError.i
State: permanent_failure
Nominal Model
Instance: gps4.gps
Data element: measurement
Effect: false
Discard Inject
Compiling 'equipment_4_gps_short.slim'... OK
Loading fault injections 'equipment.fixml'... OK
> Loaded 12 of 12 fault injections.
Compiler Logging Extended Model Metrics


COMPASS Toolset <2>
Categories
All
Propositional
Functional
Timed
Probabilistic
Patterns
propositional
absenceGlobal
existenceGlobal
universalityGlobal
precedenceGlobal
responseGlobal
responseExist
Pattern Story
The expected time to reach a state where the proposition
dpu.cmd
holds.
Description:
Time to reach correctness
Discard
Confirm
COMPASS Toolset <2>
Categories
All
Propositional
Functional
Timed
Probabilistic
Patterns
propositional
absenceGlobal
existenceGlobal
universalityGlobal
precedenceGlobal
responseGlobal
responseExist
Pattern Story
The atomic proposition
dpu.cmd
always holds.
Description:
Correctness
Discard
Confirm
COMPASS Toolset <2>
Categories
All
Propositional
Functional
Timed
Probabilistic
Patterns
probabilisticInvariance
probabilisticExistence
probabilisticUntil
probabilisticPrecedence
probabilisticResponse
absence
existence
Pattern Story
The probability that
dpu.cmd
holds continuously within
timebound [
0 min
,
2 min
].
Description:
Probability of correctness
Discard
Confirm
COMPASS Toolset <2>
Categories
All
Propositional
Functional
Timed
Probabilistic
Patterns
longRunAverage
probabilisticInvariance
probabilisticExistence
probabilisticUntil
probabilisticPrecedence
probabilisticResponse
absence
Pattern Story
The long-run average time spent in states where the proposition
dpu.cmd
holds.
Description:
Average time of correctness
Discard
Confirm

Random simulation

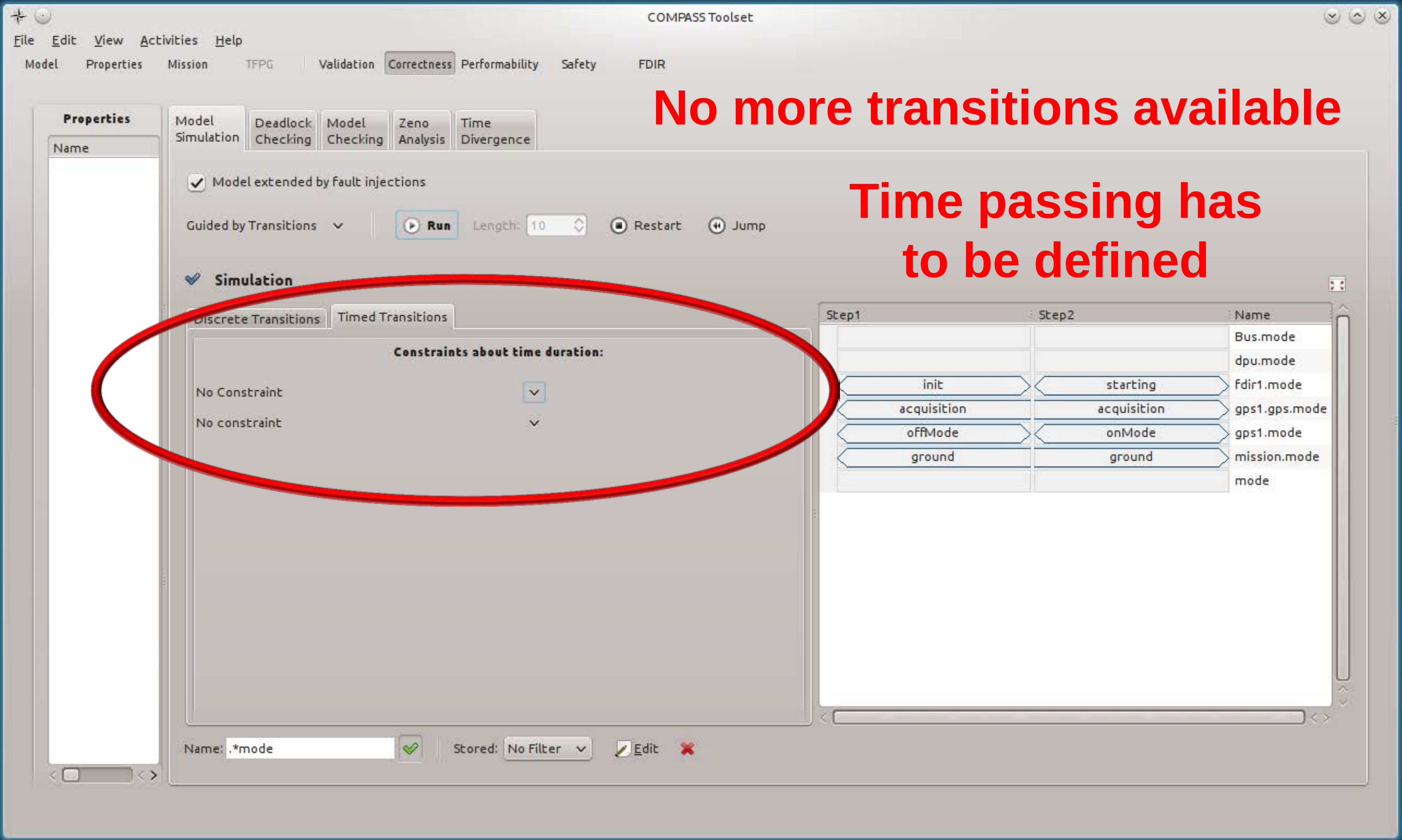


Simulation guided by transitions

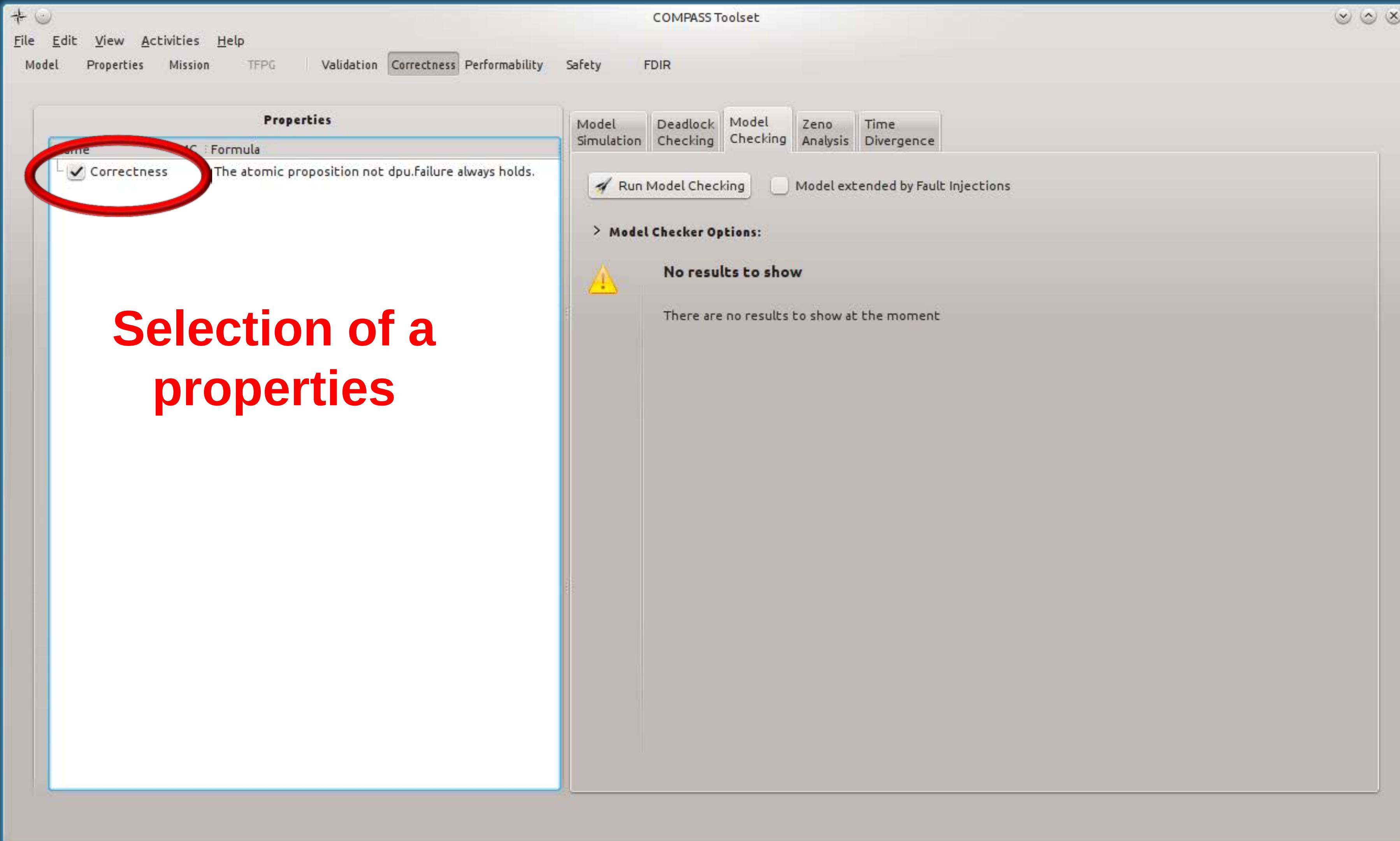


An available transition has to be selected

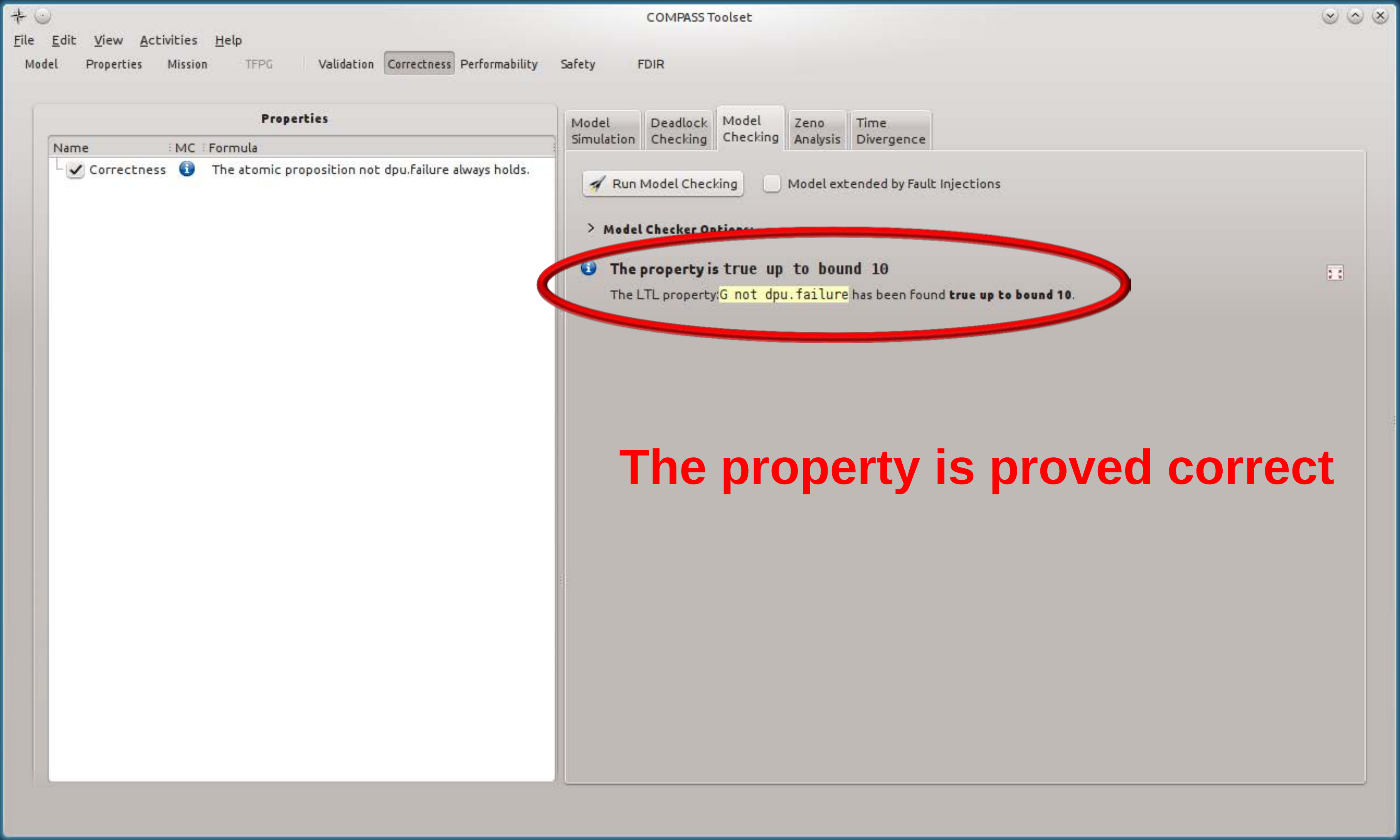
Simulation guided by transitions



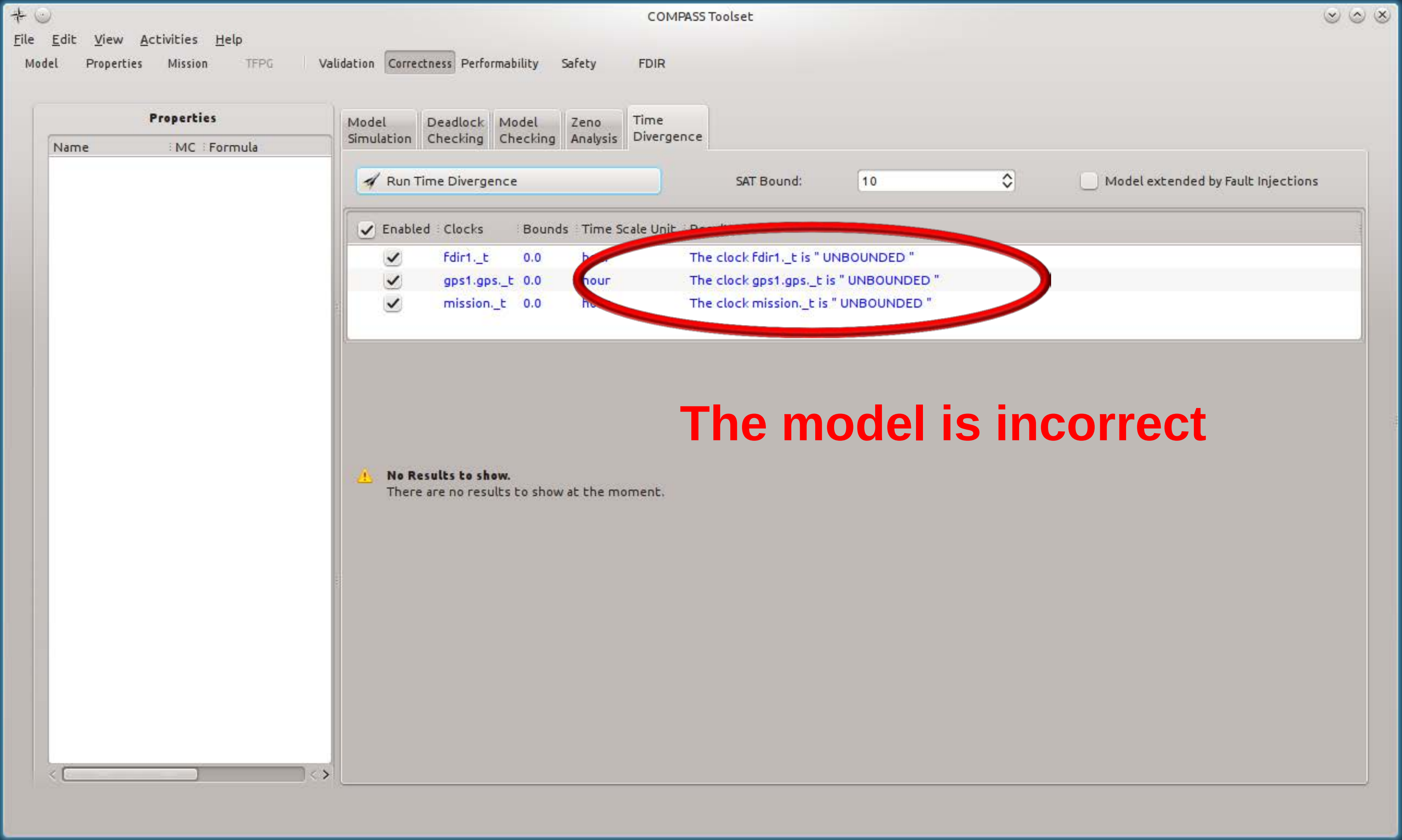
Model checking



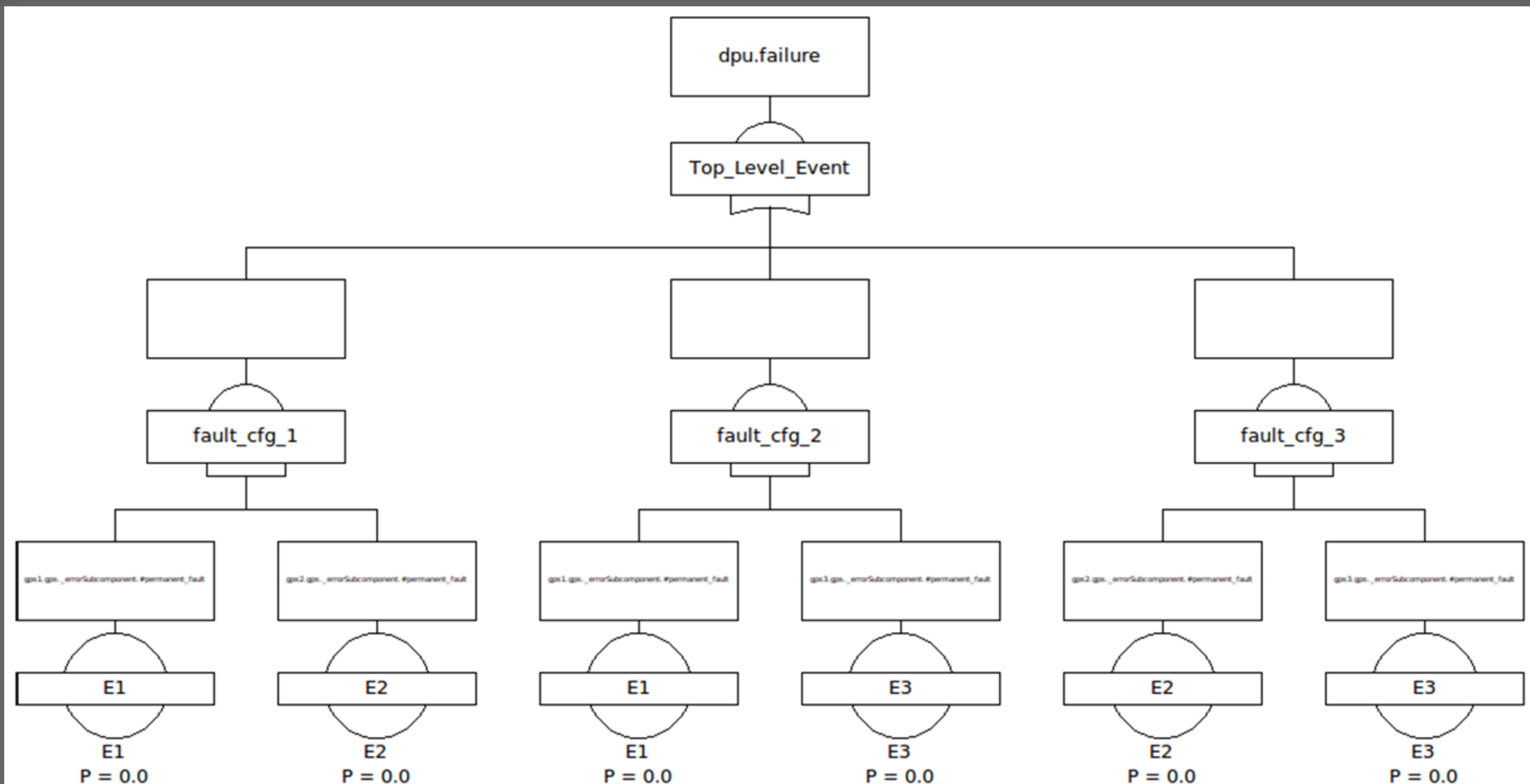
Model checking



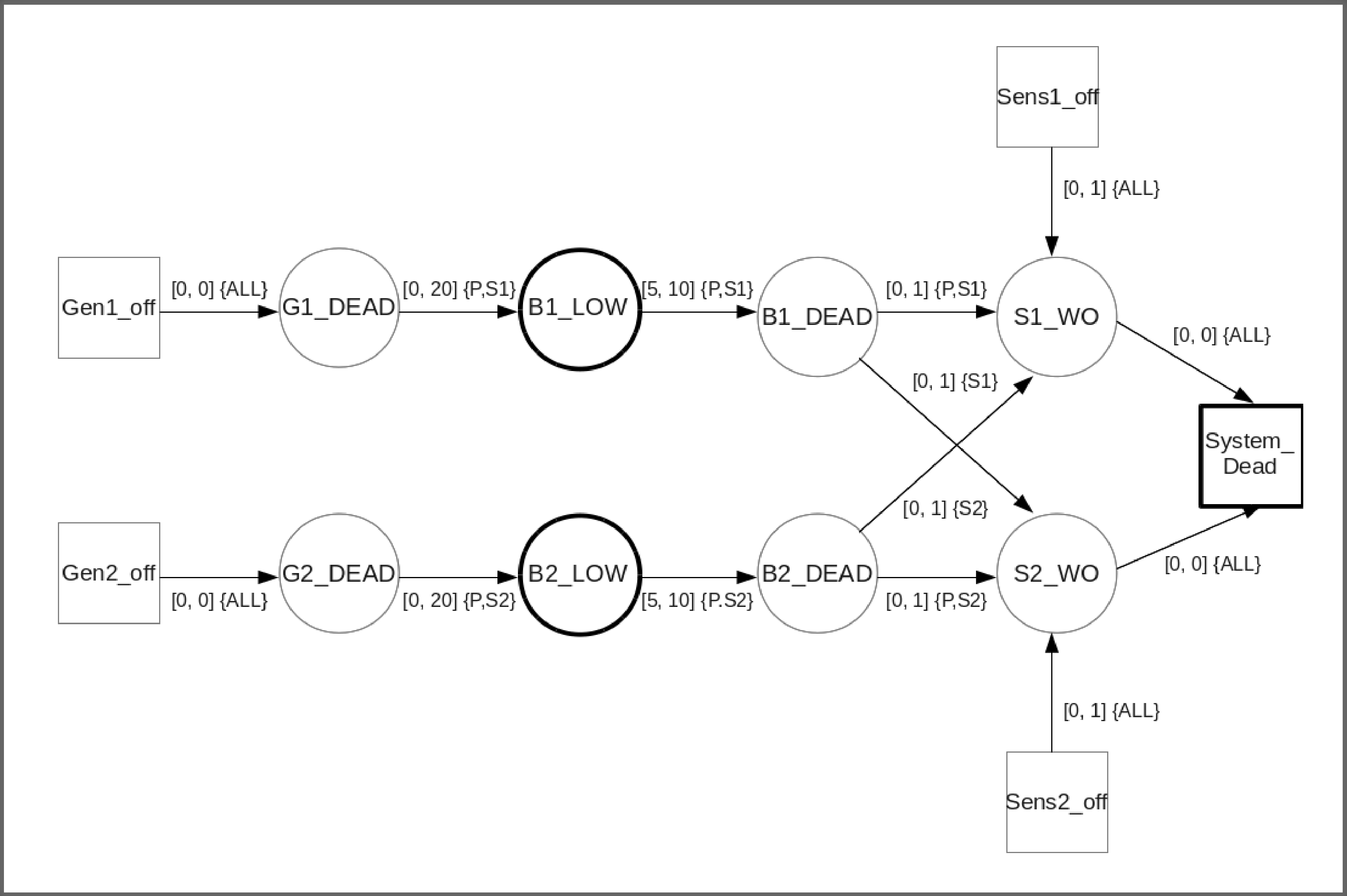
Time divergence analysis



Generation of Fault Trees



Generation Timed Failure Propagation Graphs



Agenda

- Introduction – Objectives of the HASDEL project
- The HASDEL approach
- Use cases
  - Equipment reintegration
  - ATV data handling system architecture
- Demonstration
- Conclusion

The HASDEL Toolset

Distribution

- Freely available for ESA member states
- Released under variant of GPL (GNU Public License) - restriction to ESA member states + some back-ends released under FBK's Additional Components License
- Needs ESA approval for export outside ESA member states

http://compass.informatik.rwth-aachen.de

Conclusion

- Some improvements still needed for deployment
  - Semantics of some language constructs
  - Link with SysML tool
  - Improve performances on the analysis tools
  - …
- But HASDEL could bring great benefits
  - It allows early RAMS analyses before the actual development
  - RAMS analyses are automated

http://compass.informatik.rwth-aachen.de

HASDEL

Hardware Software Dependability for Launchers

Thank you for your attention

Any question ?

David Lesens	david lesens@astrium.eads.net
Joost-Pieter Katoen	katoen@informatik.rwth-aachen.de
Alessandro Cimatti	cimatti@fbk.eu

AIRBUS DEFENCE & SPACE — FONDAZIONE BRUNO KESSLER — RWTH AACHEN UNIVERSITY