Model-based System and Software Engineering – Onboard Software

ESTEC, 8 December 2016

A. Jung, M. Verhoef, M. Perrotin (TEC-SWE)

Purpose of this workshop (reminder)

1. Share our vision on MBSE, with focus on COMPASS, TASTE and OSRA
2. Showcase experiences gained from MBSSE applications in on-going projects
3. Discuss the potential alignment with other model-based technologies
4. Identify opportunities for further collaboration, harmonization and consolidation
5. Identify next steps for technology exploitation and R&D

European Space Agency

Programme for the afternoon

This session is focus on model-based software engineering

Four solicited talks (20 min each):

1. Data Modelling on Proba3 ASPIICS payload (K. Grochowski, N7 Mobile)
2. Mixing Re-Use and Model-Based Development (A. Pasetti, PnP Software)
3. Space Automation & Robotics General cONtroller (M. Azkarate, ESA)
4. MBD of an Energy-System Embedded Controller using TASTE (R. Cavada, FBK)

Four elevator pitches (5 min each):

- Model-based design and tools for space applications (Holger Schlingloff, FhG-FOKUS)
- A model-model-based and domain-specific development environment (Andreas Wortmann, OHB)
- How dynamic is TASTE? (Roger Jacobs, Topic)
- Qgen as a qualified code generation backend for TASTE (Andres Toom, IBKrates)

Discussion session and round-up

Context

Model-based software development is about software

How to make better software ?

Do we need to make better flight software ?

Trends in desktop / web development
- new languages (Rust, Swift, F#, Scala, etc.)
- functional programming
- static code analysis, better compilers
- no rush towards MBSE

But embedded systems are different
- multi-threaded and distributed software
- heterogeneous communication, little resources
- high cost, critical in case of failure

Process and MBSE

Well established processes (ECSS) focused on milestones and quality assurance

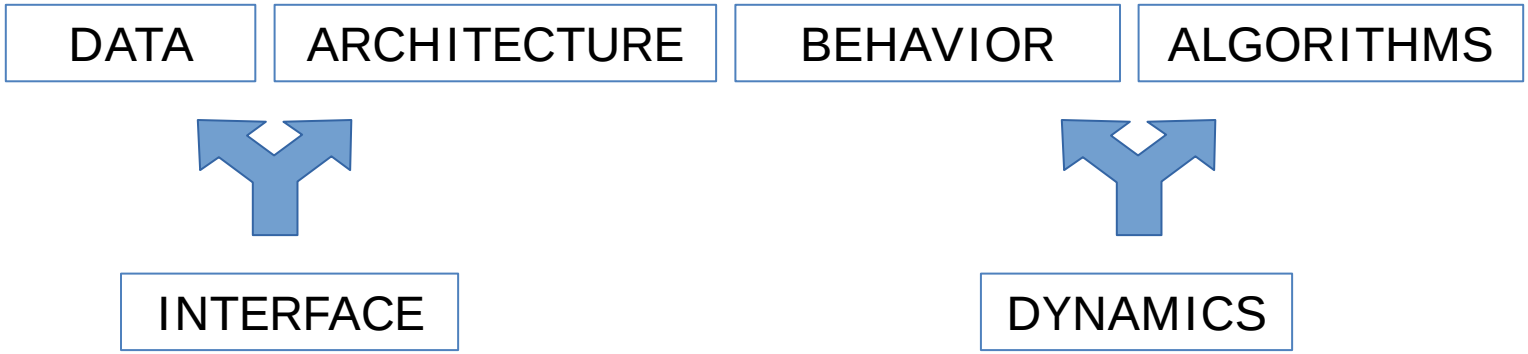
But the process is not supported by tools….

MBSE aims at filling this gap to get

1. Better specifications

1. Robust, flexible, well-documented, maintainable design and code

The key is to master the system's



Where we want to go

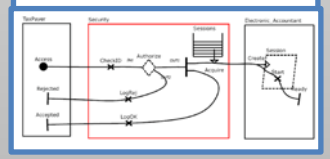
- Software correctness by construction

from	*to*	*Using methods and tools*
TEXT-BASED specifications	EXECUTABLE specifications	DATA modelling
AMBIGUOUS drawings	AUTOMATIC documentation, testing, coding	STATE MACHINES
MANUAL coding	*and ultimately*	MODEL checking
EXPENSIVE testing	BUG-FREE software	QUALIFIED code generators
CONTINUOUS patching		

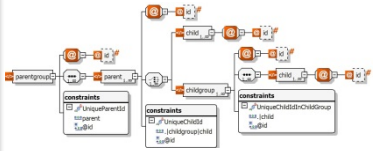
LANDSCAPE

SPECIFICATION LANGUAGES

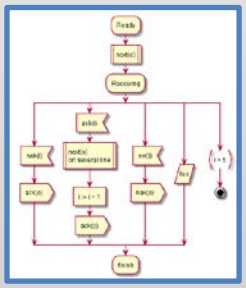
USE CASE MAPS

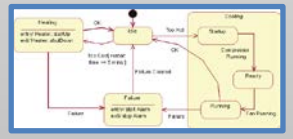


DATA MODELLING



STATE MACHINES





V&V

SPIN

MAST

CHEDDAR

ARCHITECTURE CODE

OCARINA

Some constraints

Existing software, applicable standards, in-house processes

ECSS E40, Q80, E70 (PUS), RTEMS, Ada runtimes, … must fit

Not any single all-in-one solution with batteries included!

UML, Simulink, SDL, AADL, … many partial solutions for a tiny market

Not many commercial solutions survive on the long term…

Remember Rose, Rhapsody, ObjectGEODE, Statemate, …?

Go for free, open-source solutions only ?

Yes, but this is not an absolute guarantee of success

Go for existing languages or invent DSL just for us?

DSL are attractive, but think of the long term maintenance and support

WORKING TOGETHER ON A SOLUTION IS ESSENTIAL. NOBODY CAN GO ALONE

What we have done so far

taste
The Assert Set of Tools for Engineering

Model-based software development

SAVOIR OSRA
on-board software reference architecture

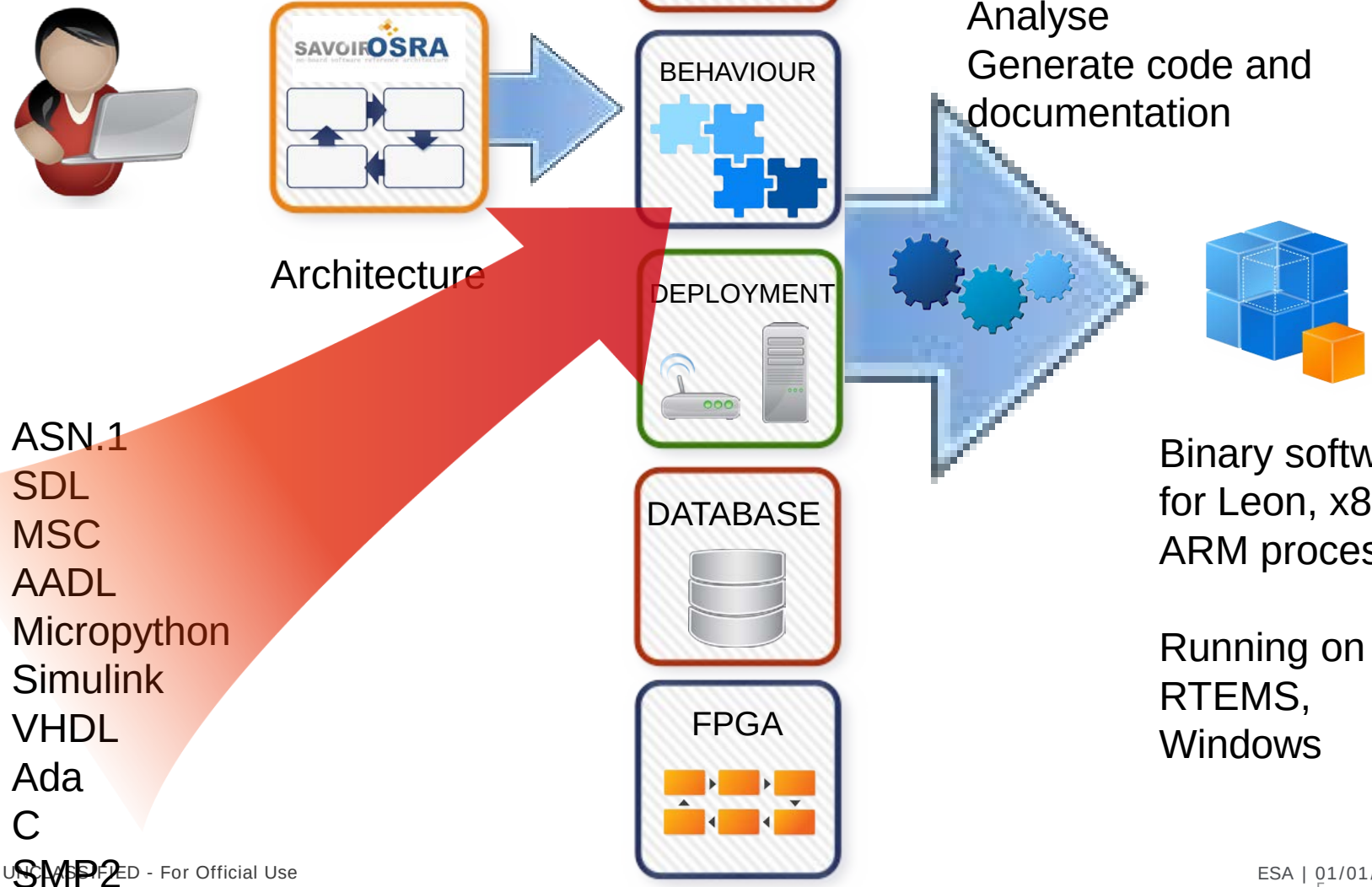
OSRA
Space Component Model

COMPASS

Dependability and model checking

European Space Agency

The global picture



How to get these tools?

TASTE is an open-source project

Get it from ESA website: https://essr.esa.int or https://taste.tools

OSRA is available as a CFI from ESA

Request a license for free

COMPASS is available here:

http://www.compass-toolset.org

What next?

On the technical side, there is still a lot to explore…

- Use case maps
- Meaningful document generation
- Usable model checking
- Full hardware-software co-engineering
- System-Software co-engineering
- Qualification of tools
- More formal methods
- Education and training material
- …..

Eventually covered by our R&D programme

But that is just the first part… The real difficulty is to **find users**!

- How to raise interest?
- How to get feedback?
- How to disseminate to real projects?

Programme for the afternoon

This session is focus on model-based software engineering

Four solicited talks (20 min each):

1. Data Modelling on Proba3 ASPIICS payload (K. Grochowski, N7 Mobile)
2. Mixing Re-Use and Model-Based Development (A. Pasetti, PnP Software)
3. Space Automation & Robotics General cONtroller (M. Azkarate, ESA)
4. MBD of an Energy-System Embedded Controller using TASTE (R. Cavada, FBK)

Four elevator pitches (5 min each):

- Model-based design and tools for space applications (Holger Schlingloff, FhG-FOKUS)
- A model-model-based and domain-specific development environment (Andreas Wortmann, OHB)
- How dynamic is TASTE? (Roger Jacobs, Topic)
- Qgen as a qualified code generation backend for TASTE (Andres Toom, IBKrates)

Discussion session and round-up