

# HASDEL

Hardware Software  
Dependability  
for Launchers

Final Presentation Days

Tuesday, 09 December 2014

Presented by David LESENS & Marco BOZZANO



# Agenda



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

**HASDEL**

Hardware Software Dependability for Launchers



**AIRBUS**  
DEFENCE & SPACE



FONDAZIONE  
BRUNO KESSLER

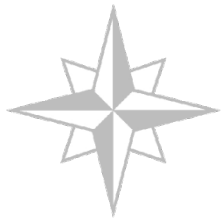
**RWTHAACHEN**  
UNIVERSITY

# 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)

---

**HASDEL**

Hardware Software Dependability for Launchers

09/12/2014 p3



**AIRBUS**  
DEFENCE & SPACE

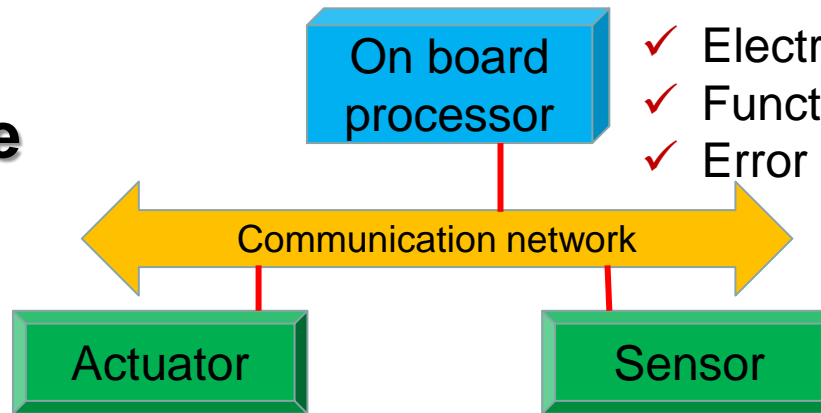


FONDAZIONE  
BRUNO KESSLER

**RWTHAACHEN**  
UNIVERSITY

# Avionics Architecture

- ✓ Electrical model
- ✓ Functional model
- ✓ Error model



- ✓ Electrical model
- ✓ Functional model
- ✓ Error model

- ✓ Electrical model
- ✓ Functional model
- ✓ Error model

# Avionics Architecture

Fault injections

On board processor

- ✓ Electrical model
- ✓ Functional model
- ✓ Error model

Communication network

Actuator

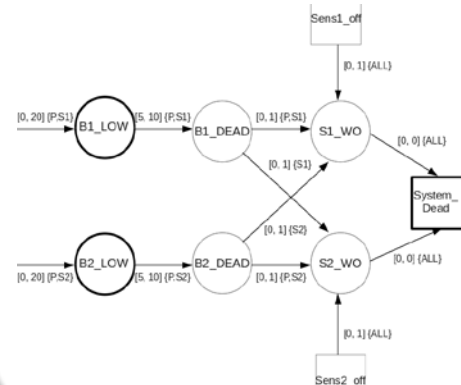
Sensor

## Analysis

Simulation  
Formal proof

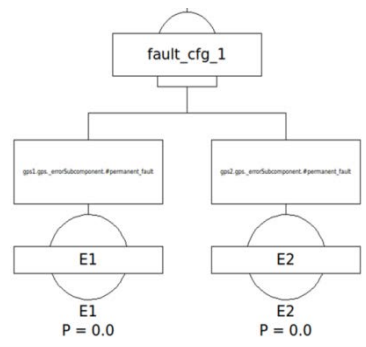
| Name           | Step1 | Step2 | Step3 | Step4 | Step5 | Step6 | Step7 | Step8 | Step9 | Step10 | Step11 | Step12 | Step13 | Step14 | Step15 | Step16 |
|----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|--------|
| Bus_actuated   |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#low   |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#high  |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#int   |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#off   |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#on    |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#stop  |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#start |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#idle  |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#run   |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#halt  |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#test  |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#stop  |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#start |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#idle  |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#run   |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#halt  |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#test  |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#stop  |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#start |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#idle  |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#run   |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#halt  |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |
| Bus_err_#test  |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |        |

## Timed Failure Propagation Graph



## Safety

Fault Tree Analysis  
Failure Mode and Effect Analysis



# Agenda



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

**HASDEL**

Hardware Software Dependability for Launchers

09/12/2014 p6



**AIRBUS**  
DEFENCE & SPACE

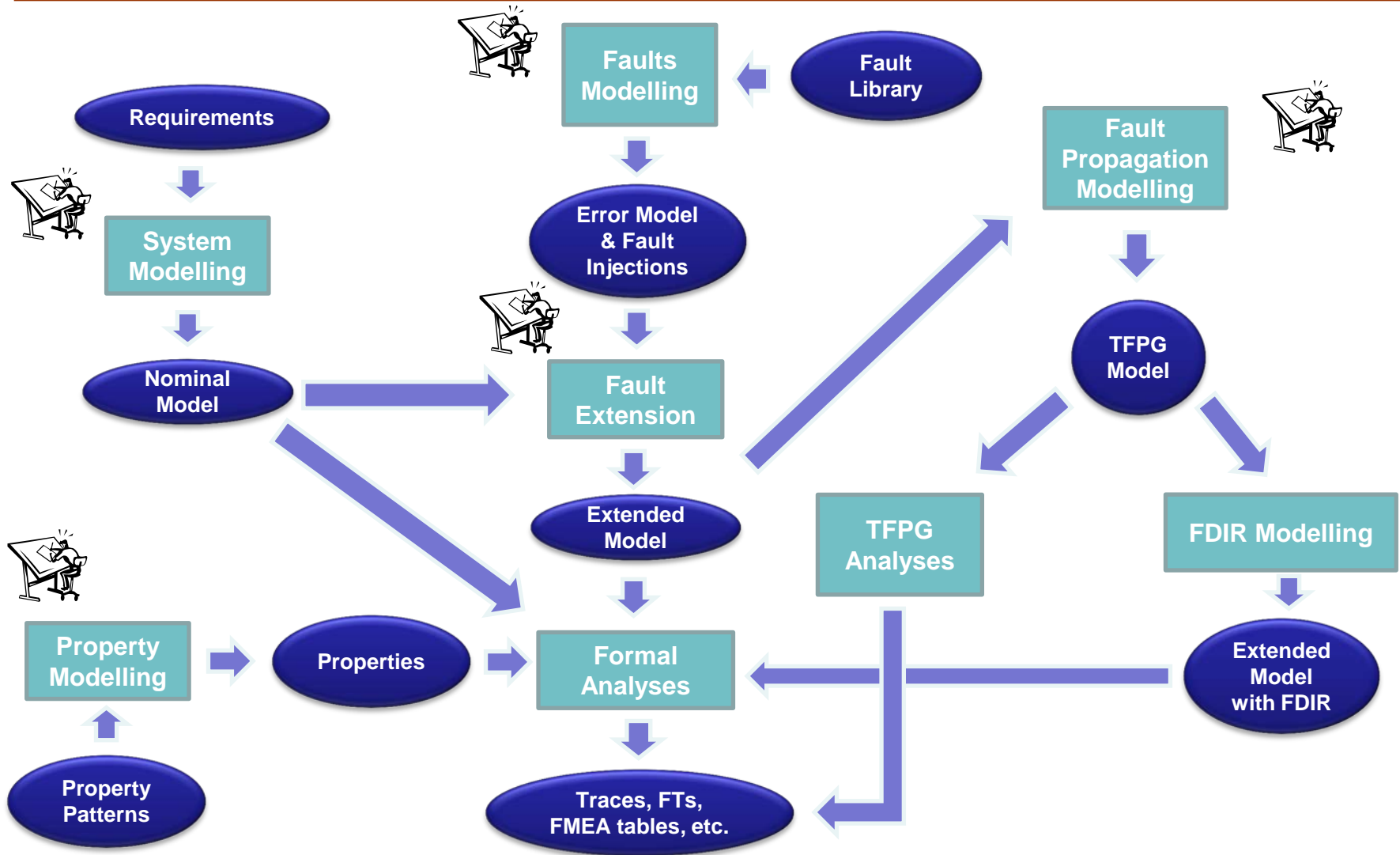


FONDAZIONE  
BRUNO KESSLER

**RWTHAACHEN**  
UNIVERSITY



# The HASDEL Approach: Flow



**HASDEL**

Hardware Software Dependability for Launchers



**AIRBUS**  
DEFENCE & SPACE



**RWTH AACHEN**  
UNIVERSITY

# 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



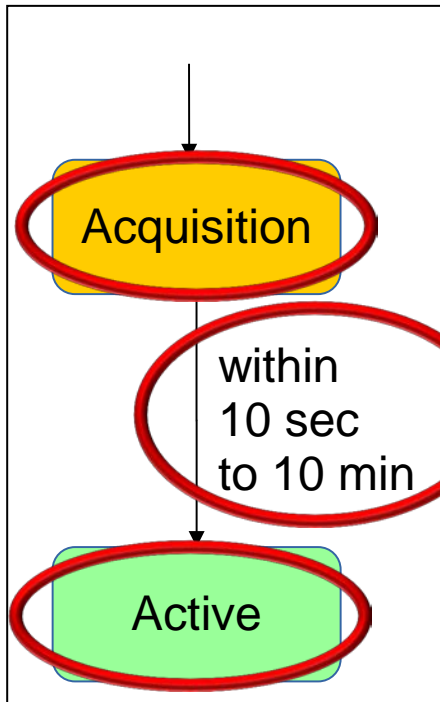
**HASDEL**

Hardware Software Dependability for Launchers



# 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



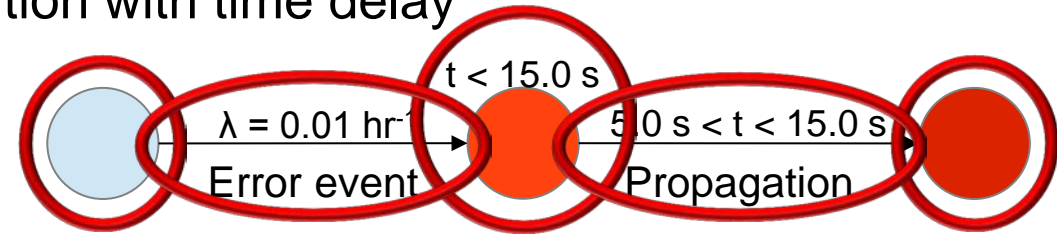
```
device gpsDevice
  features
    measurement : out data port bool default false;
  end gpsDevice;

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

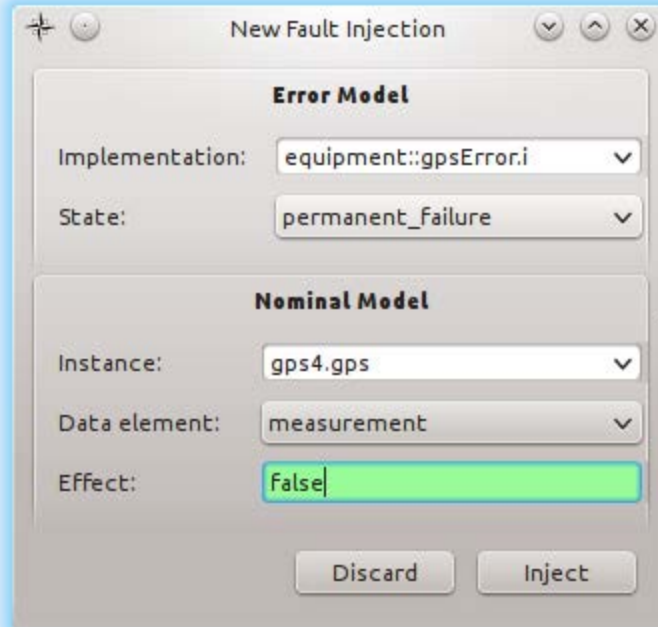


```
error model SubError
features
  err_prop : out error propagation;
end SubError;

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 patterns

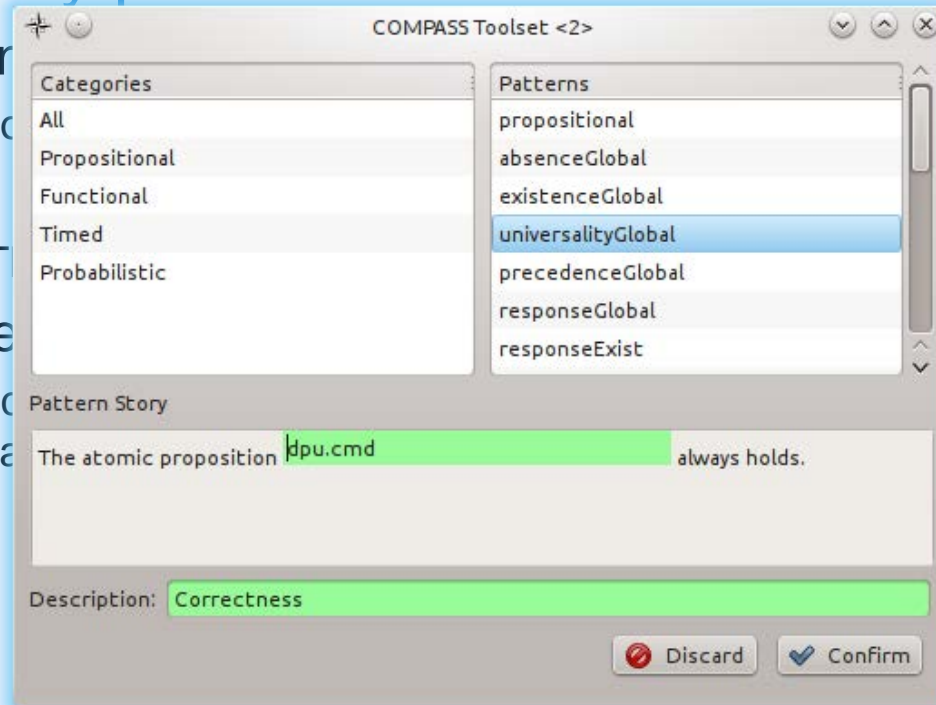
- E.g., absenceGlobal

- Timed patterns

- E.g. absence: “T

- Probabilistic patterns

- E.g., probabilistic  
Time2 with proba



me units”

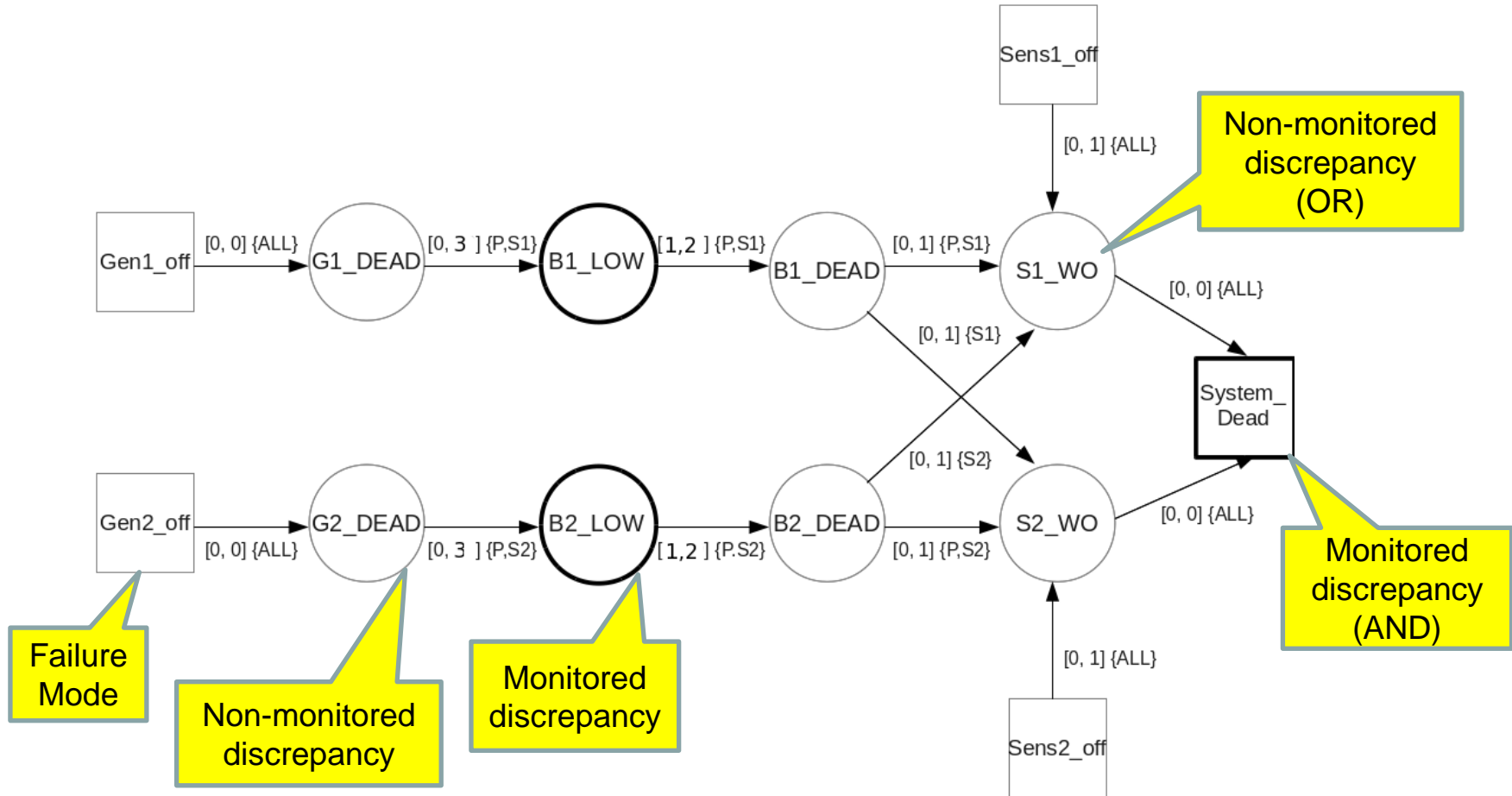
between Time1 and

# 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 TFBG





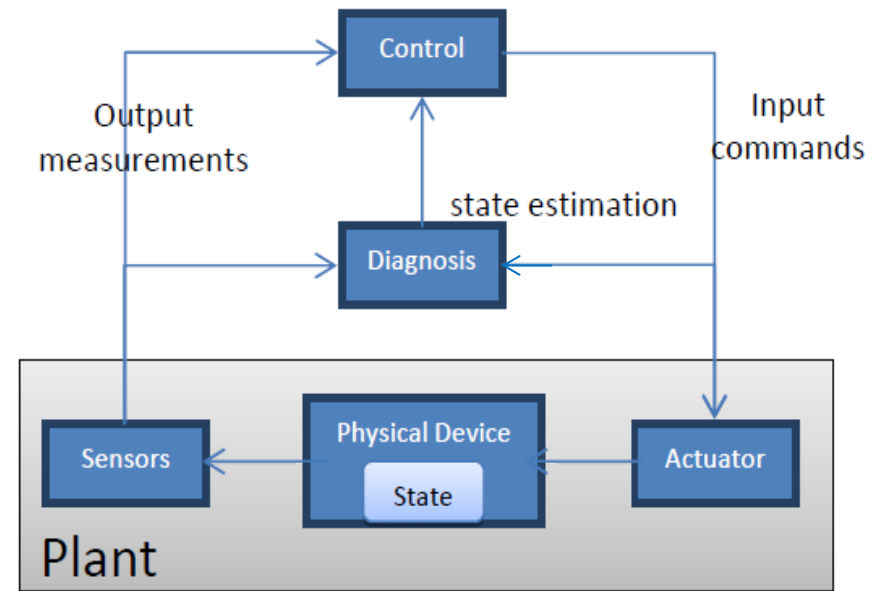
# Fault Detection, Isolation and Recovery (FDIR)

## ■ 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

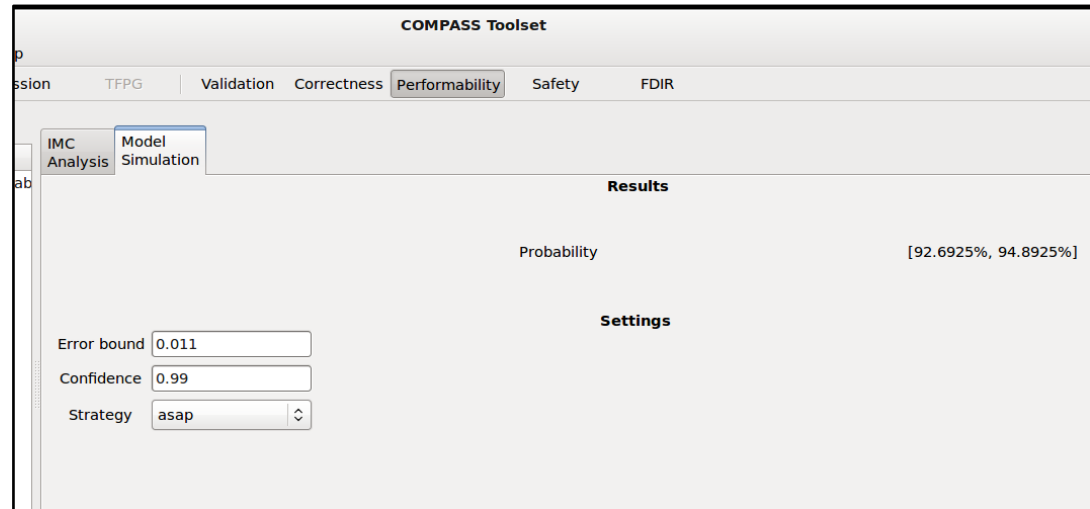
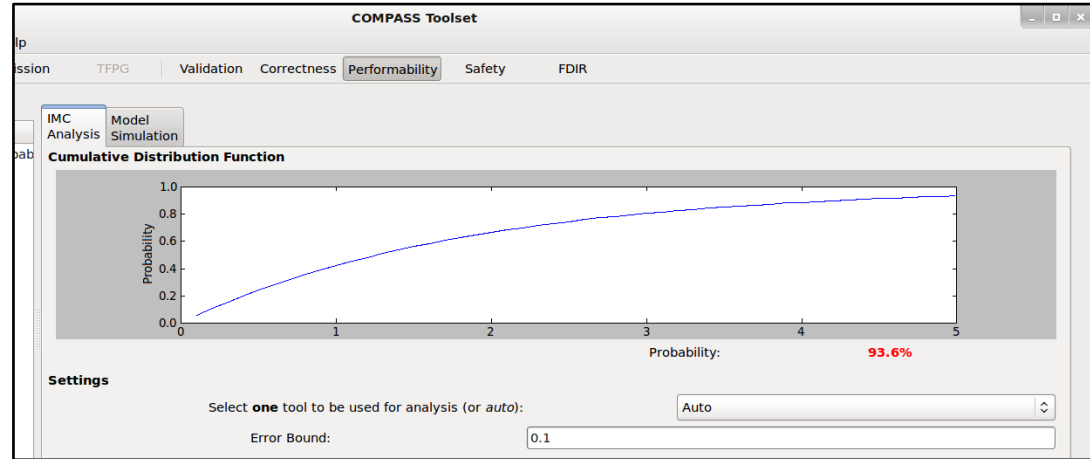
- Investigate model reliability

## ■ Numerical analysis

- Based on Markov Chain model checking

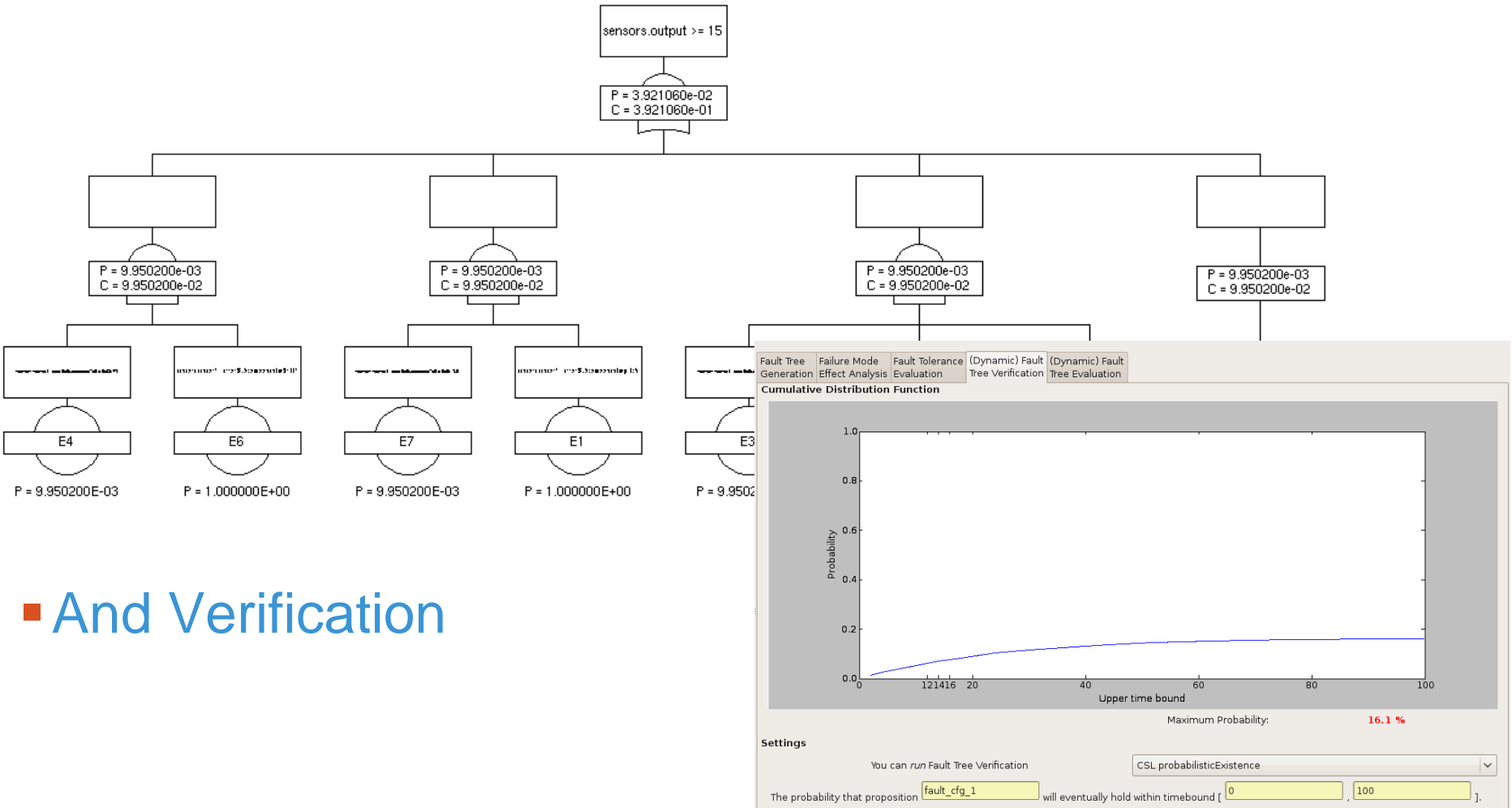
## ■ Statistical analysis

- Based on Monte Carlo method



# Fault Tree analysis

- Supports Fault Tree generation and evaluation



- And Verification

**HASDEL**

Hardware Software Dependability for Launchers



**AIRBUS**  
DEFENCE & SPACE



**RWTHAACHEN**  
UNIVERSITY

# Agenda



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

**HASDEL**

Hardware Software Dependability for Launchers

09/12/2014 p19



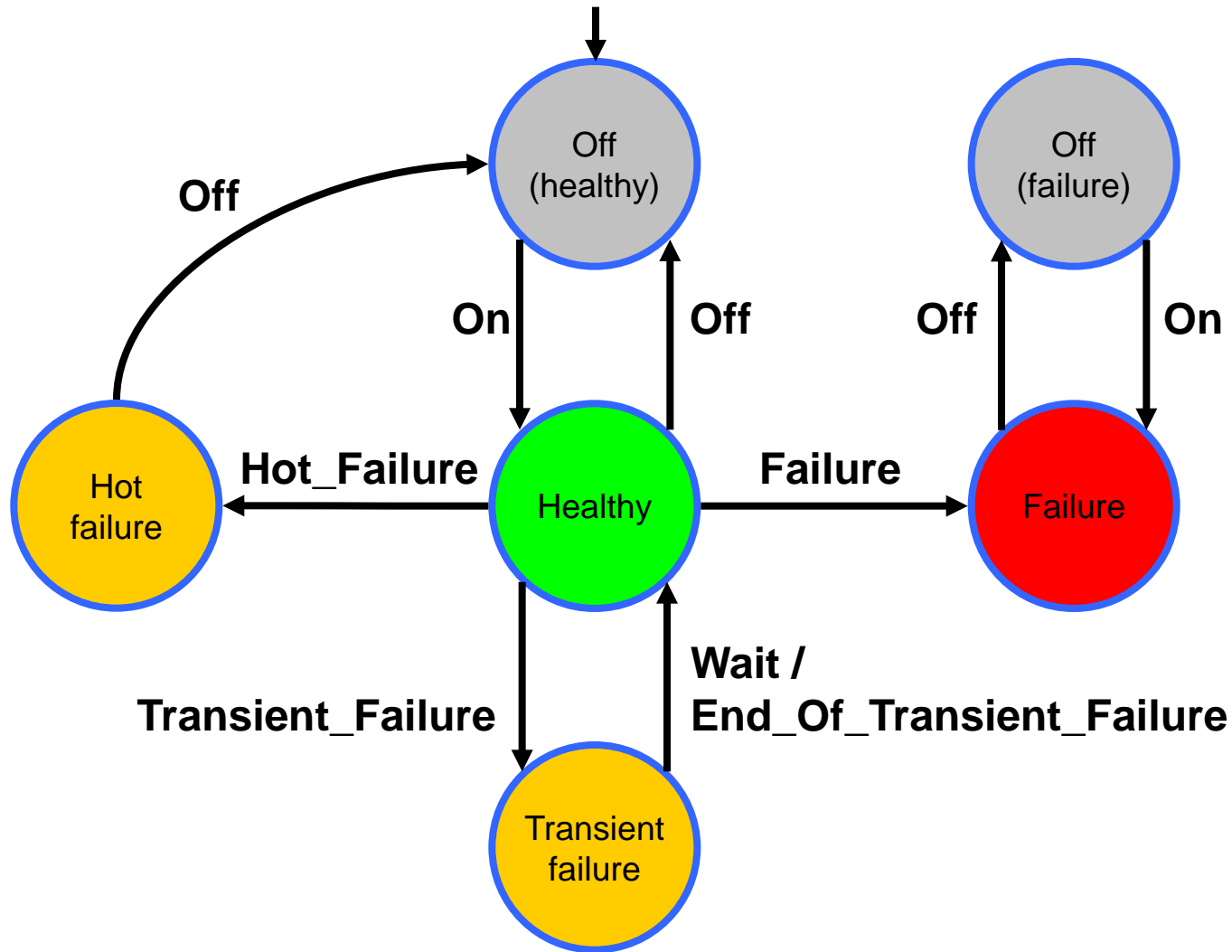
**AIRBUS**  
DEFENCE & SPACE



FONDAZIONE  
BRUNO KESSLER

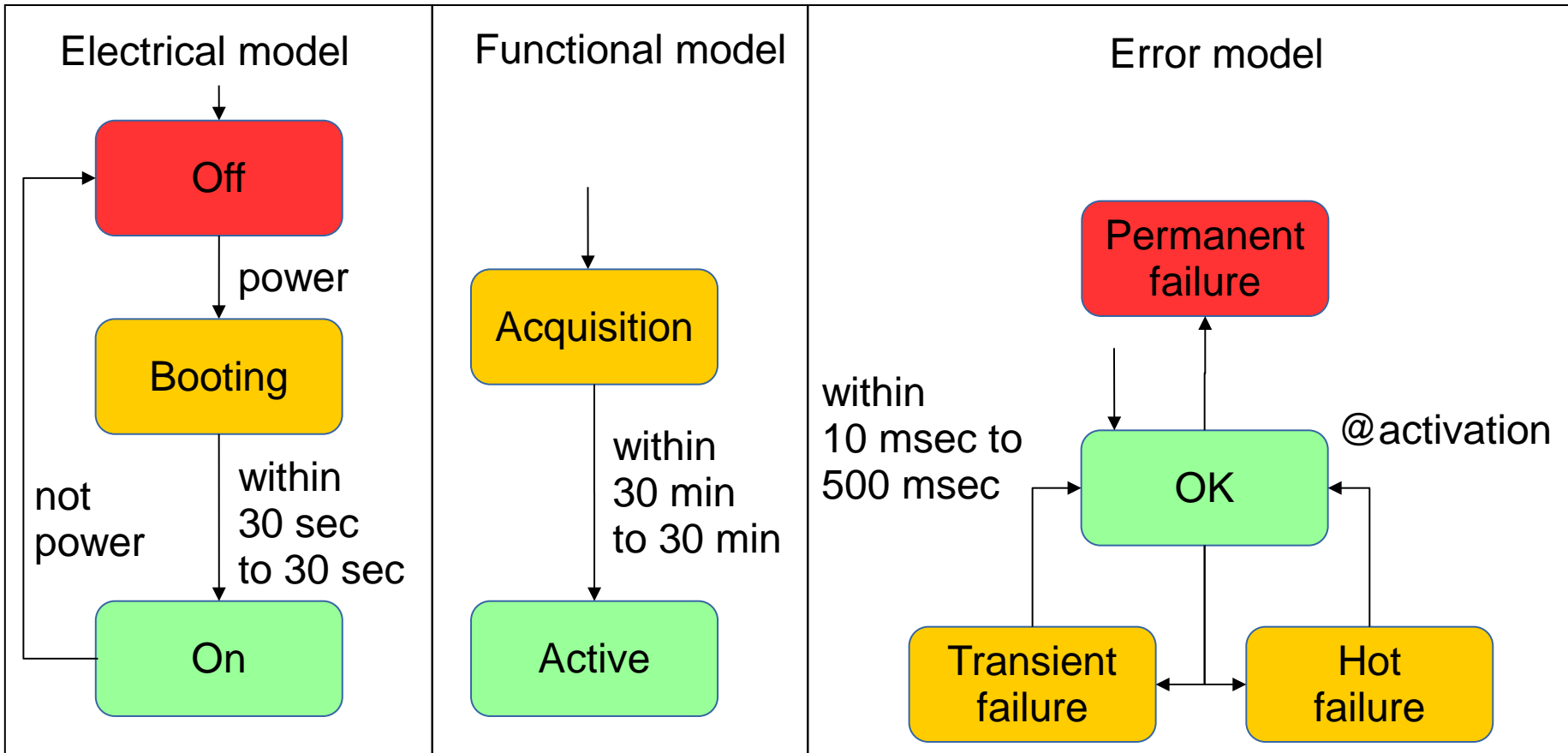
**RWTHAACHEN**  
UNIVERSITY

# Equipment reintegration

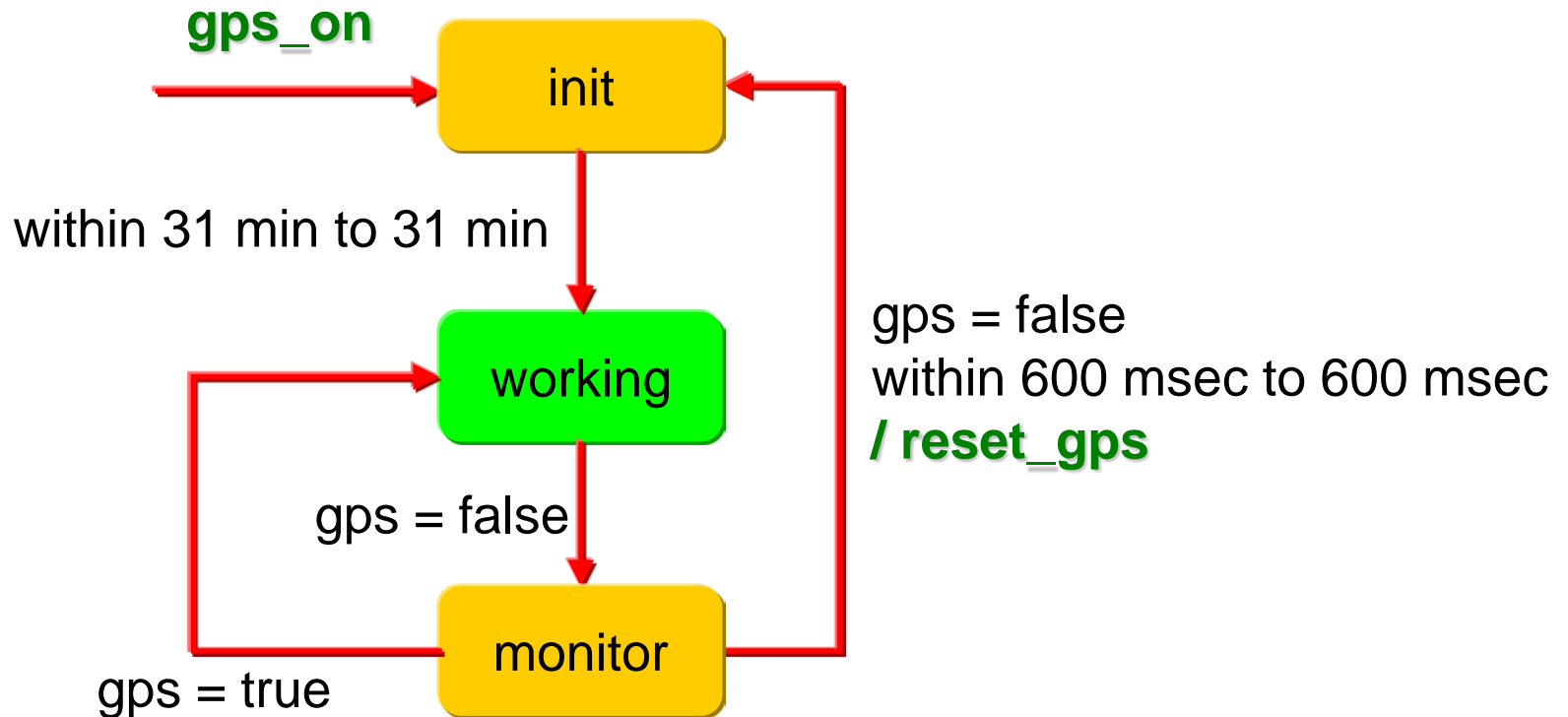




# Equipment reintegration modelling principle



# Simplified FDIR



# Agenda



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

**HASDEL**

Hardware Software Dependability for Launchers

09/12/2014 p23



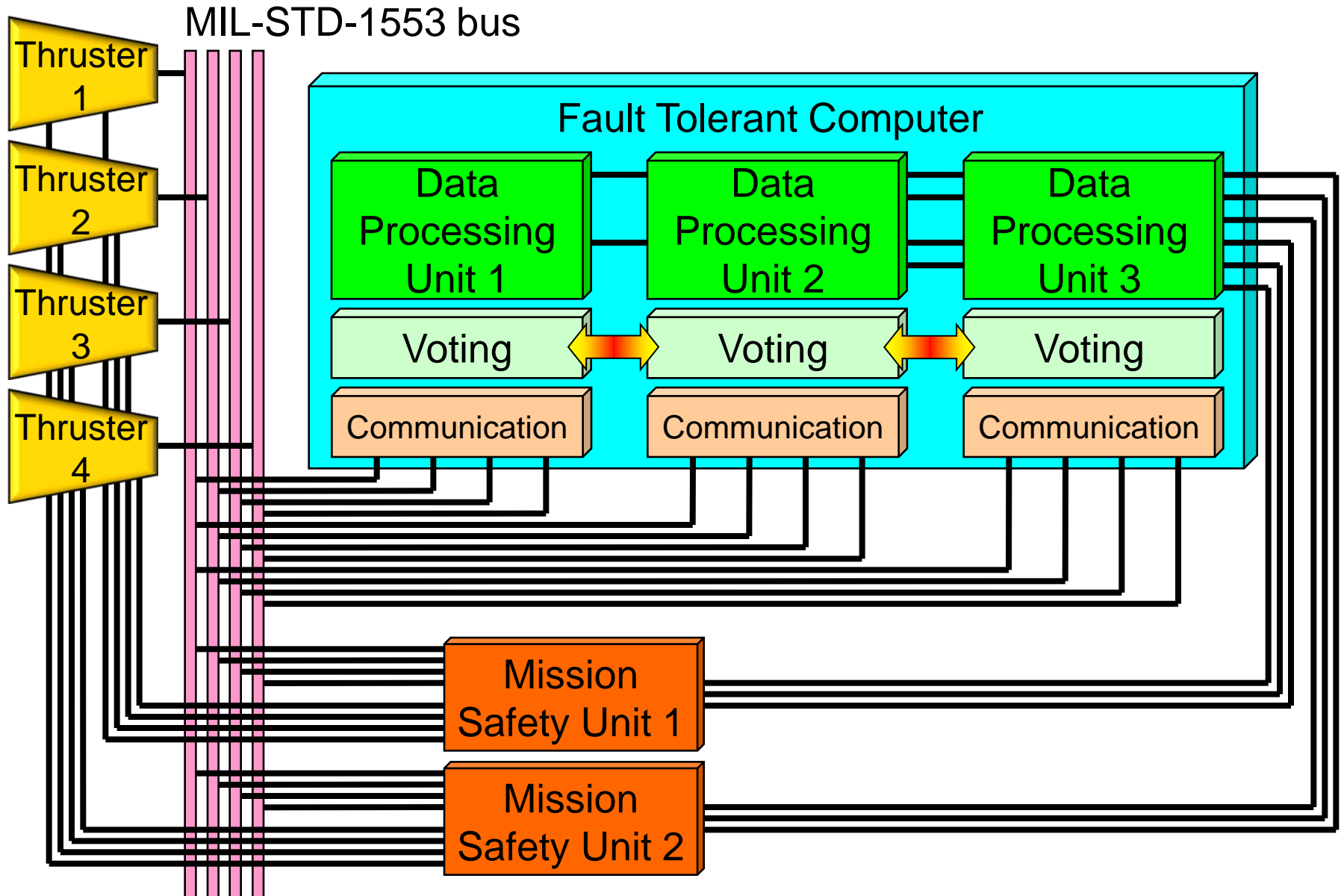
**AIRBUS**  
DEFENCE & SPACE



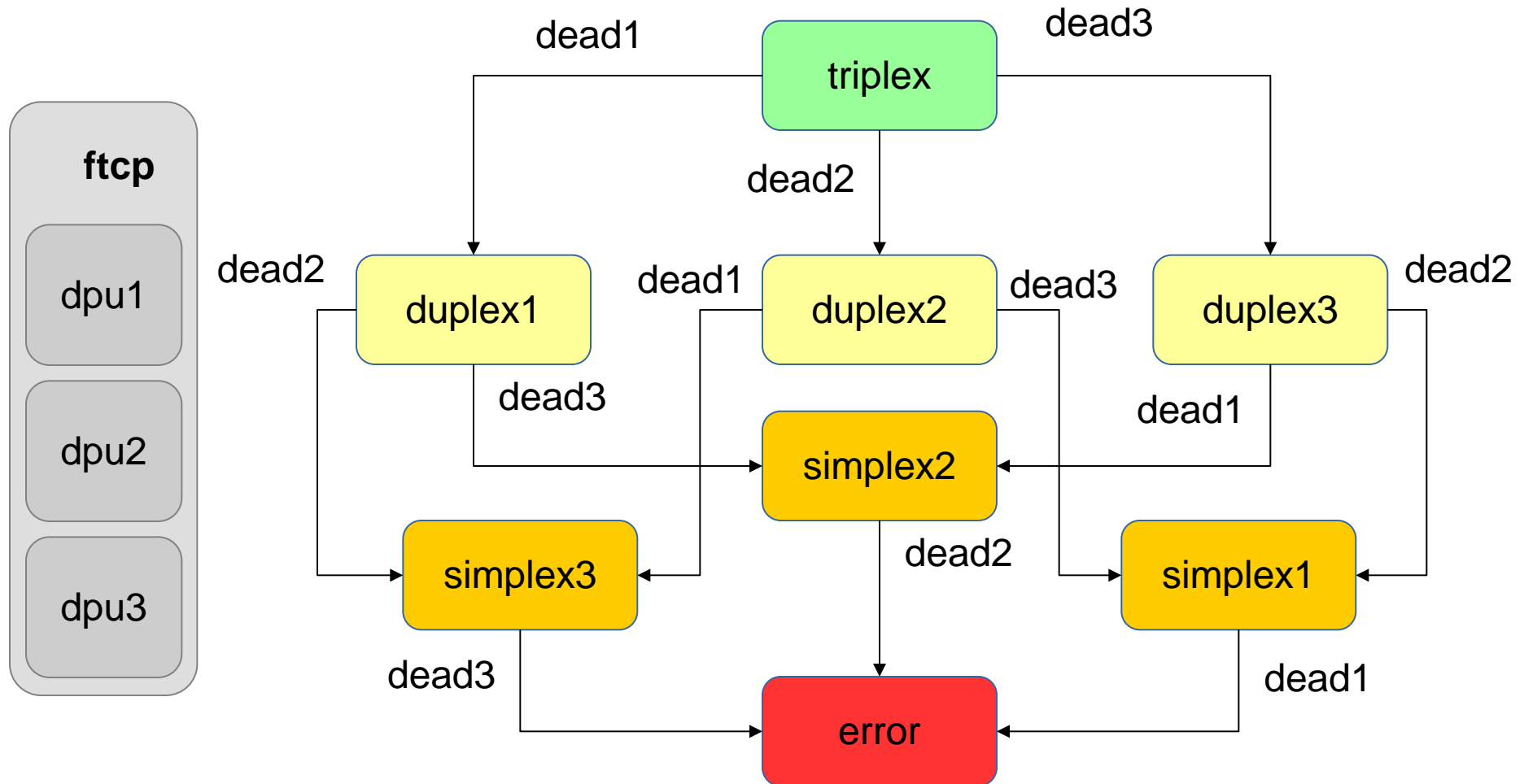
FONDAZIONE  
BRUNO KESSLER

**RWTHAACHEN**  
UNIVERSITY

# ATV data handling system architecture



# Fault Tolerant Computer model



# Properties

| Property type                  | Property description  |
|--------------------------------|---|
| <b>expectedTime</b>            | “The expected time to reach a state where the proposition <i>not failure</i> holds.”                          |
| <b>longRunAverage</b>          | “The long-run average time spent in states where the proposition <i>not failure</i> holds.”                   |
| <b>probabilisticInvariance</b> | “The probability that <i>not failure</i> holds continuously within timebound [ <i>0 min</i> , <i>2 min</i> ]” |



# Agenda



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

**HASDEL**

Hardware Software Dependability for Launchers

09/12/2014 p27



**AIRBUS**  
DEFENCE & SPACE

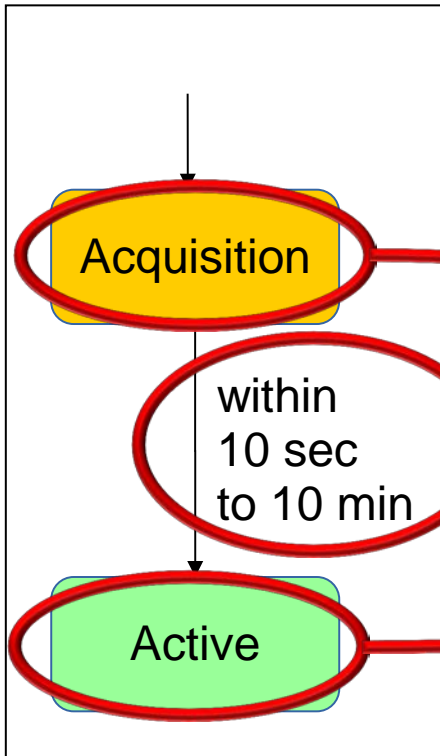


FONDAZIONE  
BRUNO KESSLER

**RWTHAACHEN**  
UNIVERSITY

# Functional view

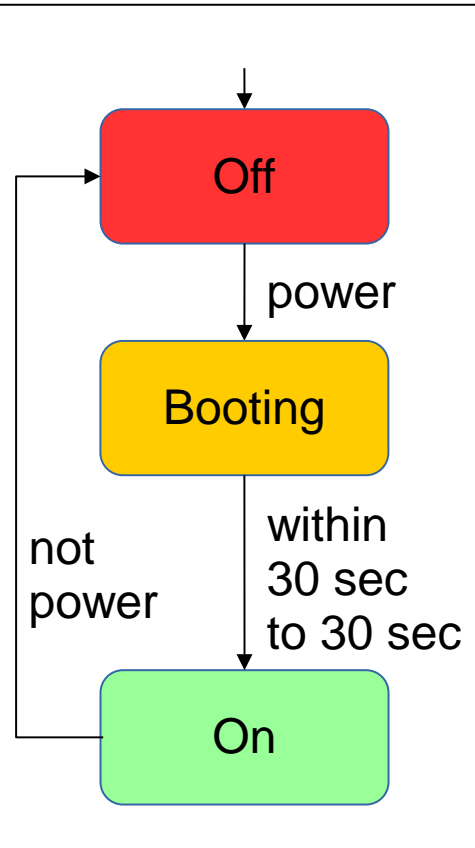
## The sensor provides correct measurement



```
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

```
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;
```

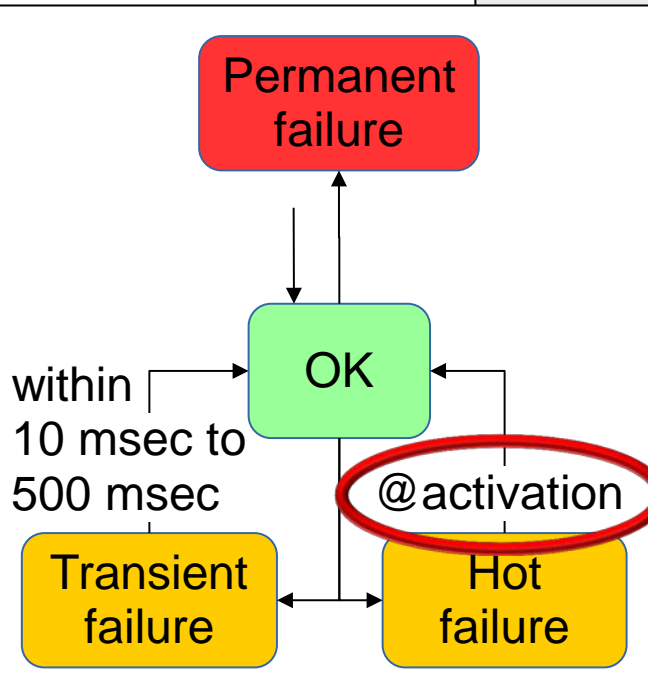
transitions

```
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;
```

## Probability of failure occurrence

## Triggered on activation



The screenshot displays the COMPASS Toolset interface with several panels and a dialog box. The 'Loaded Files' panel shows a table with columns for 'Filename' and 'Implementation', listing 'equipment\_4\_gps\_short.slim'. The 'FDIR Components' panel shows a table with columns for 'Implementation' and 'Filename', listing 'equipment::avionics.i' and 'equipment\_4\_gps\_short.slim'. The 'Root' panel shows a table with columns for 'Implementation' and 'Filename', listing 'equipment::avionics.i' and 'equipment\_4\_gps\_short.slim'. The 'Fault Injections' panel shows a table with columns for 'Use', 'Error Implementation', 'Error State', and 'Effect', listing 'equipment::gpsError.i', 'transient\_failure', and 'gps1.measurement := false'. The 'New Fault Injection' dialog box is open, showing the 'Error Model' section with 'Implementation: equipment::gpsError.i' and 'State: permanent\_failure', and the 'Nominal Model' section with 'Instance: gps4.gps', 'Data element: measurement', and 'Effect: false'. The dialog box has 'Discard' and 'Inject' buttons. The bottom of the interface shows a console window with the following text: 'Compiling 'equipment\_4\_gps\_short.slim'... OK', 'Loading fault injections 'equipment.fixml'... OK', and '> Loaded 12 of 12 fault injections.' The bottom of the interface also shows buttons for 'Compiler', 'Logging', 'Extended Model', and 'Metrics'.

**Loaded files**

**FDIR components**

**Root elements**

**Fault injections of the error model (nominal model)**

COMPASS Toolset <2>

| Categories    | Patterns           |
|---------------|--------------------|
| All           | propositional      |
| Propositional | absenceGlobal      |
| Functional    | existenceGlobal    |
| Timed         | universalityGlobal |
| Probabilistic | precedenceGlobal   |
|               | responseGlobal     |
|               | responseExist      |

Pattern Story

The expected time to reach a state where the proposition `dpu.cmd` holds.

Description: Time to reach correctness

COMPASS Toolset <2>

| Categories    | Patterns           |
|---------------|--------------------|
| All           | propositional      |
| Propositional | absenceGlobal      |
| Functional    | existenceGlobal    |
| Timed         | universalityGlobal |
| Probabilistic | precedenceGlobal   |
|               | responseGlobal     |
|               | responseExist      |

Pattern Story

The atomic proposition `dpu.cmd` always holds.

Description: Correctness

COMPASS Toolset <2>

| Categories    | Patterns                |
|---------------|-------------------------|
| All           | probabilisticInvariance |
| Propositional | probabilisticExistence  |
| Functional    | probabilisticUntil      |
| Timed         | probabilisticPrecedence |
| Probabilistic | probabilisticResponse   |
|               | absence                 |
|               | existence               |

Pattern Story

The probability that `dpu.cmd` holds continuously within timebound [ 0 min , 2 min ].

Description: Probability of correctness

COMPASS Toolset <2>

| Categories    | Patterns                |
|---------------|-------------------------|
| All           | longRunAverage          |
| Propositional | probabilisticInvariance |
| Functional    | probabilisticExistence  |
| Timed         | probabilisticUntil      |
| Probabilistic | probabilisticPrecedence |
|               | probabilisticResponse   |
|               | absence                 |

Pattern Story

The long-run average time spent in states where the proposition `dpu.cmd` holds.

Description: Average time of correctness

# Random simulation

COMPASS Toolset

File Edit View Activities Help

Model Properties Mission TFPG Validation Correctness Performability Safety FDIR

Properties

Name

Model Simulation Deadlock Checking Model Checking Zeno Analysis Time Divergence

Model extended by fault injections

Random  Run Length: 10  Restart  Jump

## Failure occurrence

Simulation

| Name                                 | Step1 | Step2       | Step3       | Step4       | Step5       | Step6       | Step7       | Step8       | Step9       | Step10      | Step11      |
|--------------------------------------|-------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Fdir1_t_#delta                       | 0     | 0           | 1/3600      | 1/900       | 0           | 1/3600      | 0           | 0           | 0           | 1/3600      |             |
| Fdir1_val_gps                        |       |             |             |             |             |             |             |             |             |             |             |
| gps1_activated                       |       |             |             |             |             |             |             |             |             |             |             |
| gps1_do_powerOff                     |       |             |             |             |             |             |             |             |             |             |             |
| gps1_do_powerOn                      |       |             |             |             |             |             |             |             |             |             |             |
| gps1_do_#tau                         |       |             |             |             |             |             |             |             |             |             |             |
| gps1.error                           | ok    | hot_failure | hot_failure | hot_failure | hot_failure | hot_failure | hot_failure | hot_failure | hot_failure | hot_failure | hot_failure |
| gps1_errorSubcomponent_activated     |       |             |             |             |             |             |             |             |             |             |             |
| gps1_errorSubcomponent_do_#hot_fault |       |             |             |             |             |             |             |             |             |             |             |
| gps1_errorSubcomponent_do_resetEvent |       |             |             |             |             |             |             |             |             |             |             |
| gps1_errorSubcomponent_do_#tau       |       |             |             |             |             |             |             |             |             |             |             |
| gps1_errorSubcomponent_reactivated   |       |             |             |             |             |             |             |             |             |             |             |
| gps1_errorSubcomponent_t_#delta      | 0     | 0           | 1/3600      | 1/900       | 0           | 1/3600      | 0           | 0           | 0           | 1/3600      |             |
| gps1.gps_activated                   |       |             |             |             |             |             |             |             |             |             |             |
| gps1.gps_do_#tau                     |       |             |             |             |             |             |             |             |             |             |             |
| gps1.gps.measurement                 |       |             |             |             |             |             |             |             |             |             |             |

Name:   Stored: No Filter  Edit



# Simulation guided by transitions

The screenshot shows the COMPASS Toolset interface. The 'Correctness' tab is active, and the 'Simulation' section is expanded. The 'Transitions' list is visible, with one transition highlighted and circled in red. A red text overlay reads 'An available transition has to be selected'. The 'Step1' window shows a sequence of modes: Bus.mode, dpu.mode, Fdir1.mode, gps1.gps.mode, gps1.mode, mission.mode, and mode.

**An available transition has to be selected**

| Name                   | Impl                           |
|------------------------|--------------------------------|
| Extended_avionics1.i   |                                |
| Fdir1 (1)              | Extended_Fdir1_gps_Fdir1.i     |
| gps1 (3)               | Extended_gps1_gps1.i           |
| _errorSubcomponent (2) | gpsError1.Implementation       |
| gps                    | Extended_gps1_gps_gpsDevice1.i |
| mission                | Extended_mission_mission1.i    |

| Step1       | Name          |
|-------------|---------------|
|             | Bus.mode      |
|             | dpu.mode      |
| init        | Fdir1.mode    |
| acquisition | gps1.gps.mode |
| offMode     | gps1.mode     |
| ground      | mission.mode  |
|             | mode          |

Transitions

- offMode -[\_errorSubcomponent.#hot\_fault when \_errorState = \_ok]-> offMode;
- offMode -[powerOff when \_errorState = \_transient\_failure]-> offMode;
- offMode -[powerOn when \_errorState = \_no\_more\_error]-> onMode;
- offMode -[\_errorSubcomponent.#transient\_fault when \_errorState = \_ok]-> onMode;
- onMode -[powerOff when \_errorState = \_transient\_failure]-> offMode;
- offMode -[powerOn when \_errorState = \_ok]-> onMode;
- offMode -[powerOff when \_errorState = \_ok]-> offMode;
- onMode -[\_errorSubcomponent.#transient\_fault when \_errorState = \_ok]-> onMode;

Name: \*.mode    Stored: No Filter    Edit



# Simulation guided by transitions

**No more transitions available**

**Time passing has to be defined**

**Constraints about time duration:**

No Constraint

No constraint

| Step1       | Step2       | Name          |
|-------------|-------------|---------------|
|             |             | Bus.mode      |
|             |             | dpu.mode      |
| init        | starting    | Fdir1.mode    |
| acquisition | acquisition | gps1.gps.mode |
| offMode     | onMode      | gps1.mode     |
| ground      | ground      | mission.mode  |
|             |             | mode          |

# Model checking

The screenshot displays the COMPASS Toolset interface. The 'Properties' panel on the left shows a list of properties, with 'Correctness' selected and circled in red. The 'Model Checker Options' panel on the right shows a warning icon and the text 'No results to show'.

**Selection of a properties**

# Model checking

The screenshot displays the COMPASS Toolset interface. The 'Correctness' tab is active, showing a table of properties. The first property, 'Correctness', is checked and has the formula 'The atomic proposition not dpu.failure always holds.' The 'Run Model Checking' button is visible, along with a checkbox for 'Model extended by Fault Injections'. Below this, the 'Model Checker Options' section is expanded, showing a message: 'The property is true up to bound 10' and 'The LTL property: G not dpu.failure has been found true up to bound 10.' This message is circled in red. The text 'The property is proved correct' is overlaid in red at the bottom of the image.

COMPASS Toolset

File Edit View Activities Help

Model Properties Mission TFPG Validation Correctness Performability Safety FDIR

**Properties**

| Name          | MC | Formula  |
|---------------|----|--|
| ✓ Correctness | ⓘ  | The atomic proposition not dpu.failure always holds. |

Run Model Checking  Model extended by Fault Injections

> Model Checker Options

ⓘ **The property is true up to bound 10**  
The LTL property: G not dpu.failure has been found true up to bound 10.

**The property is proved correct**

# Time divergence analysis

COMPASS Toolset

File Edit View Activities Help

Model Properties Mission TFGP Validation Correctness Performability Safety FDIR

Model Simulation Deadlock Checking Model Checking Zeno Analysis Time Divergence

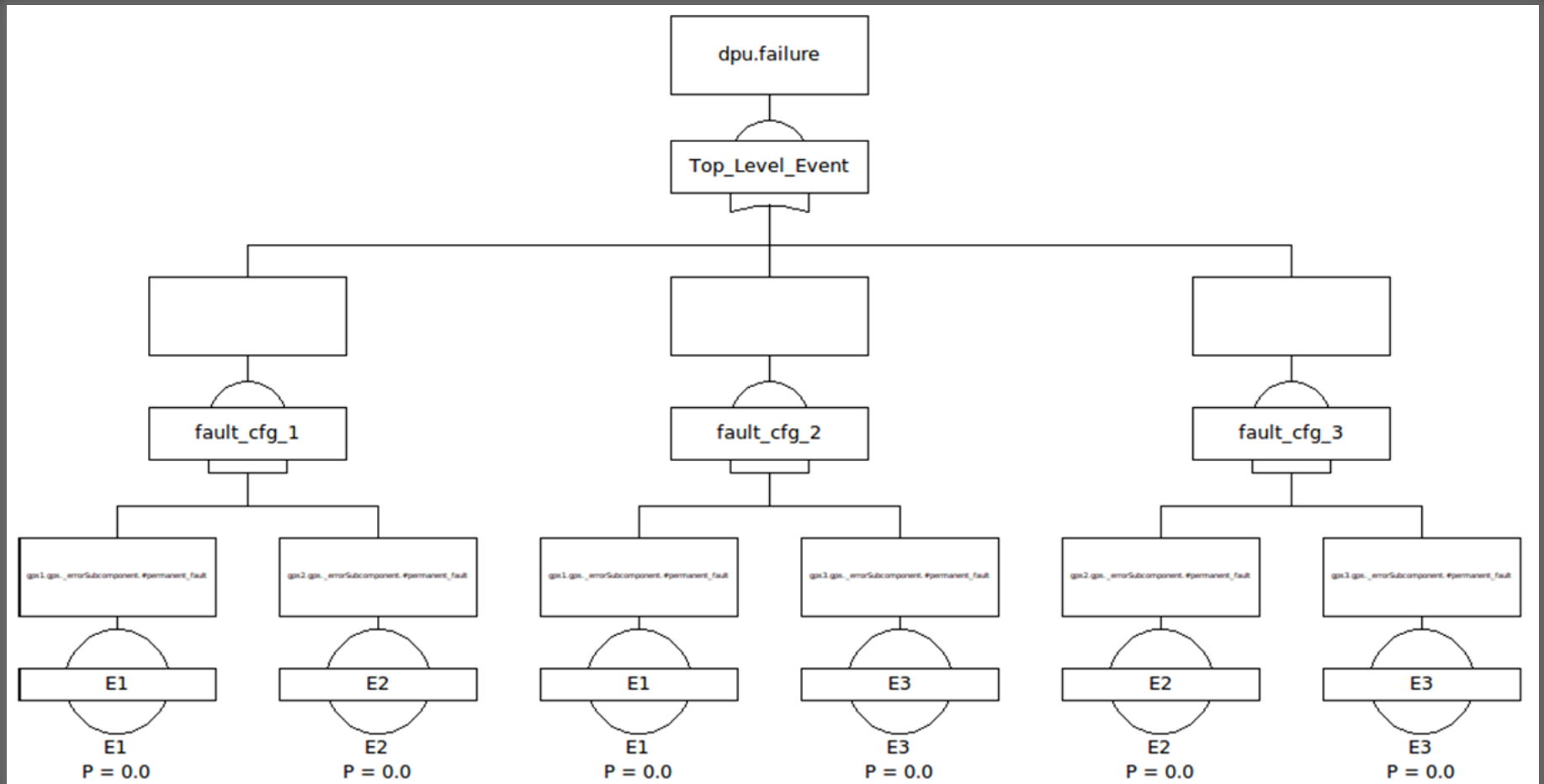
Run Time Divergence SAT Bound: 10 Model extended by Fault Injections

| Enabled                             | Clocks      | Bounds | Time Scale Unit | Result                               |
|-------------------------------------|-------------|--------|-----------------|--------------------------------------|
| <input checked="" type="checkbox"/> | Fdir1._t    | 0.0    | hour            | The clock Fdir1._t is "UNBOUNDED"    |
| <input checked="" type="checkbox"/> | gps1.gps._t | 0.0    | hour            | The clock gps1.gps._t is "UNBOUNDED" |
| <input checked="" type="checkbox"/> | mission._t  | 0.0    | hour            | The clock mission._t is "UNBOUNDED"  |

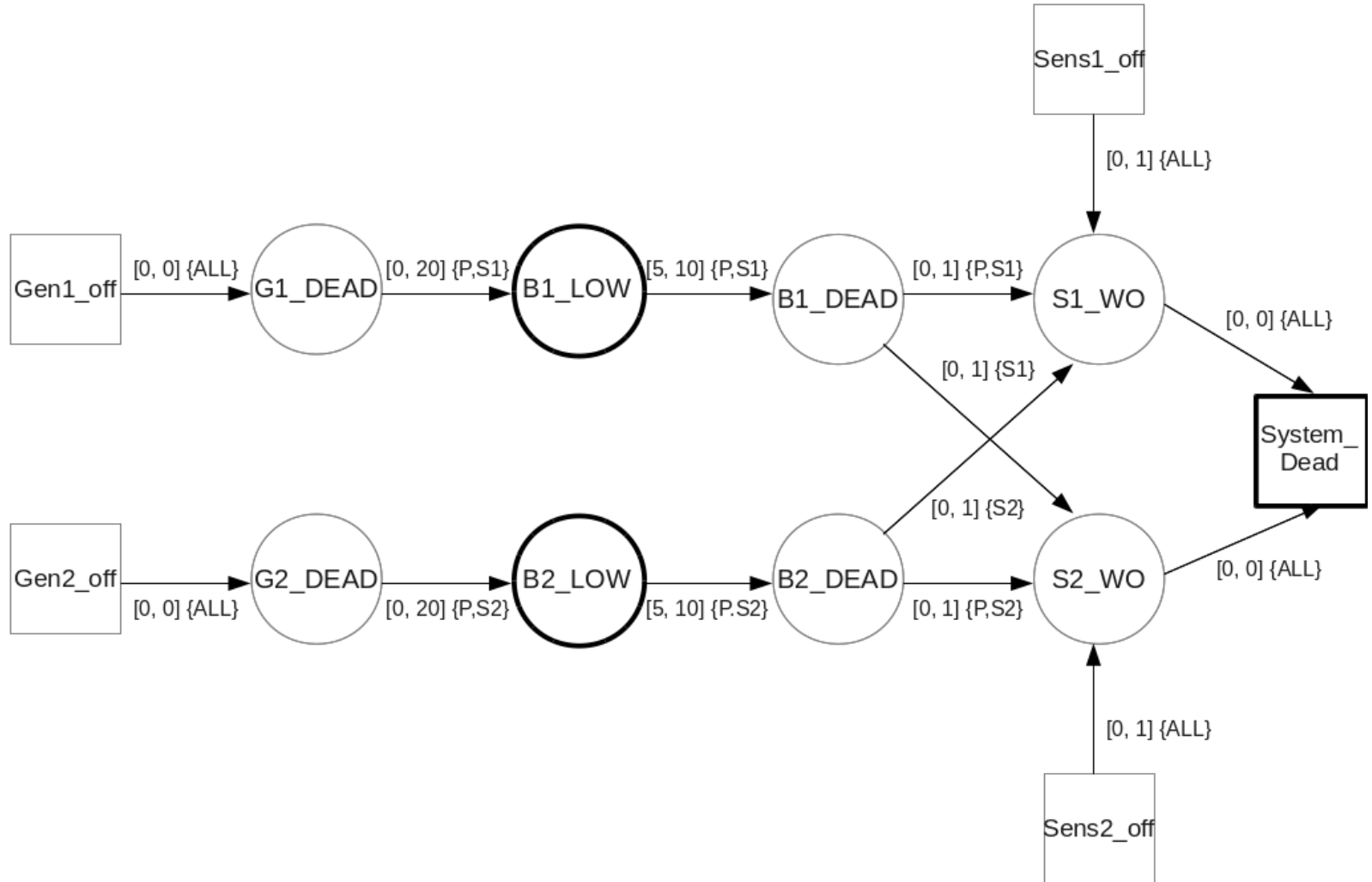
**The model is incorrect**

**⚠ No Results to show.**  
There are no results to show at the moment.

# 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>

---

**HASDEL**

Hardware Software Dependability for Launchers

09/12/2014 p42



**AIRBUS**  
DEFENCE & SPACE



FONDAZIONE  
BRUNO KESSLER

**RWTHAACHEN**  
UNIVERSITY



# 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

09/12/2014 p43



**AIRBUS**  
DEFENCE & SPACE



**RWTHAACHEN**  
UNIVERSITY

# HASDEL

Hardware Software  
Dependability  
for Launchers

## Thank you for your attention

Any question ?

David Lesens  
Joost-Pieter Katoen  
Alessandro Cimatti

[david.lesens@astrium.eads.net](mailto:david.lesens@astrium.eads.net)  
[katoen@informatik.rwth-aachen.de](mailto:katoen@informatik.rwth-aachen.de)  
[cimatti@fbk.eu](mailto:cimatti@fbk.eu)



**AIRBUS**  
DEFENCE & SPACE

**FBK**  
FONDAZIONE  
BRUNO KESSLER

**RWTHAACHEN**  
UNIVERSITY