MODEL-BASED SOFTWARE ENGINEERING at ESA

Maxime Perrotin – Andreas Jung – Marcel Verhoef | TEC-SW

ADCSS-2019

ESA UNCLASSIFIED - For Official Use

Outline

1. What is MBSE ?
2. History : some background on modelling technologies
3. A vision for the future
4. Current state of the art (TASTE, OSRA)
5. Use in projects
6. Conclusion and the short-term future

What is MBSE ?

GOALS

Simplify the development and improve the quality of computer-based systems using

- **Mature and well-defined** languages and processes
- Tools to achieve **correctness by construction**

Make software engineering part of the System activities via MBSE

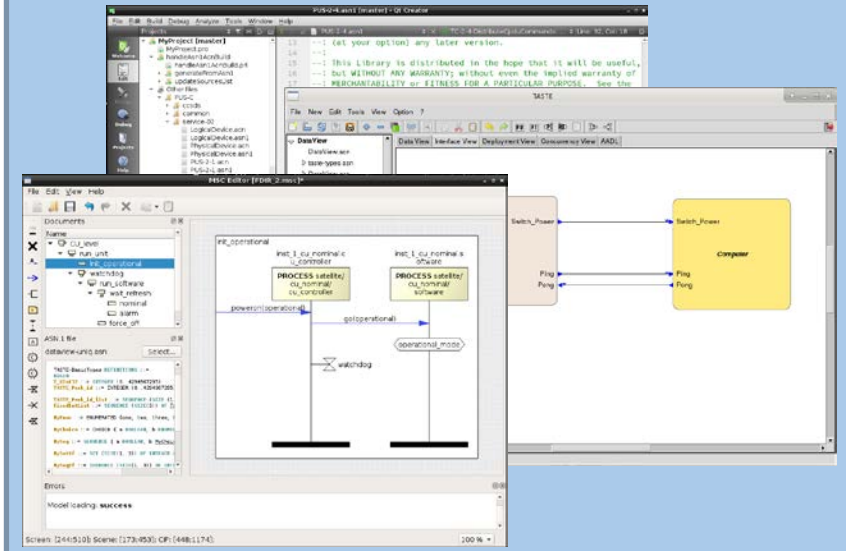
TARGETS

Real-time, distributed embedded systems (flight and ground)

MBSE : Models and Methodology covering…

SPECIFICATION

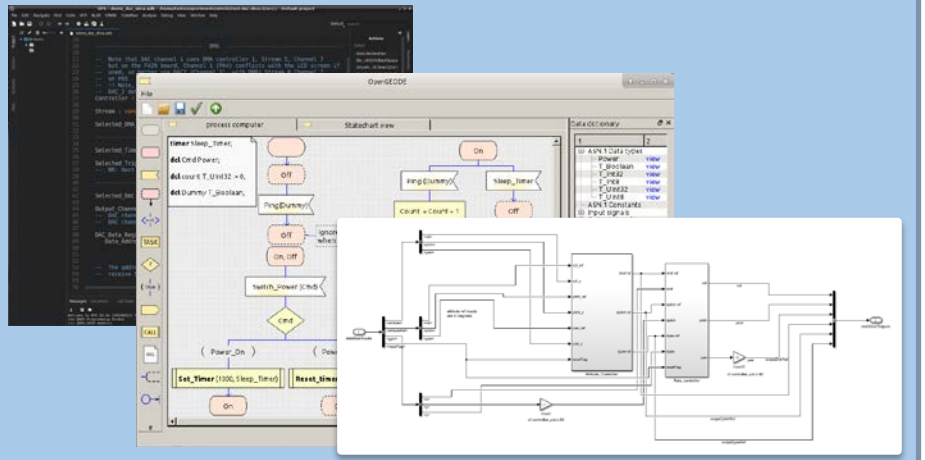
Architecture – Data - Dynamics



DESIGN AND CODE

Mix models with code

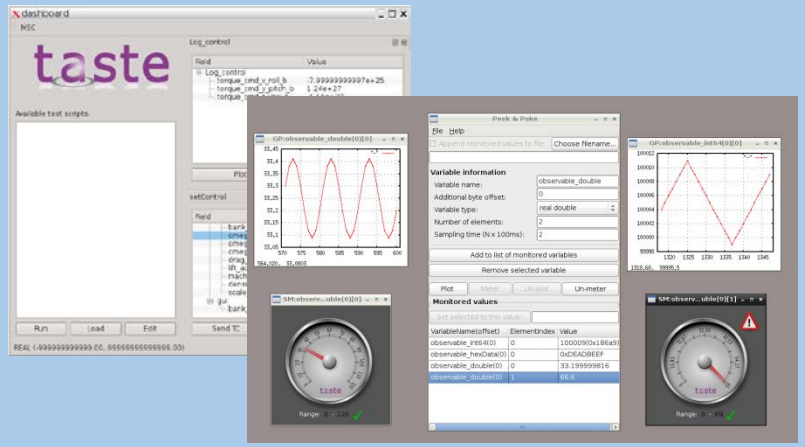
C, C++, Ada, Simulink, SDL, Micropython, VHDL, Modelica...



Use models to create a system

SIMULATION AND TESTING

…As early as possible…



GENERATION OF CODE, TESTS, AND DOCUMENTATION

On Proba-3, 300 pages of ICD:

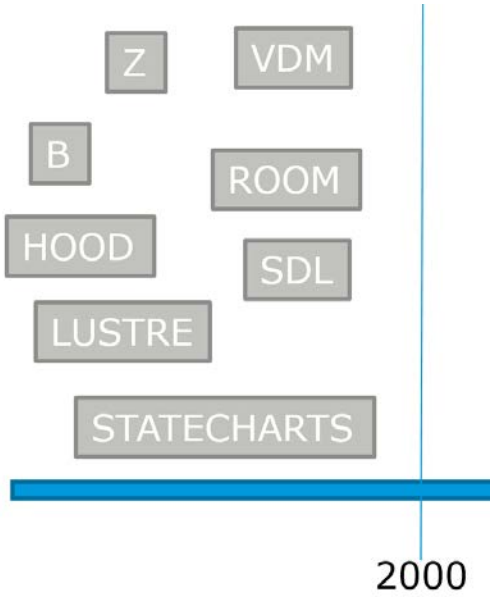
BSW-TM-SourceData (CHOICE) — Min: 0 bytes Max: 1030 bytes

List representing all possible data structures contained by TM.
Only single item from this list can be present in TM at once.
Present item is determined by reporting service type and sub-type.

No	ACN Parameters	Type
1	type	UInt8
2	subType	UInt8

No	Field	Comment	Present	Type	Min Bits	Max Bits
1	ackSuccess	Data in PUS(1,1) report - Telecommand Acceptance Report - Success.	type=1 AND subType=1	TM-PUS-1-1-AckSuccess	32	32
2	ackFailure	Data in PUS(1,2) report - Telecommand Acceptance Report - Failure.	type=1 AND subType=2	TM-PUS-1-2-AckFailure	40	40
…	…	…	…	…	…	…
11	connectionReport	Data in PUS(17,2) report - Link Connection Report.	type=17 AND subType=2	TM-PUS-17-2-LinkConnectionReport	0	0

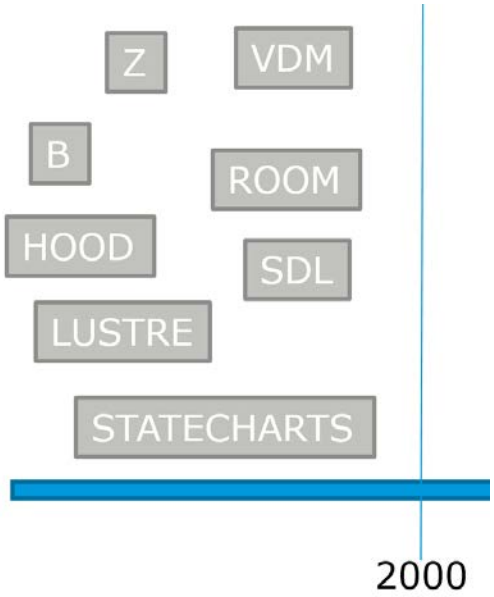
Before 2000



The “golden age” of formal methods, proof engines, visual modelling

Major commercial tools all included a model checker (Opengeode, Statemate, SDT)

Before 2000



But… the need was not understood

Code generation was hardly supported

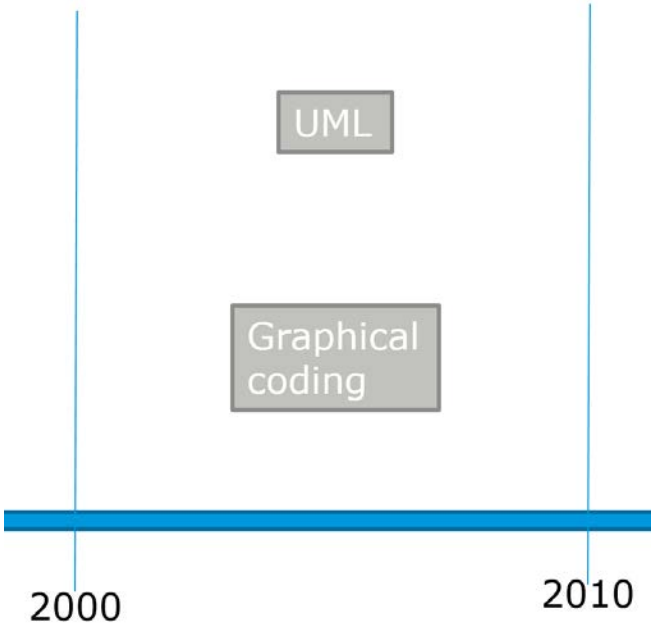
Embedded platforms allowed only tiny applications to run (1 MHz CPU!)

Tools were expensive and model checkers required a lot of memory to run

What happened then…

Most tool vendors gave up and sold business

In the 2000s, another attempt



One informal language to rule them all.

Syntax and semantics do not matter anymore

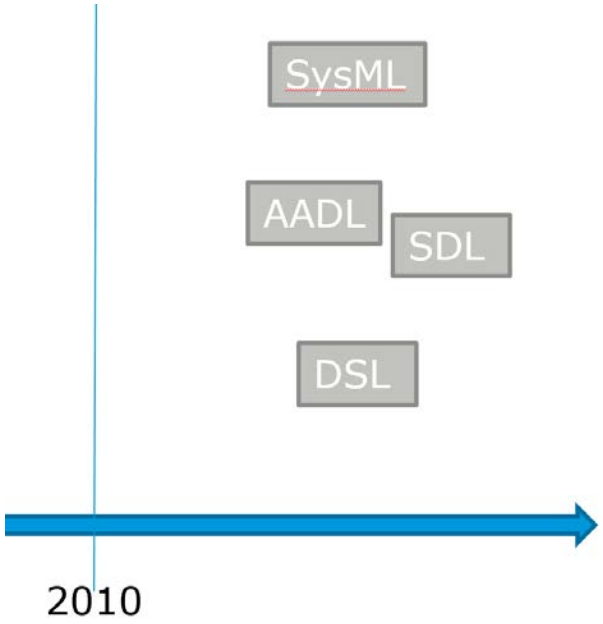
Methodology and process come with the tools, not with the language

“Cheap” tools and heavy marketing

What happened then…

It failed, too.

Since 2010: learn from the mistakes?



Make collaborative, open-source tools

Mix formal and informal languages

Address system and software altogether

Think long term

Use on real, large scale projects

What will happen next? A vision for the future

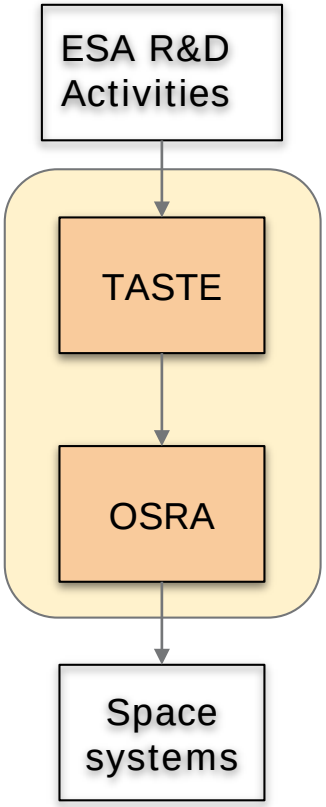
Objective: **to have within 5 years a large adoption of MBSE in all new ESA projects**

- Executable specifications (*state machines in the loop)*
- Data models (*no more manual ICDs)*
- Automatic code generation
- Updated ECSS standards to support this evolution….

In parallel, R&D shall focus on **more advanced use of models**

- Proof of system properties
- Optimization of resources
- AI to ease the production of the “right system”

Current state of the art



Explore languages, operating systems, avionics, compilers, … based on the needs from projects.

Technology placeholder forming the state of the art in MBSE methods and tools

Define a software architecture and component model tailored for the space domain

TASTE TECHNOLOGY PLACEHOLDER

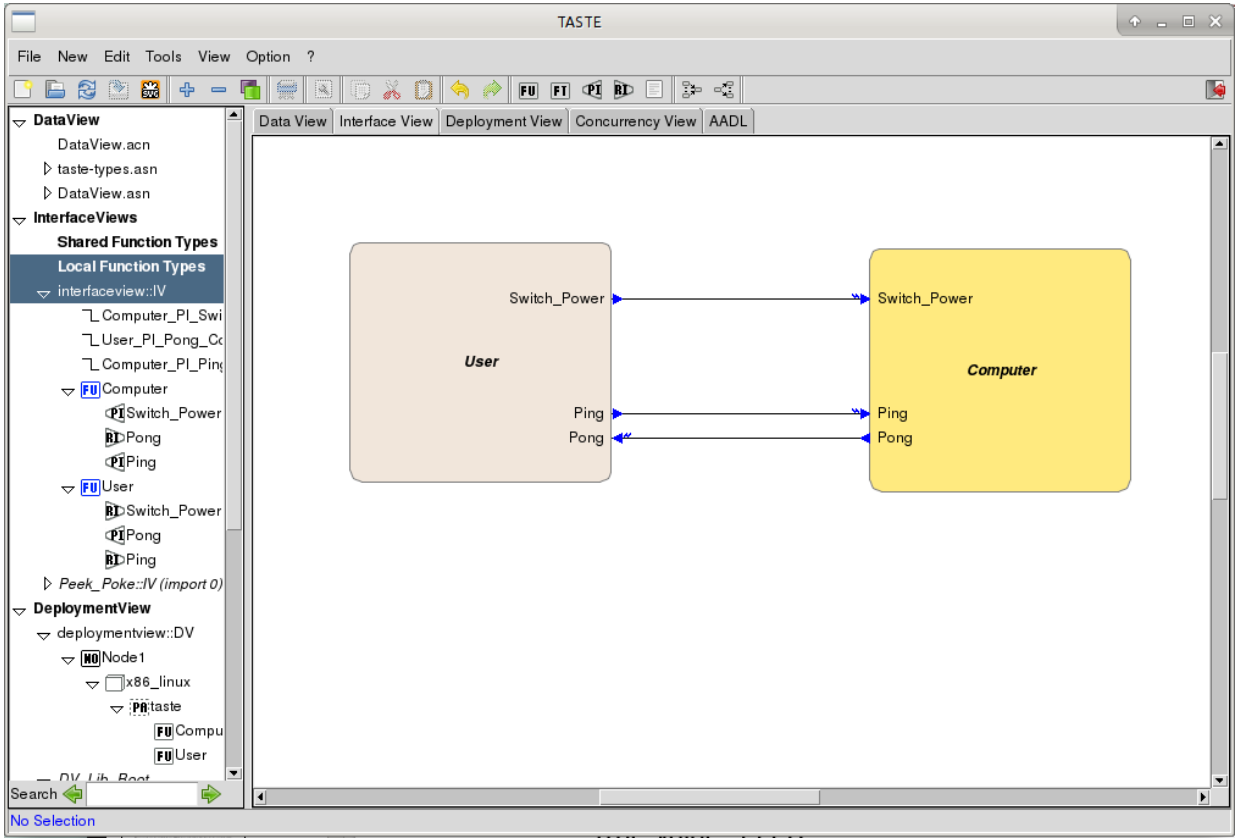
TASTE is putting together the ingredients to support technology exploration

It provides the bricks required by OSRA to disseminate MBSE to operational projects:

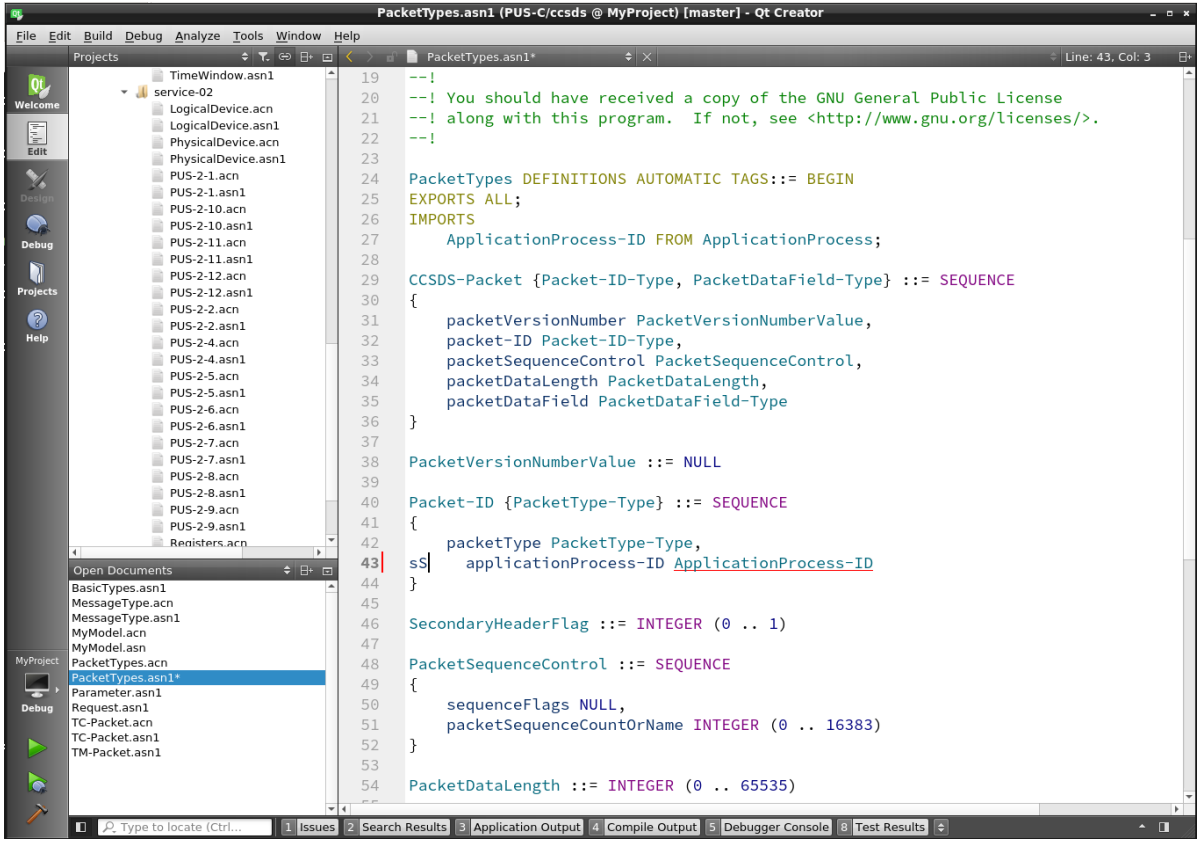
- Formal description techniques and modelling languages
- Architecture, data, and behaviour modelling
- Automation
- Clear, unambiguous steps to reach the desired results

The TASTE/OSRA ecosystem favors free and open-source tools

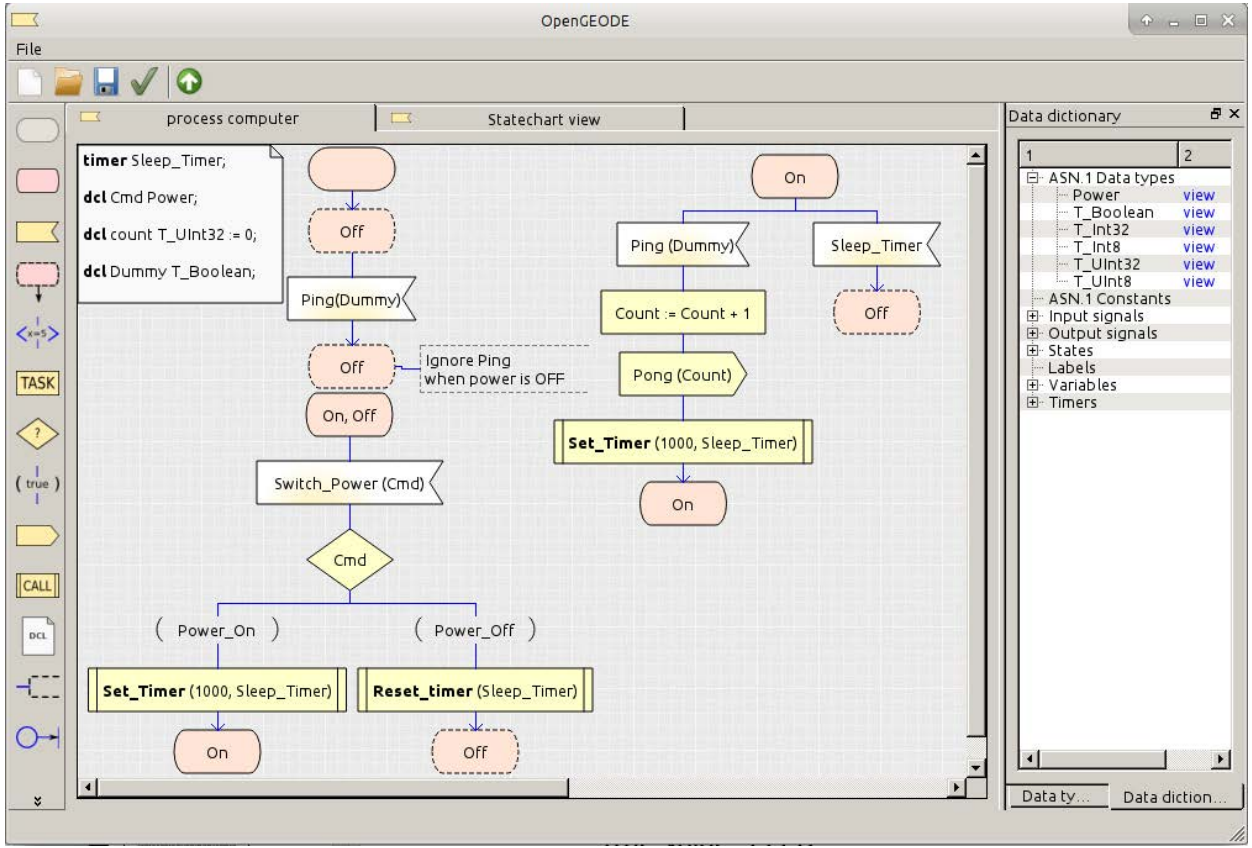
Step 1: logical architecture (AADL)



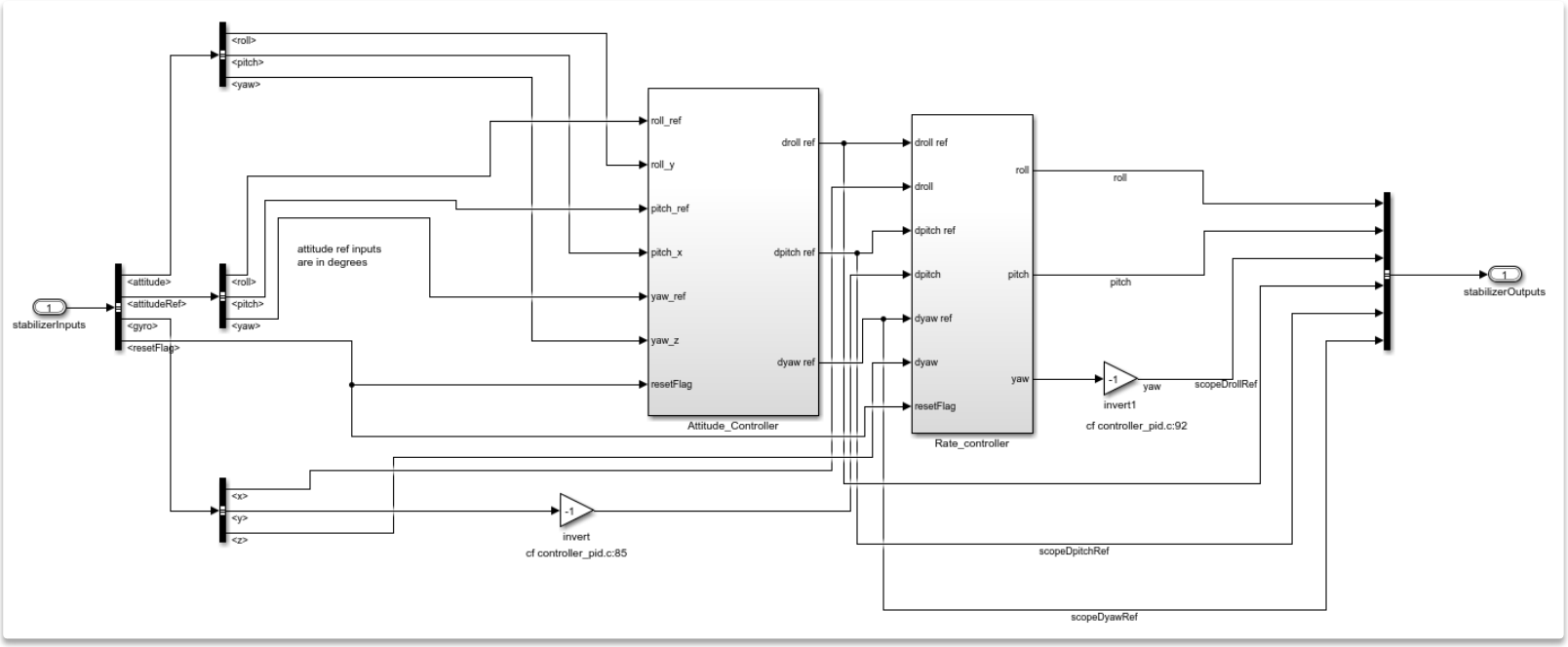
Step 2: Define all data structures (ASN.1)



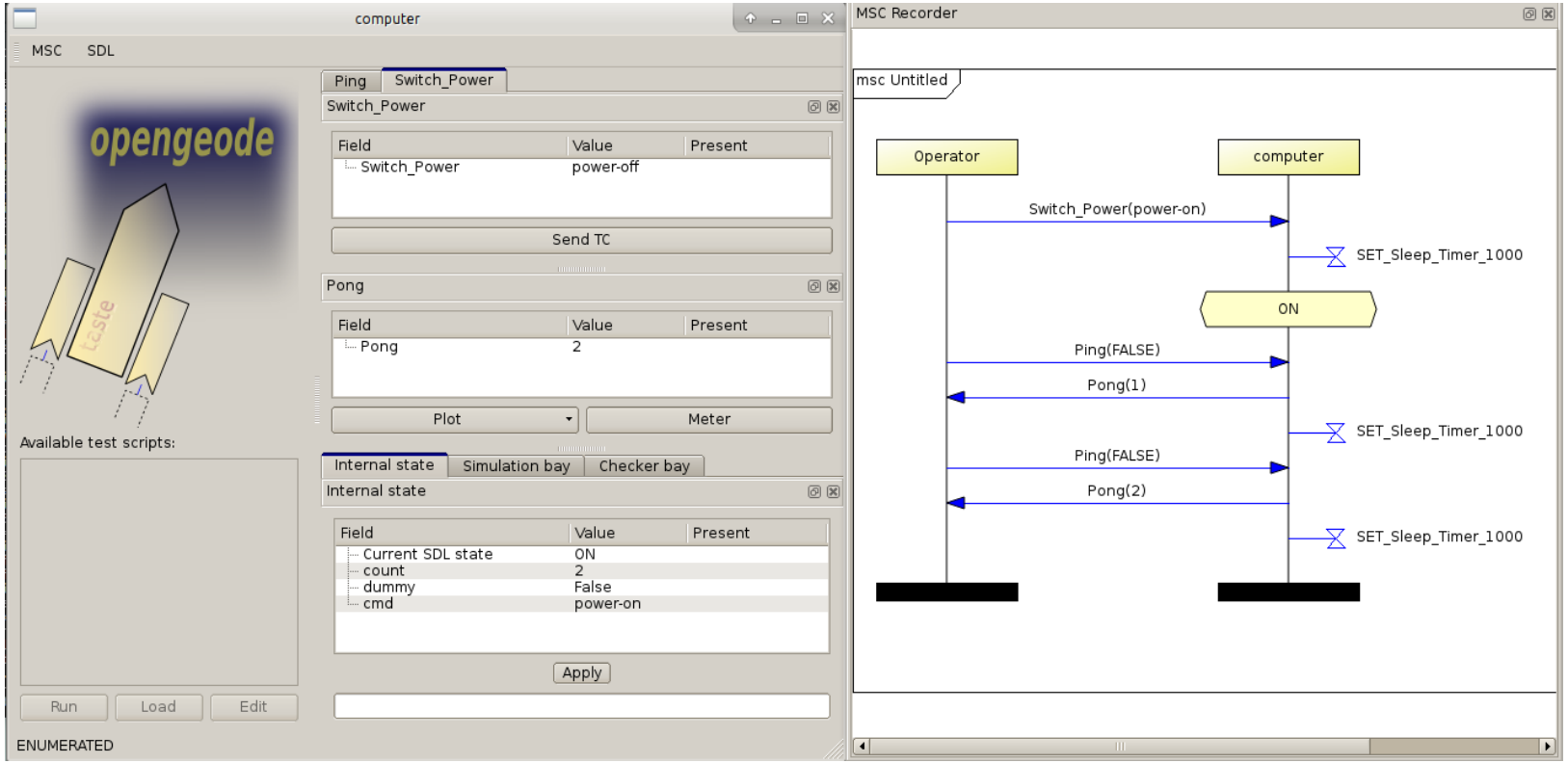
Step 3: Model the behavior (SDL)



Step 4: Model or code the algorithms (Simulink)



Step 5: Simulate and check properties

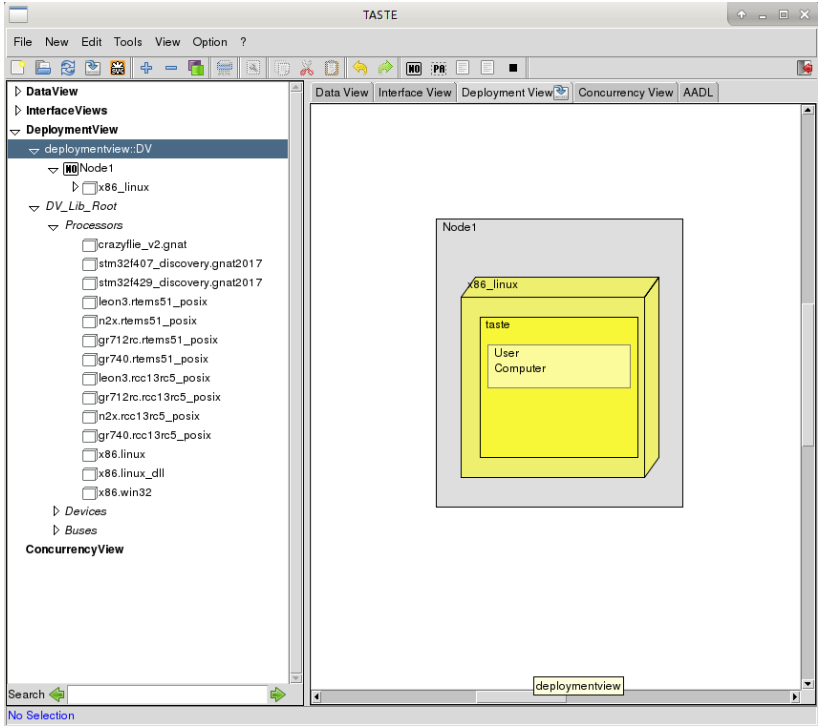


Step 6: Generate or write test cases, and run them

```
42  @Scenario
43  def Exercise_user(queue):  # queue is actually an instance of the Scenario class
44      '''user processing'''
45      queue.sendMsg('Switch_Power', 'power-on', lineNo=89)
46      time.sleep(0.5)
47      queue.sendMsg('Ping', 'FALSE', lineNo=86)
48      try:
49          queue.expectMsg('Pong', '1', lineNo=94, ignoreOther=False)
50      except TypeError as err:
51          raise
52      time.sleep(1.5)
53      queue.sendMsg('Ping', 'FALSE', lineNo=86)
54      (msgId, val) = queue.getNextMsg(timeout=10)
55      if msgId == 'Pong':
56          print ("Something went wrong, Pong was not expected ")
57      return 0
58
```

```
taste@home:~/my_project/binary.c/binaries/user-GUI$ ./runtest.py
Opening msgQ: 1001_user_PI_Python_queue
 [INFO] Exercise_user - Starting scenario
[INFO] Exercise_user - Sending Switch_Power
[INFO] Exercise_user - Sending Ping
[INFO] Exercise_user - Waiting for Pong(1)
[INFO] Exercise_user - Received and verified message content, all OK
[INFO] Exercise_user - Sending Ping
[INFO] Exercise_user - Waiting for the next message
[ERROR] Exercise_user - Timeout expired
```

Step 7: Deploy on a target



In reality...

... Has someone ever used all (or any of) that?

ESA operational project using MBSE

proba-3

5.3.6.2 Telemetry

This module contains all types specific for ASW Telemetry.

Table 13 - ASW-TM data structure definition.

ASW-TM (SEQUENCE)						Min: 20 bytes	Max: 1051 bytes
Structure representing all telemetry sent by CI ASW.							
No	Field	Comment	Present	Type	Constraint	Min Bits	Max Bits
1	packetHeader	Packet header as defined in [RD4].	always	TM-PacketHeader	N.A.	48	48
2	packetDataField	Packet data as defined in [RD4] with ASW specific contents.	always	ASW-TM-PacketDataField	N.A.	112	8360

Table 14 - ASW-TM-PacketDataField data structure definition.

ASW-TM-PacketDataField (SEQUENCE)						Min: 14 bytes	Max: 1045 bytes
Structure representing data field of all telemetry generated by CI ASW.							
No	Field	Comment	Present	Type	Constraint	Min Bits	Max Bits
1	dataFieldHeader	Data field header as defined in [RD4].	always	TM-DataFieldHeader	N.A.	96	96
2	sourceData	Data specific for service report repeesented by current TM.	always	ASW-TM-SourceData	N.A.	0	8248
3	packetErrorControl	Checksum calculated as defined in Annex A.2 [RD4].	always	Uint16	N.A.	16	16

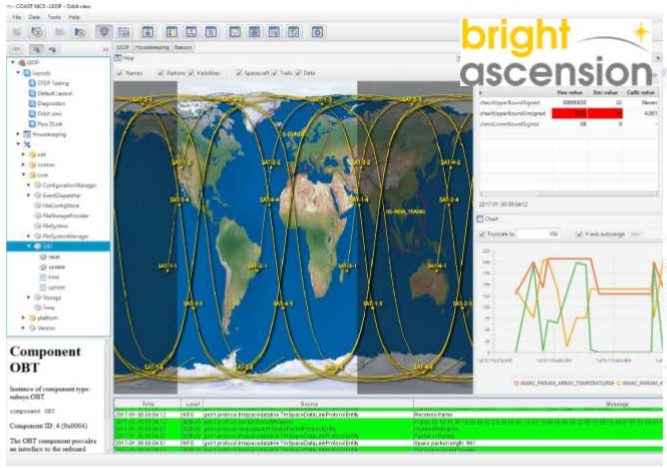
PUS-C implementation on the payload with TASTE

Code, Tests and 300 pages of ICD generated automatically

Flight and ground segments

n7 SPACE

Other (non-ESA) projects



ThalesAlenia Space

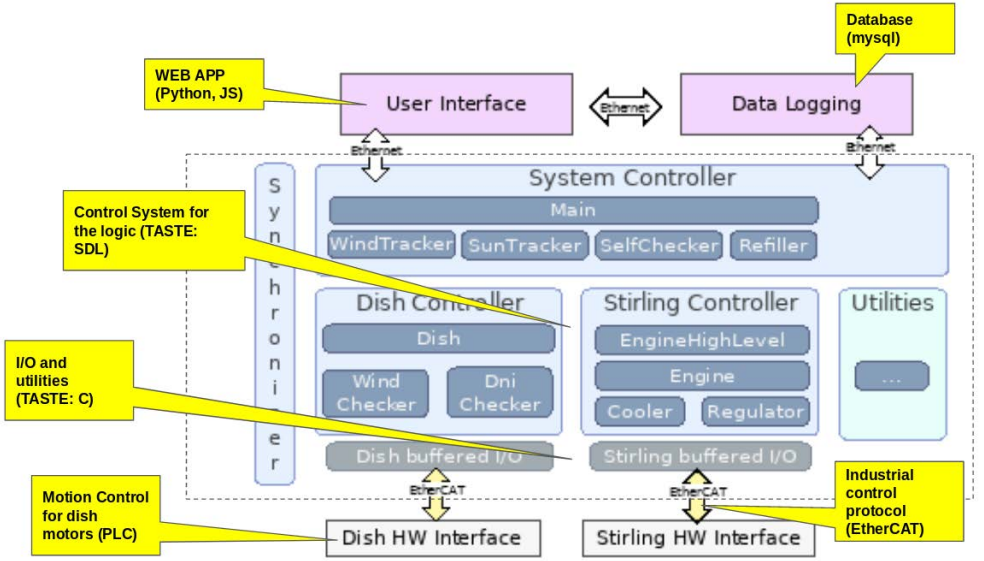
Thales

Other (non-ESA) projects

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH
Robotics Innovation Center

gmv
INNOVATING SOLUTIONS

Other (large) non-ESA projects

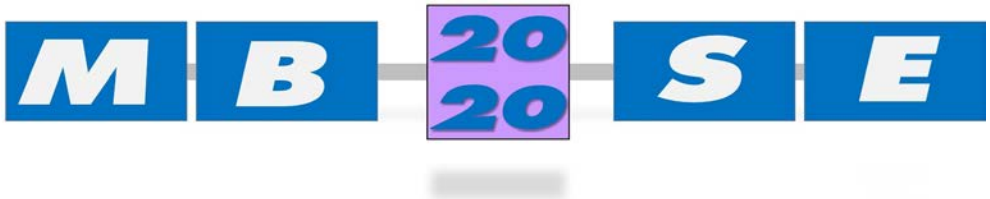


FONDAZIONE BRUNO KESSLER

The future

- The TASTE Steering Committee kick off meeting is this afternoon
  - with FBK, TAS, Airbus, DFKI, N7S, GMV
- The MB4SE working group

We shall all meet again for the



conference in September next year in ESTEC

Outline – Questions?

1. What is MBSE ?
2. History : some background on modelling technologies
3. A vision for the future
4. Current state of the art (TASTE, OSRA)
5. Use in projects
6. Conclusion and the short-term future

taste
The Assert Set of Tools for Engineering

Visit
https://taste.tools

Conclusion

1. MBSE is triggering a huge interest for future space programmes

2. ESA has the vision and can ensure consistency and long-term support

3. Industry and community involvement are essential to continue