OMEGA

Correct development of Real-Time embedded systems in UML

EADS SPACE Transportation Case Study

Ariane 5 Flight Software

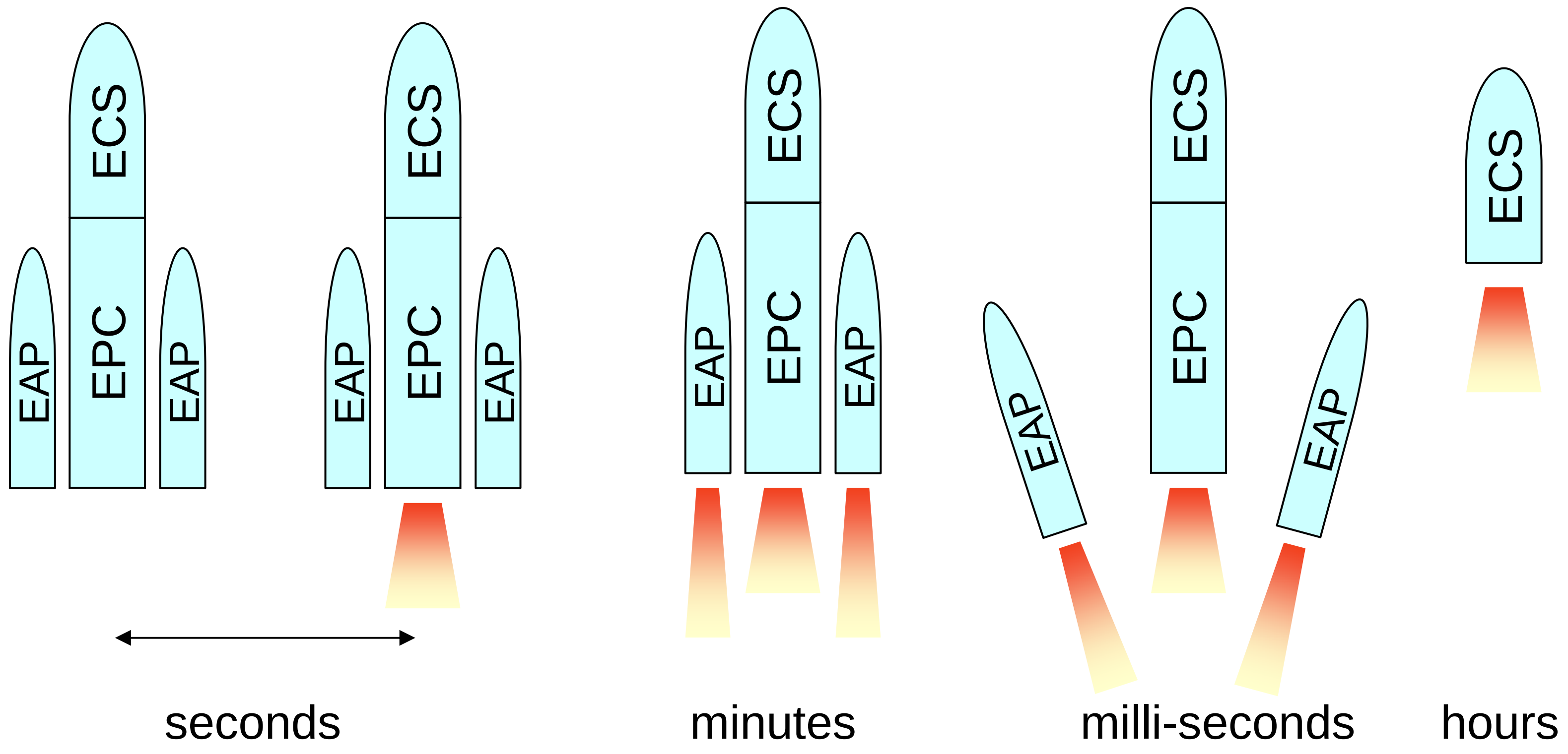
David LESENS

Email: david.lesens@space.eads.net

Overview

- **Description of the Ariane 5 case study**
  - Merge asynchronous and cyclical behaviors
  - Environment
- **Asynchronous behavior**
- **Cyclic behavior**
- **Tools**
- **Evaluation & Conclusion**

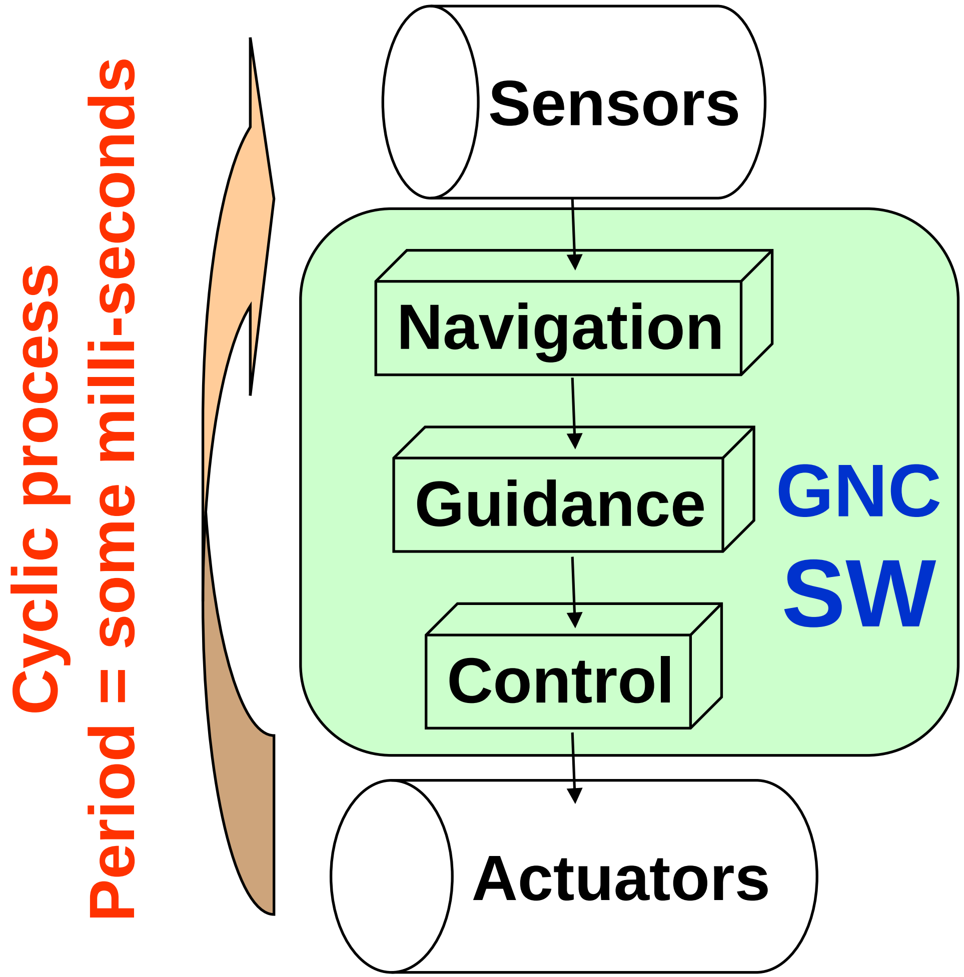
Spacecraft management / mission phases



Spacecraft management deals with timed sporadic events

⇨ Use of timed asynchronous semantics

Spacecraft control / command



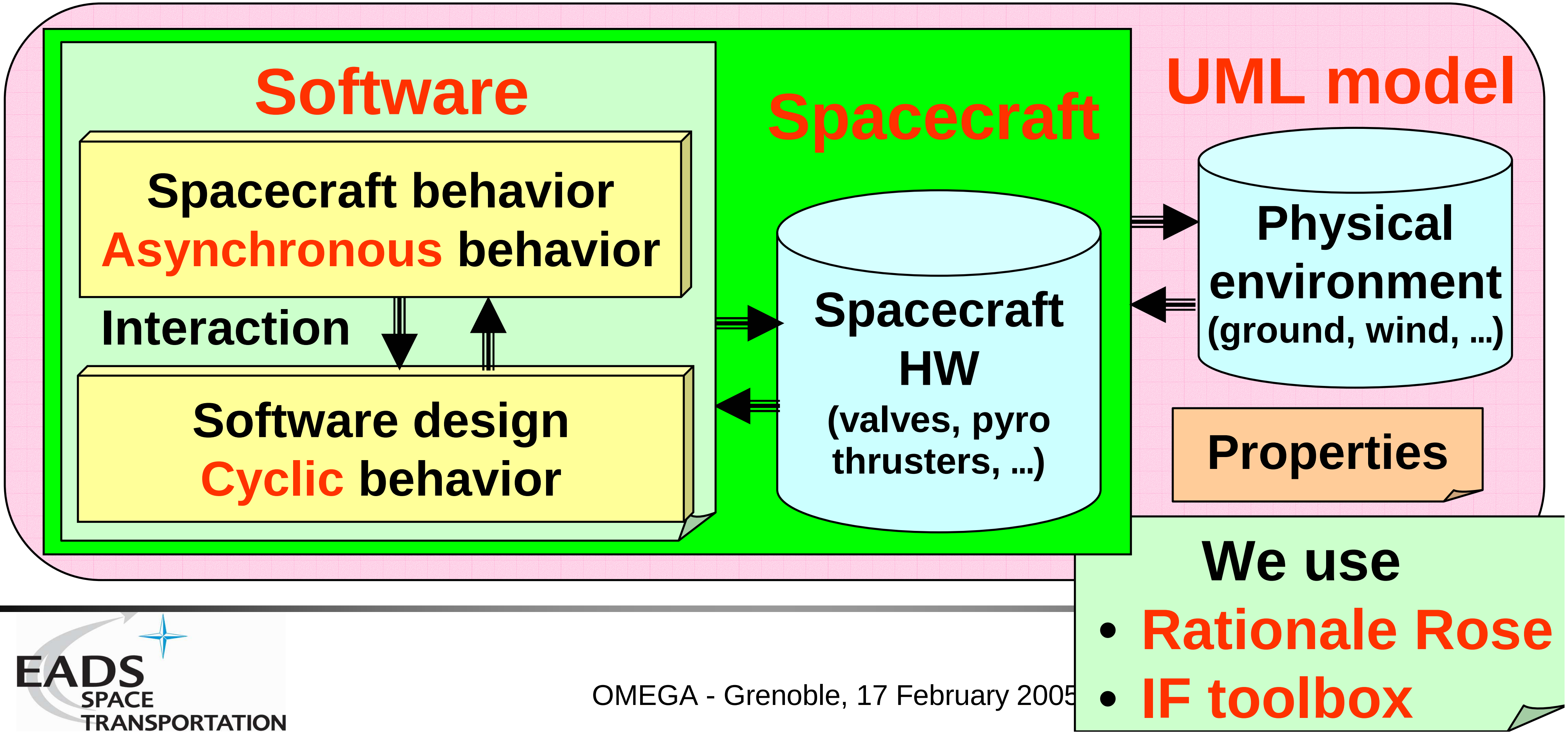
- Acquisition of measurement
- Where am I ?
- Where shall I go ?
- Compute the commands
- Send commands to actuators

Spacecraft control command

Globally Asynchronous / Locally Synchronous (GALS)

Construction of the UML model

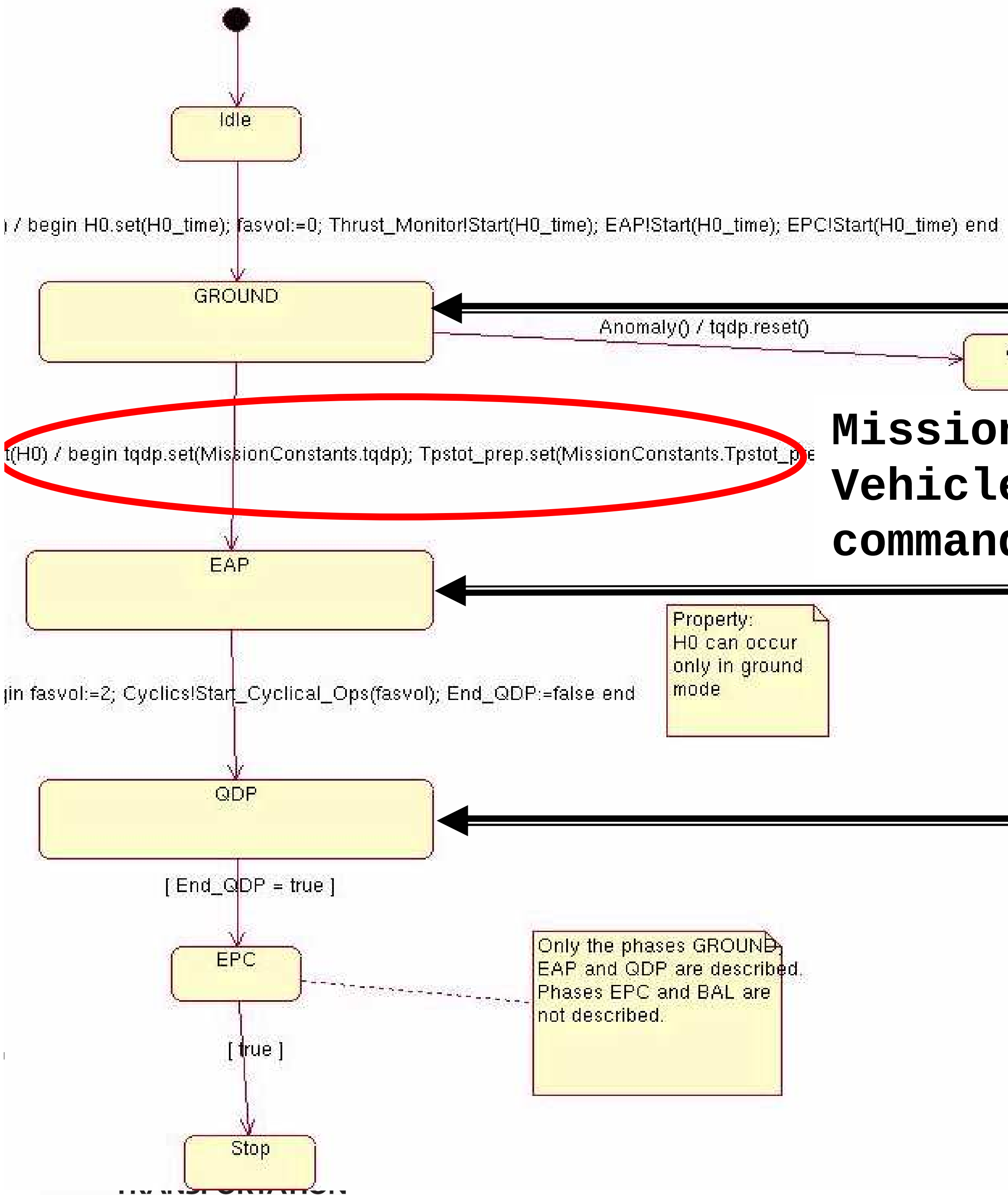
- 1) Development of the spacecraft behavior model
- 2) Addition of complement for the SW ctrl / cmd design
- At each stage: environment and properties



Overview

- **Description of the Ariane 5 case study**
- **Asynchronous behavior**
  - Model
  - Property description
- **Cyclic behavior**
- **Tools**
- **Evaluation & Conclusion**

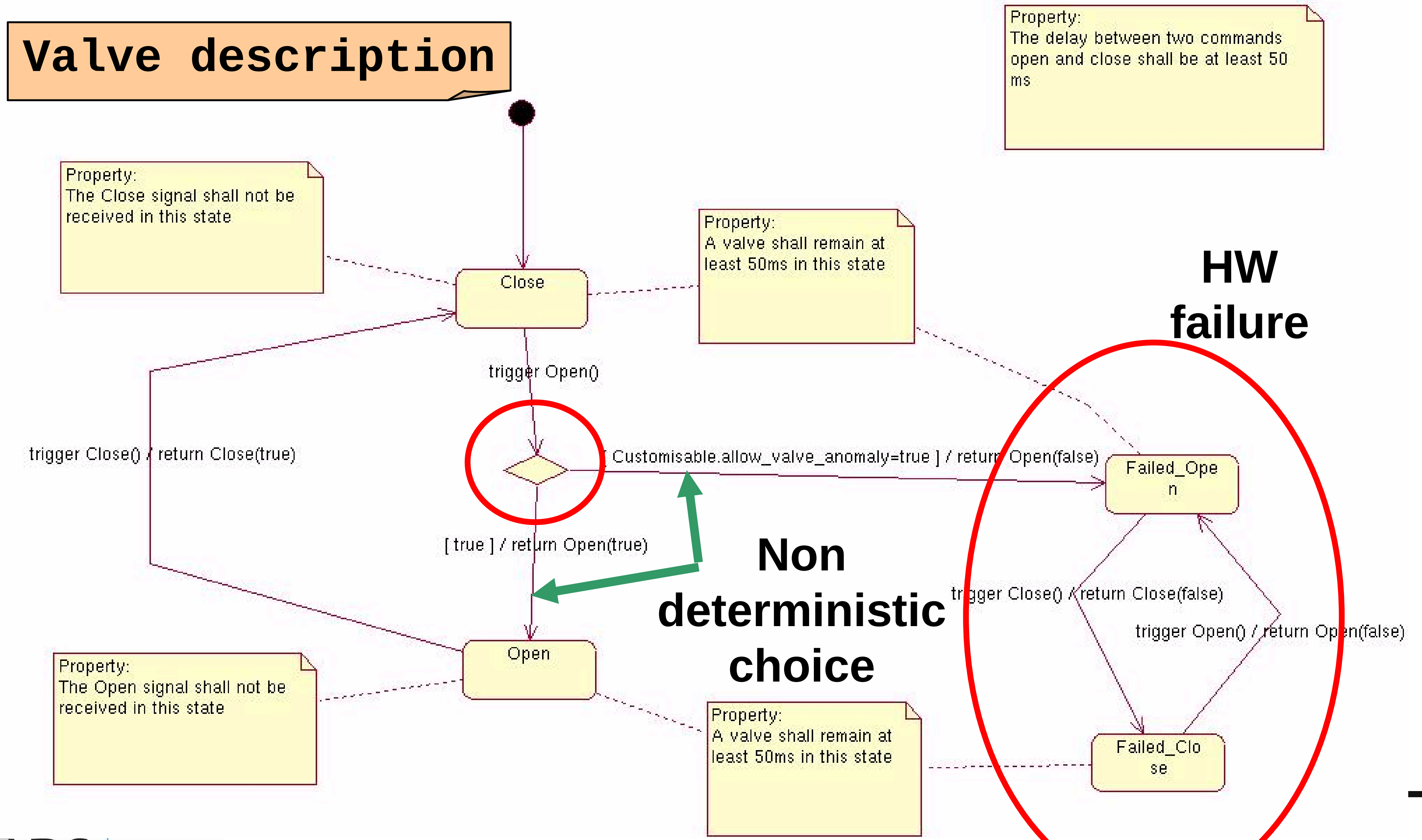
Mission description



Mission timing
Vehicle mode change command

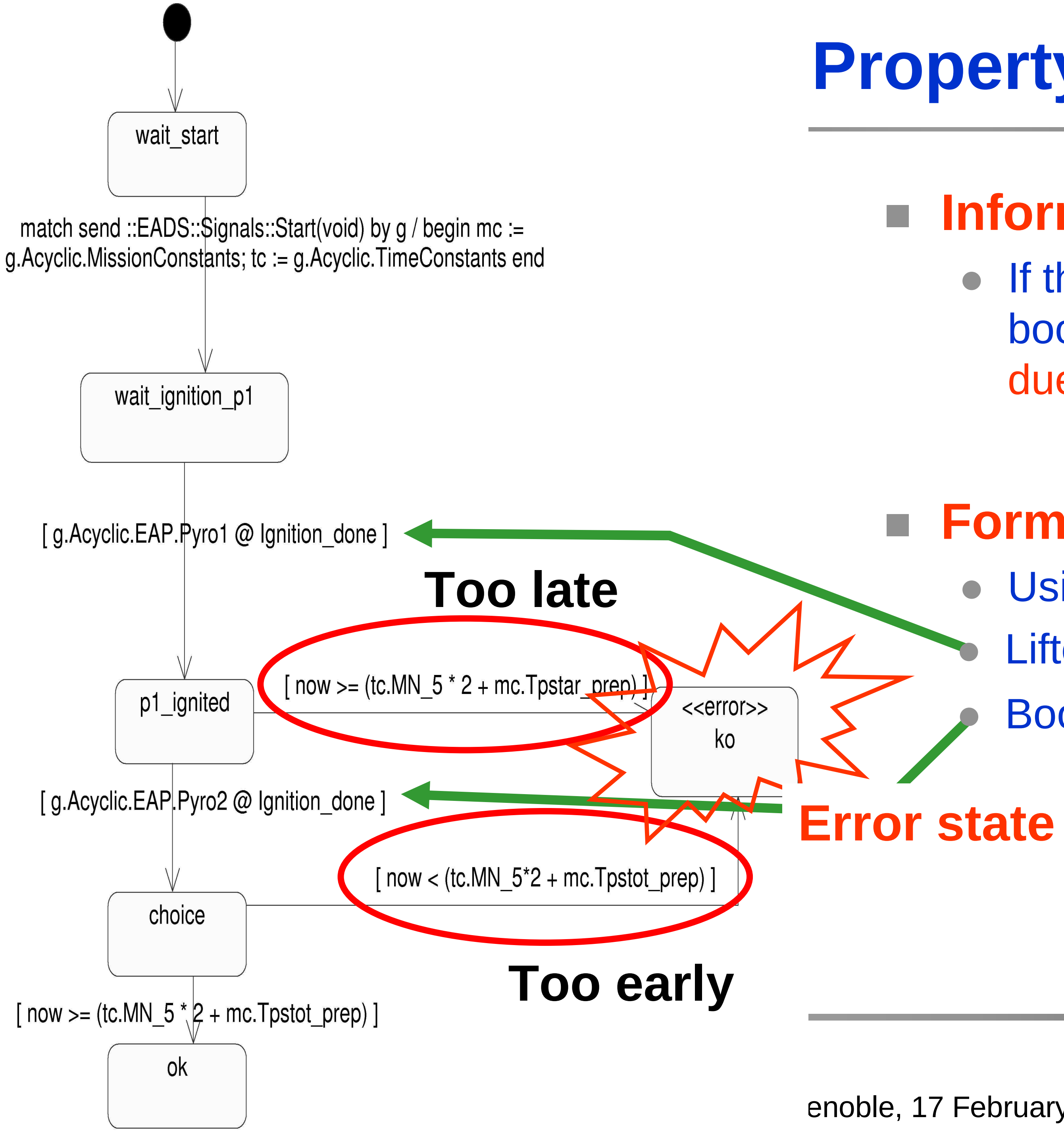
Flight phases

Environment description



Property example (timed)

- **Informal description**
  - If the liftoff is performed, the boosters shall be released at due time.
- **Formal description**
  - Using an observer
  - Liftoff = pyro1.ignition
  - Boosters release = pyro2.ignition



Overview

- Description of the Ariane 5 case study
- Asynchronous behavior
- Cyclic behavior
  - Bus model
  - Multitasking model
  - CPU consumption model
- Tools
- Evaluation & Conclusion

Real time software design

Design of the Ariane 5 Flight Software (also used for ATV, Vega, …)

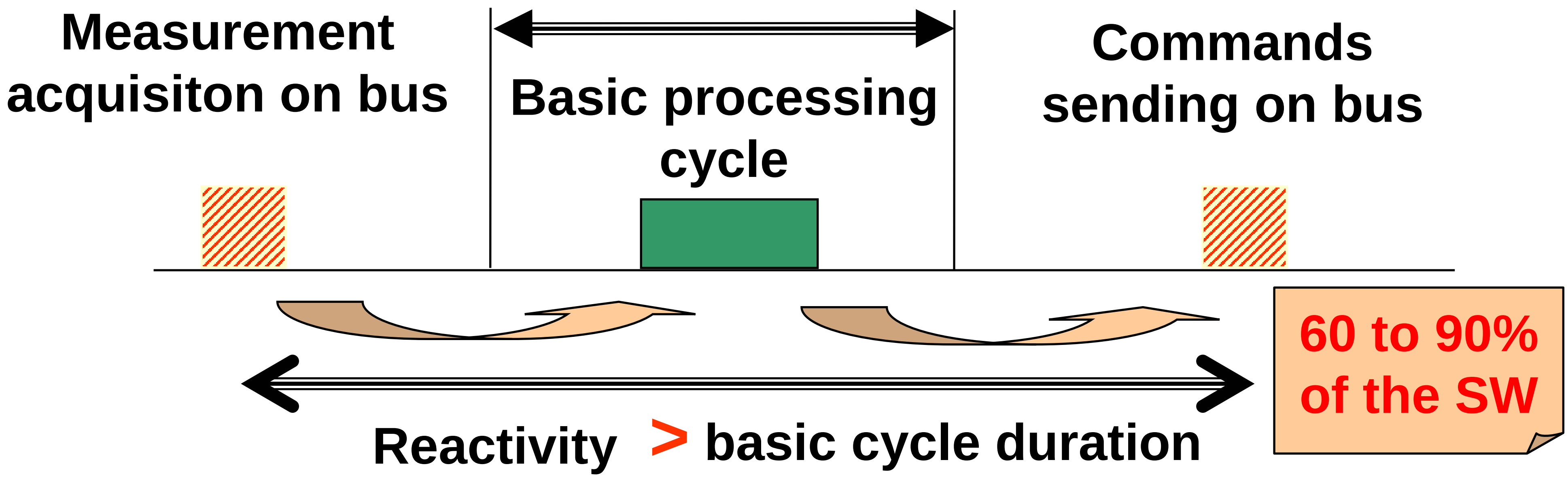
- **Use of a 1553 MIL BUS**
  - Reservation of predefined timed slot for each type of transfer
    - Bus access forbidden during physical transfer
  - Definition of a bus frame with respect to the required reactivity
- **Multitasking**
  - One thread by frequency
    - 1Hz, 10Hz, acyclic, …
  - Preemptive with fix priority
    - The higher frequency has the higher priority

A real time scheduler runs the different processes, taking into account the multitasking and the bus frame

Real time design : low reactivity

- Bus frame construction (depending of the required reactivity)
- Real time scheduler
- Measurement available at cycle start
- Commands sent at cycle end

Synchronous hypothesis
Use of SCADE
Correct “a priori”



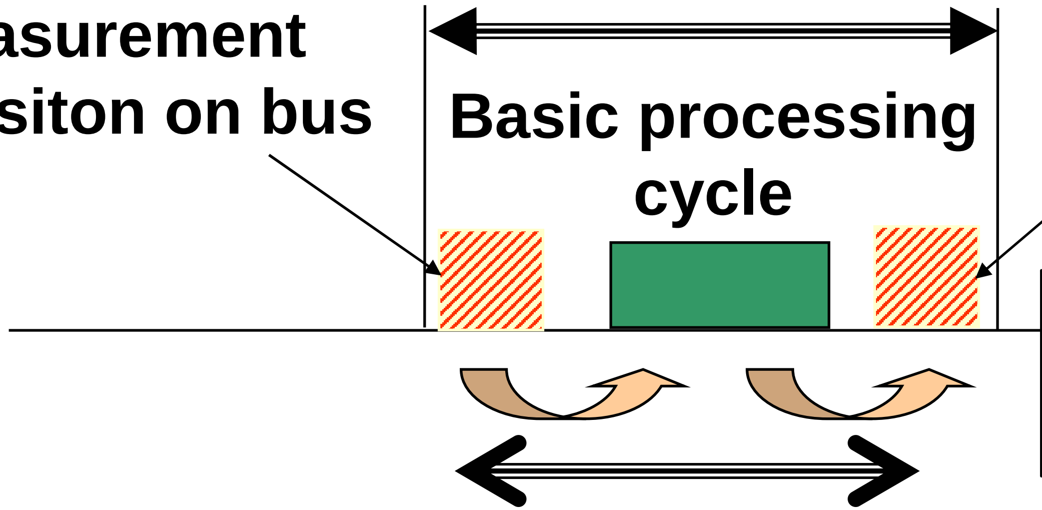
Real time design : high reactivity

- Bus frame construction (depending of the required reactivity)
- Real time scheduler
- Measurement available during the cycle
- Commands sent during the cycle

Synchronous hypothesis violated
Use of Ω UML
Verification "a posteriori"

Measurement acquisiton on bus

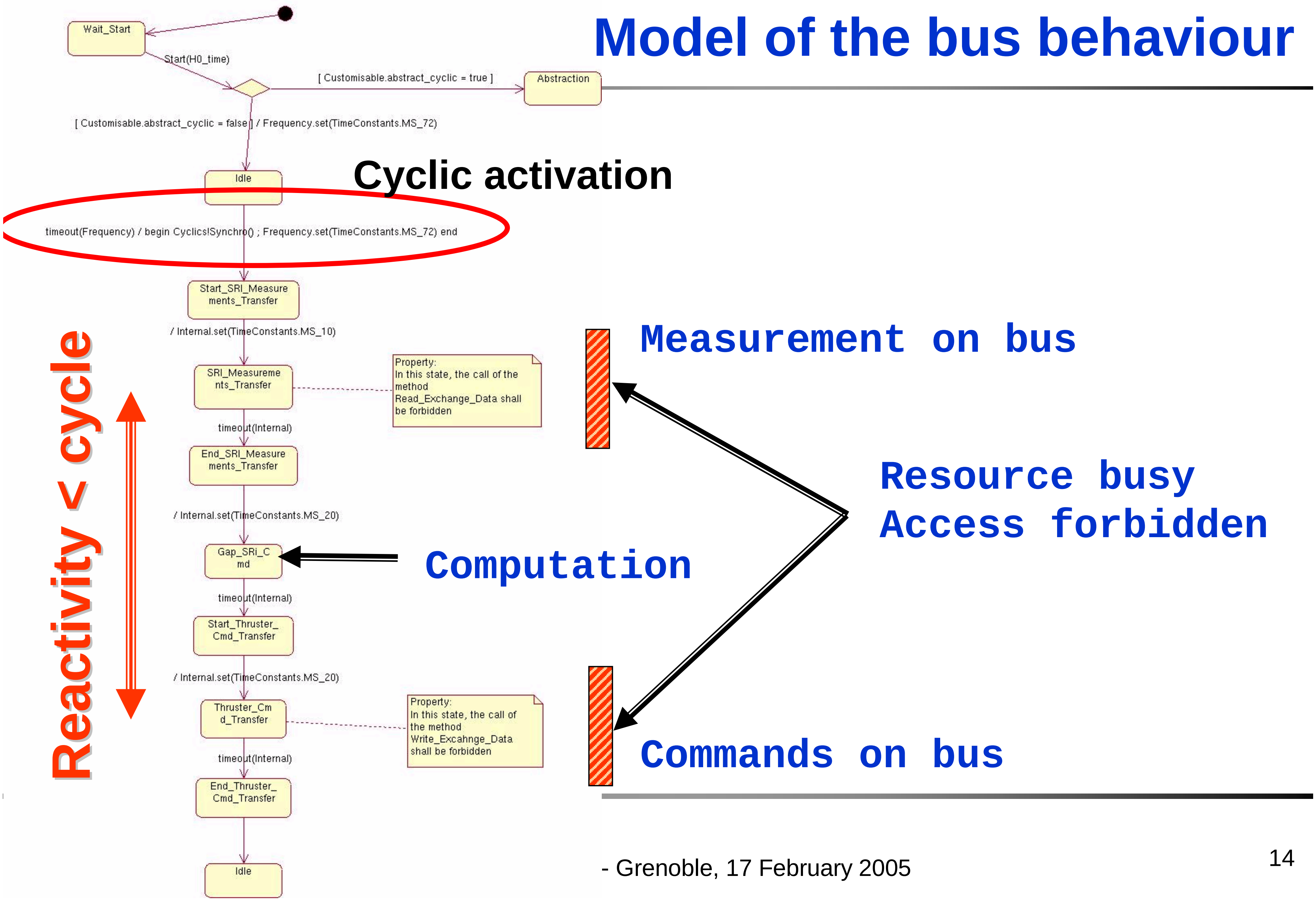
Commands sending on bus



10 to 40% of the SW
⇨ Hard point analysis

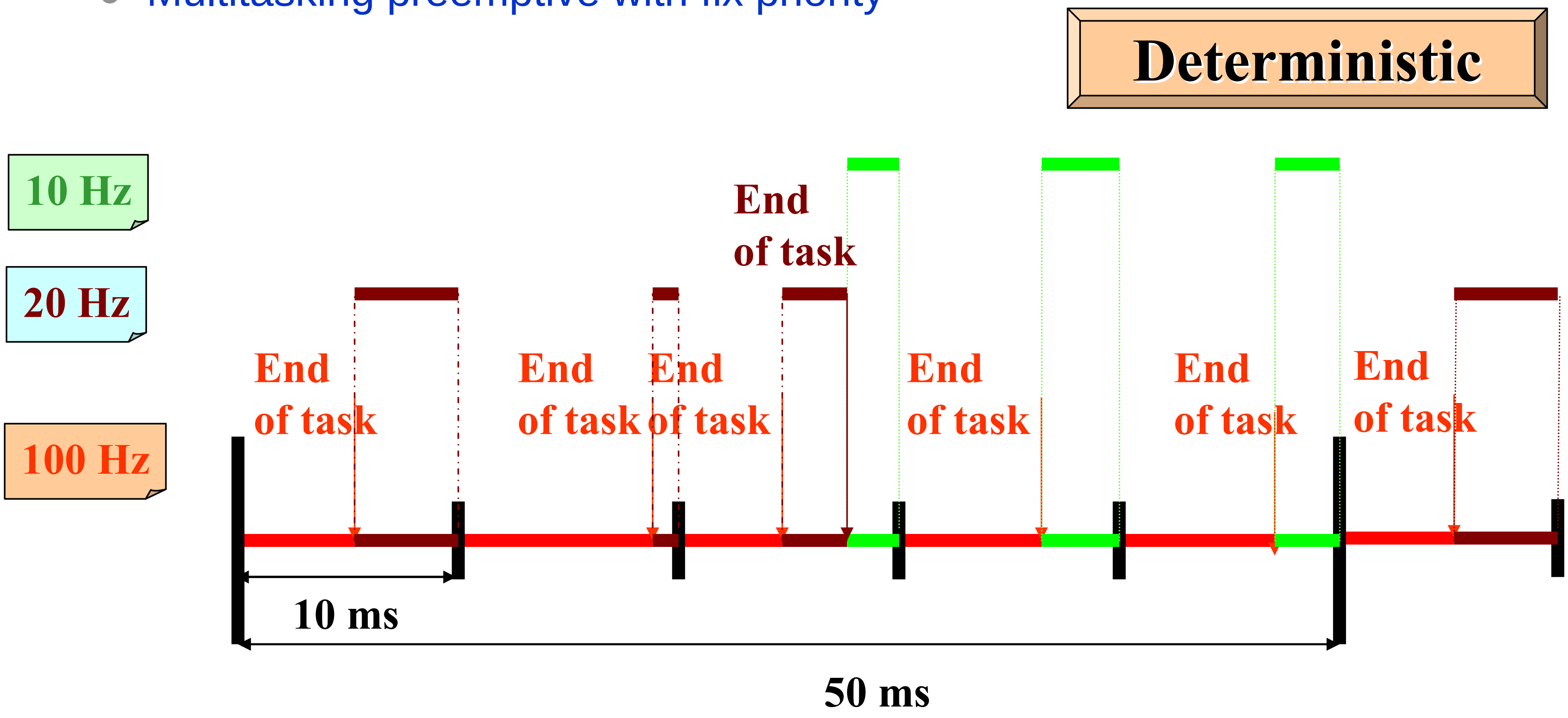
Reactivity < basic cycle duration

Model of the bus behaviour



Multitasking at EADS ST

- Multitasking preemptive with fix priority



Model of multitasking in Ω UML

- **Definition of task priority**

```
begin
   theTask := new::CPU::Task::Task(1, Acyclic.Ground.CPU)
end
```

This task has the first priority

- **Definition of CPU consumption for each function**

```
begin
   Cyclics.theTask.exec(5)
end
```

This action consumes 5 units of time

Overview

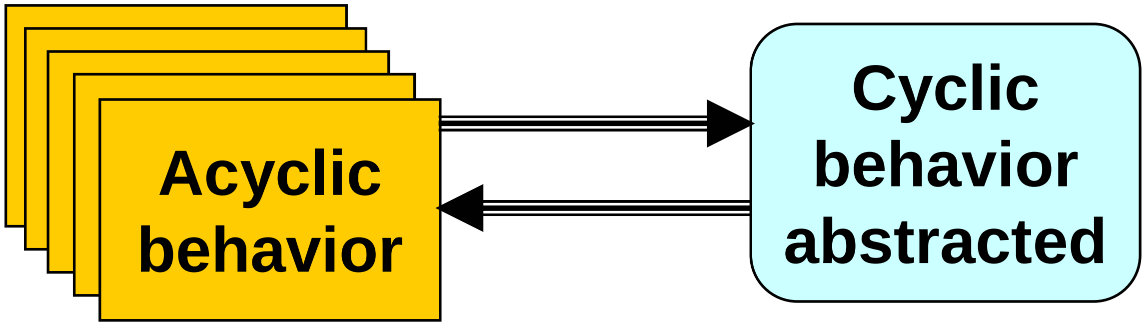
- **Description of the Ariane 5 case study**
- **Asynchronous behavior**
- **Cyclic behavior**
- **Tools: IF Toolbox**
  - Problem of time scale
  - Simulator
  - Proof tool
- **Evaluation & Conclusion**

Time scale problem

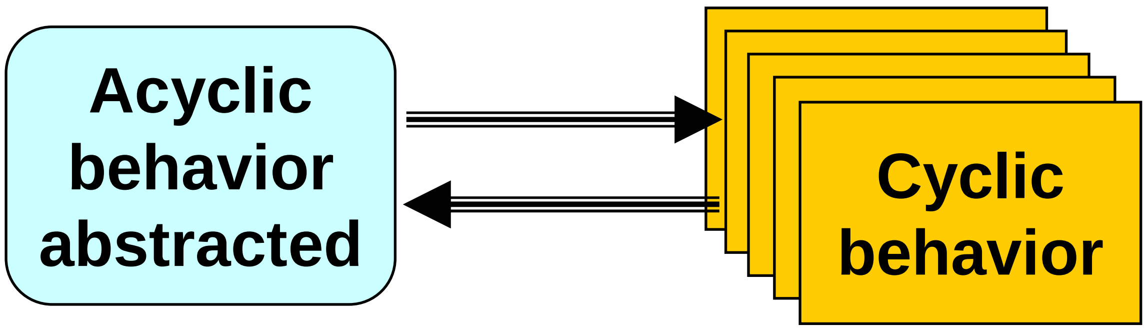
- **Basic cycle of the cyclic behavior**
  - 100 ms
  - About 100 steps
- **1 hour mission**
  - 3 600 000 steps
- **6 months mission**
  - 15 000 000 000 steps
- **15 years mission**
  - 300 000 000 000 steps
- **Explosion of the number of states**
  - Several hours/days/… of simulation => not usable
  - Limit of the proof tools reached

Time scale problem: first solution

- **Abstraction of the cyclic parts**
  - Proof of the asynchronous part without the cyclic part



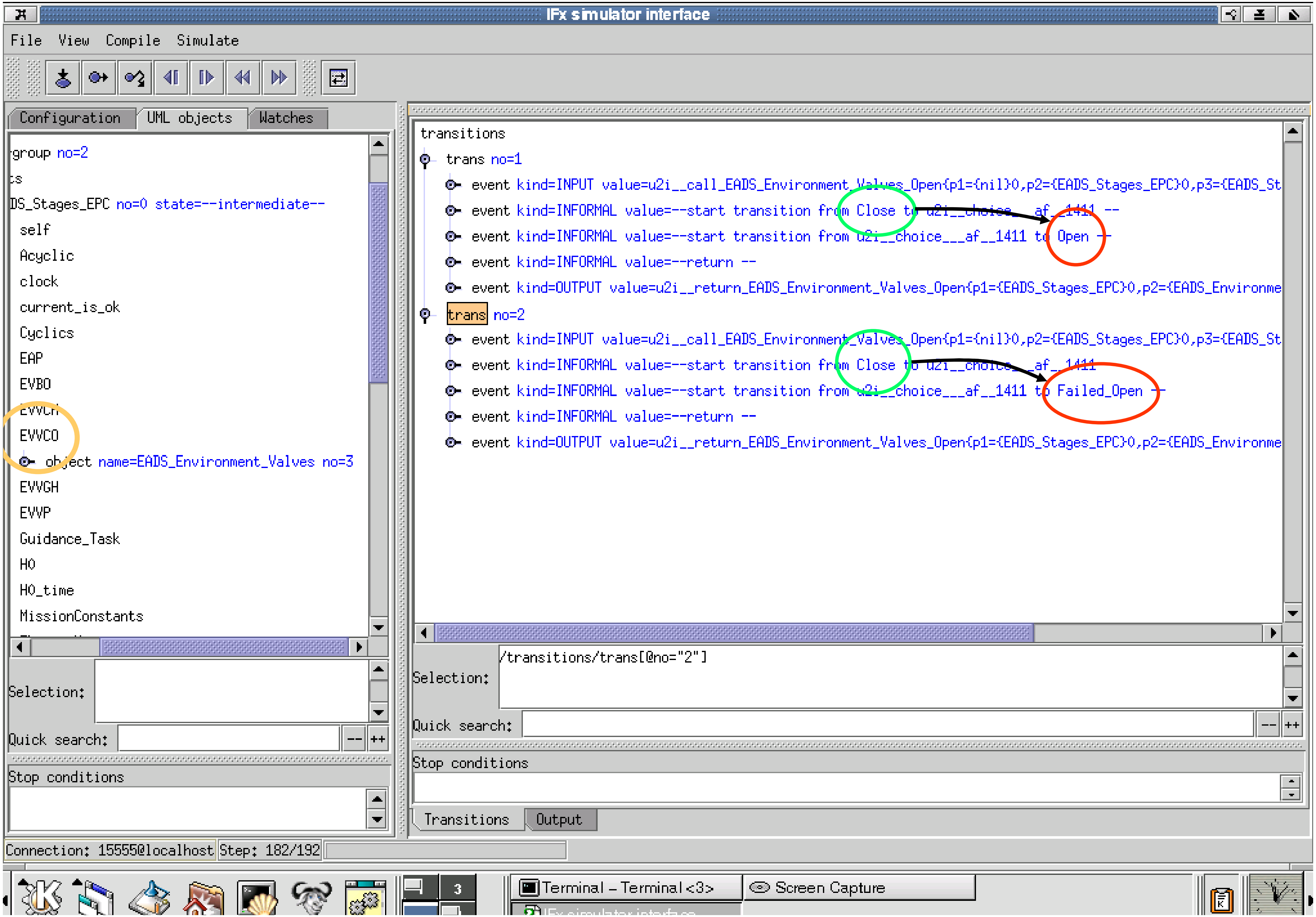
- **Abstraction of the asynchronous parts**
  - Proof of the cyclic part without the asynchronous part



Time scale problem: second solution

- **Reduction of the mission duration**
  - 30 seconds mission instead of 1 hour
  - 30 000 steps
  - Whole system validated (cyclic + acyclic)

	Mission duration	Number of states	Number of transitions	Proof duration
1	7 000 ms	51 324	54 697	00:03:30
2	15 000 ms	161 956	171 734	00:12:06
3	22 000 ms	303 496	321 206	00:11:33
4	30 000 ms	463 932	490 901	00:22:58
5	37 000 ms	658 981	696 031	00:34:53


IFx simulator interface
File View Compile Simulate
Configuration
UML objects
Watches
group no=2
DS_Stages_EPC no=0 state=--intermediate--
self
Acyclic
clock
current_is_ok
Cyclics
EAP
EVBO
EVVCO
object name=EADS_Environment_Valves no=3
EVVGH
EVVP
Guidance_Task
HO
HO_time
MissionConstants
Selection:
Quick search:
Stop conditions
transitions
trans no=1
event kind=INPUT value=u2i__call_EADS_Environment_Valves_Open{p1={nil}0,p2={EADS_Stages_EPC}0,p3={EADS_St
event kind=INFORMAL value=--start transition from Close to u2i__choice___af__1411 --
event kind=INFORMAL value=--start transition from u2i__choice___af__1411 to Open --
event kind=INFORMAL value=--return --
event kind=OUTPUT value=u2i__return_EADS_Environment_Valves_Open{p1={EADS_Stages_EPC}0,p2={EADS_Environme
trans no=2
event kind=INPUT value=u2i__call_EADS_Environment_Valves_Open{p1={nil}0,p2={EADS_Stages_EPC}0,p3={EADS_St
event kind=INFORMAL value=--start transition from Close to u2i__choice___af__1411
event kind=INFORMAL value=--start transition from u2i__choice___af__1411 to Failed_Open --
event kind=INFORMAL value=--return --
event kind=OUTPUT value=u2i__return_EADS_Environment_Valves_Open{p1={EADS_Stages_EPC}0,p2={EADS_Environme
Selection: /transitions/trans[@no="2"]
Quick search:
Stop conditions
Transitions
Output
Connection: 15555@localhost
Step: 182/192
Terminal – Terminal <3>
Screen Capture

IF proof tool example

```
(m023206.musun03176) A5_RAF_25_error_EVVGH_twice.x -dfs -po -me -ce
reached error state [335]
00:00:00      536/s      569/t
   queue table :     112 items     13/entry  4/min 16/max  8.62/avg
 message table :      78 items     13/entry  4/min  8/max  6.00/avg
instance table :    3738 items    127/entry  0/min 260/max 29.43/avg
  config table :     822 items     29/entry 16/min 40/max 28.34/avg
   chunk table :    1678 items     59/entry 20/min 40/max 28.44/avg
   label table :     984 items     59/entry  8/min 24/max 16.68/avg
   event table :     896 items     29/entry 20/min 48/max 30.90/avg
(m023206.musun03176) ll *.scn
-rw-rw-r--   1 m023206  fas_dev   181483 May 27 09:36 e1.scn
```

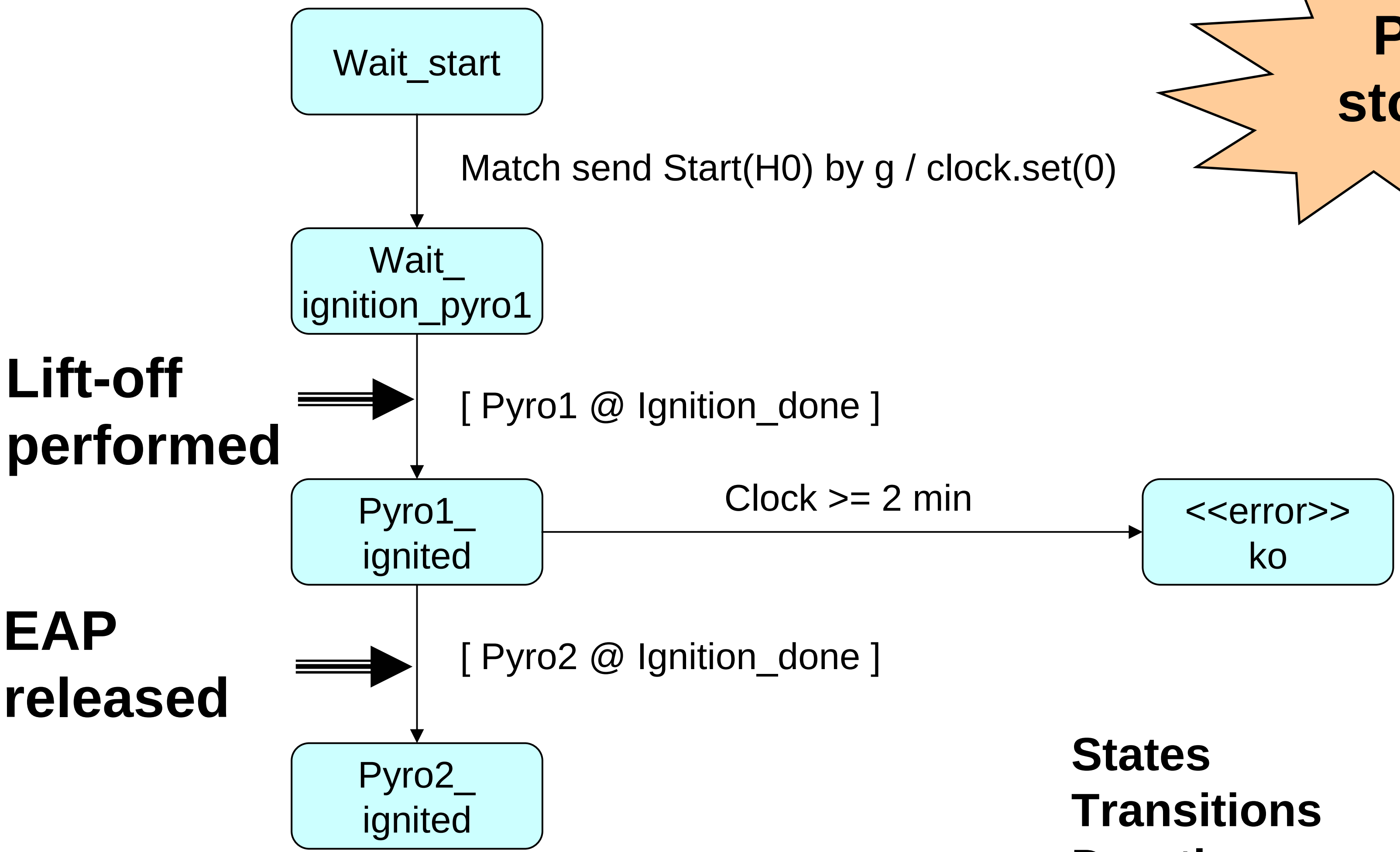
- depth first
- partial order
- mark errors
- cut on errors

A counter example is generated

Metrics on the proof tool

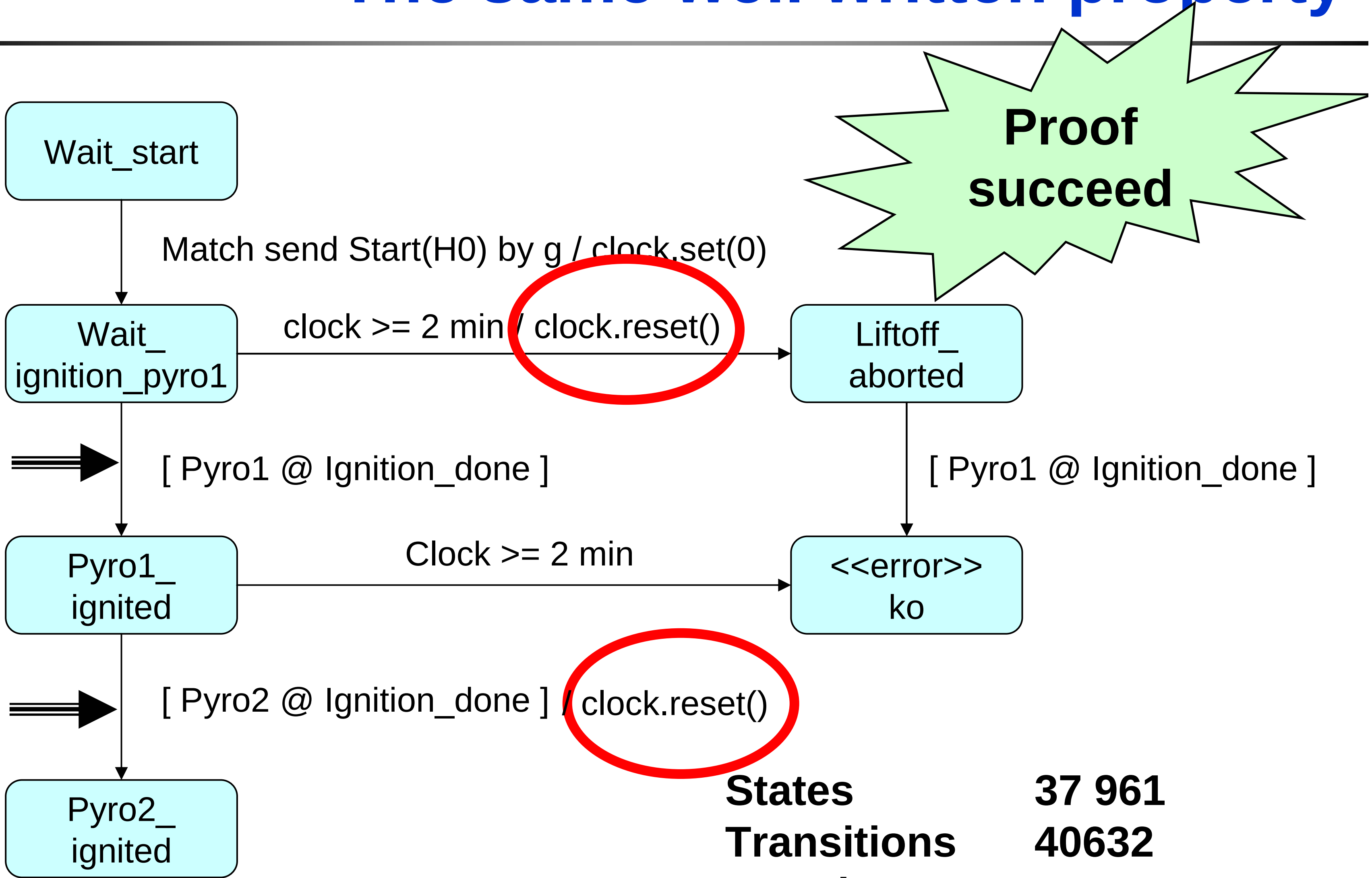
Property	Number of states	Number of transitions	Proof duration
liftoff_aborted_right	36037	38149	00:00:36
pyro_not_ignited_twice	35988	38092	00:00:42
valve_not_abused	36082	38210	00:00:37
valve_not_close_in_close	36010	38114	00:00:44
valve_not_open_in_open	35998	38102	00:00:38
liftoff_performed_right1	46075	48713	00:00:49
liftoff_performed_right2	37897	40550	00:00:55
liftoff_performed_right3	37961	40632	00:01:12
liftoff_performed_right3 no_clock_reset	1048028 abort	1347402 abort	00:42:50 abort
liftoff_performed_right4	35986	38090	00:00:38
CPU_not_in_error	35980	38084	00:00:53
G_cycle_is_schedulable	36012	38116	00:00:48
NC_cycle_is_schedulable	36380	38484	00:00:39
read_write_coherence	36618	38722	00:00:47

A bad written property



States	**> 1 048 028**
Transitions	**> 1 347 402**
Duration	**> 00:42:50**

The same well written property



Lift-off performed

EAP released

States	**37 961**
Transitions	**40632**
Duration	**00:01:12**

Overview

- Description of the Ariane 5 case study
- Asynchronous behavior
- Cyclic behavior
- Tools
- Evaluation & Conclusion

Summary of work

- **Model of**
  - The spacecraft behavior (mission management, asynchronous)
  - The ctrl / cmd SW (control / command, cyclic behavior)
    - Multitasking
    - CPU consumption
  - The environment
    - Avionics (valve, pyrotechnic commands, ground control center, …)
    - Communication bus
- **Validation of all the specified properties**
  - By simulation
  - By proof

  Proof is a complement of test but does not replace test

  - Detection of intentional bugs of the model

Cyclical behavior

- **For “low” reactivity needs**
  - Use of the synchronous hypothesis
  - Control command described using SCADE
  - 60% to 90% of the software

- **For “high” reactivity needs**
  - When the synchronous hypothesis is violated
    (Required reactivity < basic cycle duration)
  - Asynchronous semantics
  - Important effort of modeling (several thousands of bus transfers)

⇨ **OMEGA UML for hard point analysis**

IF simulator

- Respect the OMEGA syntax/semantics
- Powerful debugging facilities

- Precise interpretation of results requires some knowledge of tool internals
  - No automated feedback from the VERIMAG tool towards the UML tool
  - Objects of the IF model are visible, even if not defined by the user
  - Slow for big scenario (>30Mb, >30000 transitions)

  ⇒ Hard to use in practice for “cyclical” debugging (several hours)

Property description

Use of observers

- **Powerful**
  - Untimed and timed properties
  - Intrusive properties (“integration tests”)
  - Non intrusive properties (“validation tests”)
- **Property description formalism very easy to understand**
  - Finite state machine
  - Defined in OMEGA UML syntax and semantics
  - Intuitive concept of real time

IF proof tool

- **Answers all the user needs**
  - Very quick result
  - In case of non satisfied property, computation of a failed scenario
  - All properties proved

- **Same defaults as the simulator**
  - **No feedback from the VERIMAG tool toward the UML tool**
  (for the computed failed scenario)
  - No usable failed scenario for big models
  => Debugging not easy

General conclusion

- **OMEGA UML allows to model the real time behaviors of a spacecraft**
- **Validation early in the development cycle**
  - Improve the software quality
  - Decrease the software development costs

- **No link with SCADE**

We need industrial tools to use OMEGA UML

Questions ?