Model-Based Design of an Energy-System Embedded Controller using TASTE

Roberto Cavada[1], Alessandro Cimatti[1], Luigi Crema[2], Mattia Roccabruna[2] and Stefano Tonetta[1]

[1]Embedded Systems Unit
Fondazione Bruno Kessler

[2]Applied Research on Energy Systems
Fondazione Bruno Kessler

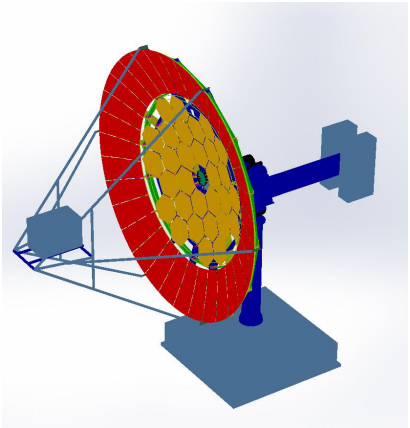
MBSSE Workshop, ESA-ESTEC, Dec 8 2016

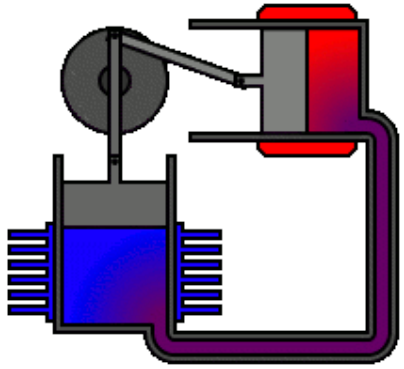
Outline

The *Contest* project

Goal: efficient energy co-generation from Concentrated Solar Power, for domestic/industrial usage

- Combines a large *Solar Collector* and a *Stirling Engine*
- An heterogeneous, large, critical, complex system





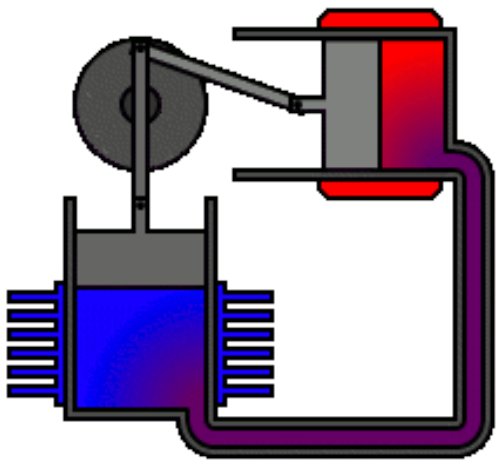
Credits: *Zephyris at the English Wikipedia*

The Solar Collector

- Dish: 8.5m diameter, 5 tons structure
- Concentration factor: 3k
- Movement precision is relevant: $\leq 0.1^\circ$ when tracking Sun

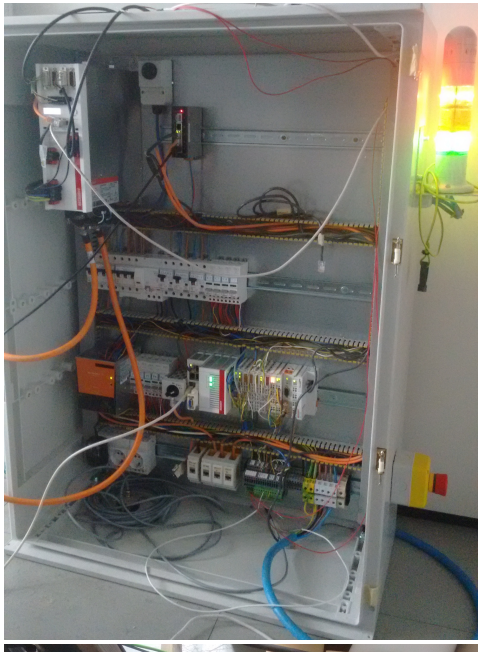
The Stirling Engine

- Heat to rotation, via cyclic compression/expansion of He
- Regulated by changing He pressure P [20..200 bar]
- Stirling temperature T depends on heat received and on P
- Efficiency is proportional to (P, T),
  but higher P cause higher drop of T



Credits: *Zephyris at the English Wikipedia*

The plant: some pictures



Safety

Critical factors

- T and P can vary at high rate
- When fed, Stirling can melt in 2 seconds ($T > 2000\,^{\circ}\mathrm{C}$)
- When not fed, Stirling freezes in 2 seconds ($T < -100\,^{\circ}\mathrm{C}$)
- Many fault sources

System requirements and specifications

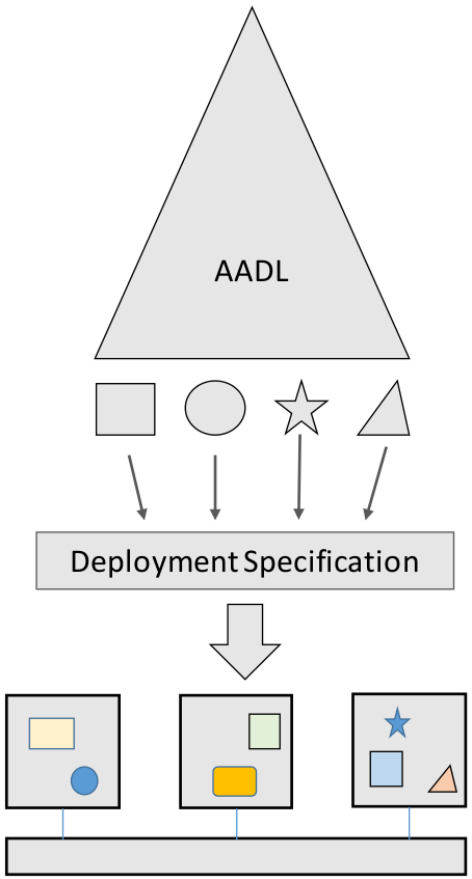
- About 200, in natural language. For example:

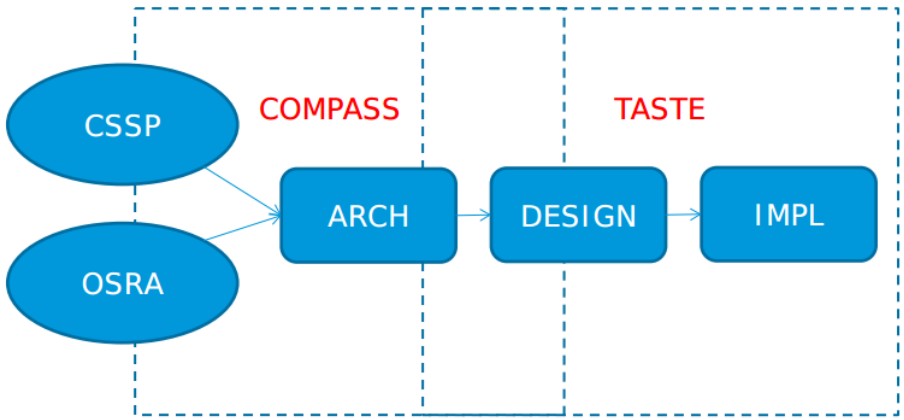
  "When rpm $\geq$ RPM_FS_CONS, after OIL_PRESS_CHECK_TO sec if oil_press $<$ MIN_OIL_PRESS, Engine shall stop with Immediate Procedure and go to Error mode."

- Most in form "p always holds", "p always triggers response q within a time bound"

The TASTE tool

The ASSERT Set of Tools for Engineering http://taste.tuxfamily.org/





Credits: *ESA*

TASTE: features we exploited

http://taste.tuxfamily.org/

- System design: AADL functional blocks (*Interface View* editor)
- Data types: ASN.1 notation used throughout the models
- Behavior design: *SDL* (*Opengeode*)
- Integration glue: *C* code
- Deployment: Xenomai, Intel x86 (*Deployment View* editor)

TASTE: features we exploited

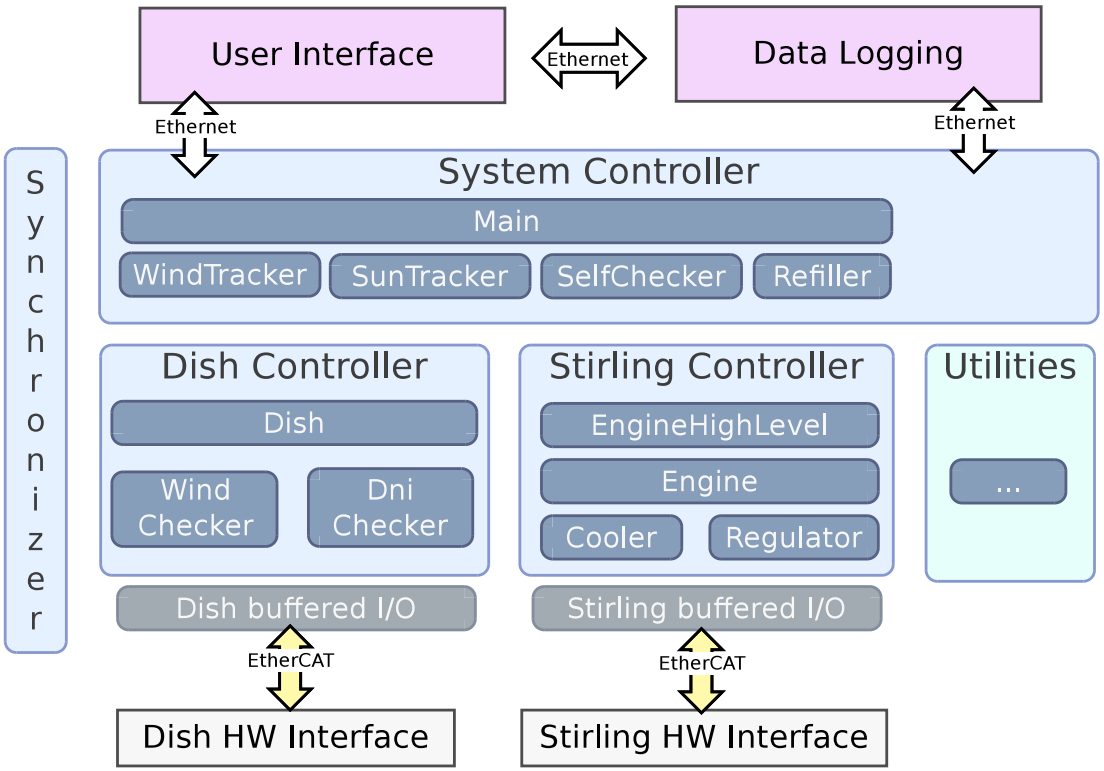
http://taste.tuxfamily.org/

- System design: AADL functional blocks (*Interface View* editor)
- Data types: ASN.1 notation used throughout the models
- Behavior design: *SDL* (*Opengeode*)
- Integration glue: *C* code
- Deployment: Xenomai, Intel x86 (*Deployment View* editor)

- Generation of a *database* for logging/status dumping
- Generation of a *GUI*: used for testing
- Generation of a Python *API*: filling the DB, communicating with the User Interface, scripting

Model of the Control System

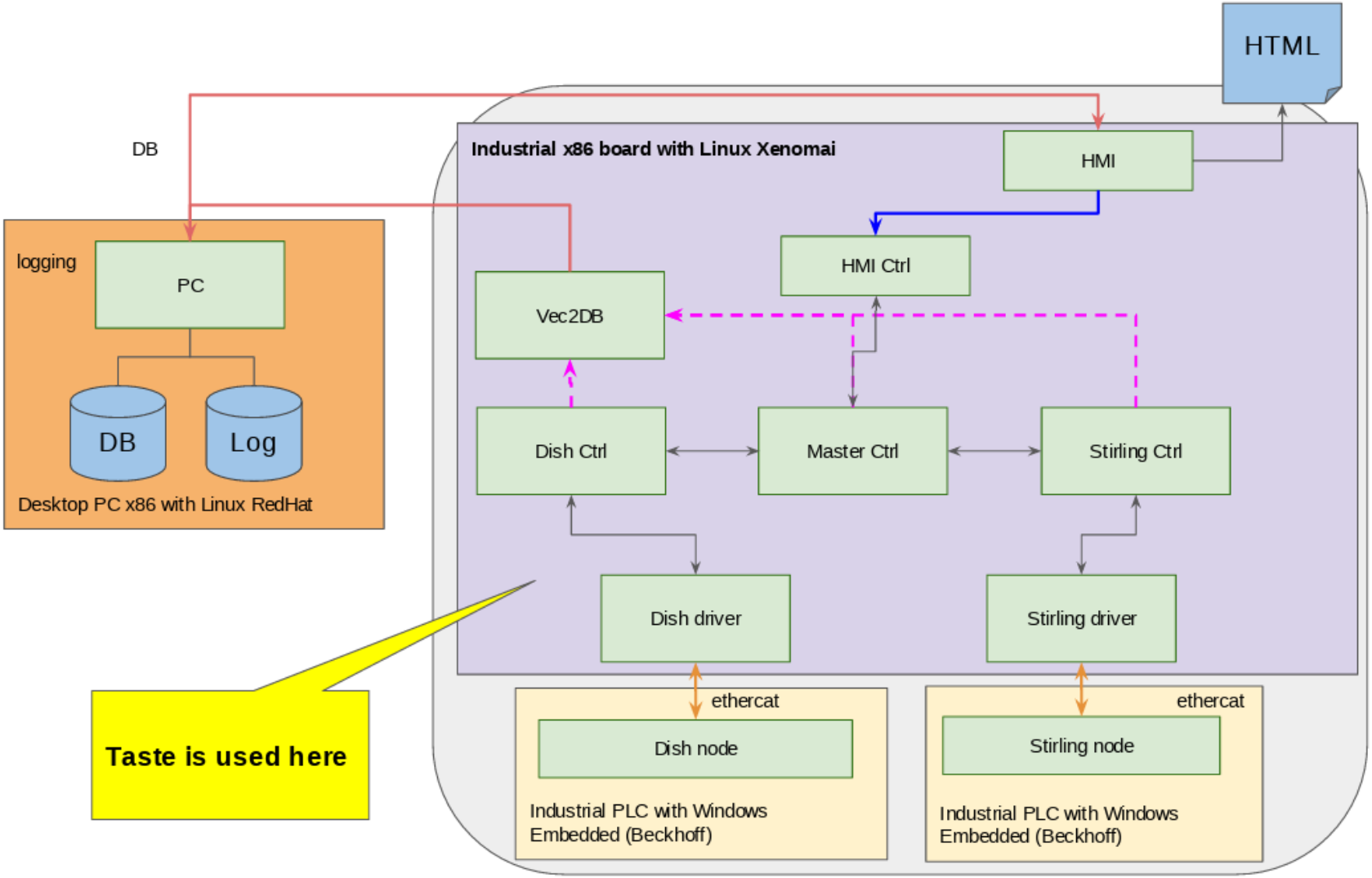
See https://gitlab.fbk.eu/ITC4Energy/contest



- 12 SDL blocks, FSMs have 86 locations and 175 transitions (not counting self-loops). Use 12 distinct timers
- Most blocks are *on/off* type. Regulator is *prop/der*. In Sun tracker uses *prop* to adjust Sun position errors

The *Contest* physical architecture

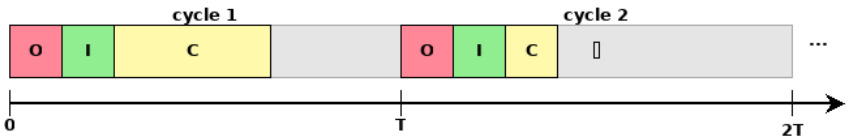
TASTE and heterogeneous systems



Interesting issues in the model

Some issues from the notion of delta-cycle loop, with an asynchronous control system

- We wanted delta-cycles to be sequence of *O*utputs, *I*nputs, *C*ontrol



- We forced O/I to run before C, adding a coordinating block and introducing a *start_cycle* event
- We need to assure that C runs to completion each cycle
- Timers are forced to expire at $t_{curr} + T_{ck}$, $T_{ck} = kT, k \in \mathbb{N}$
- Empirical data show a 2ms bound for the $O/I/C$ triplet

What we have learnt

TASTE performances

- TASTE effectively support deployment of complex ES, and creates efficient code: 76 RT tasks, 0.2% CPU usage, 14 MB
- It cannot be used if micro seconds are required: target 100 μs is desirable
- Some usability issues with large models
- Building system is slow

What we have learnt (2)

Modeling in SDL

Modeling in SDL is terrific but requires active discipline. We wished TASTE supported:

- Model navigation (e.g. *which inputs is (not) accepted in each state?*)
- Block instances (with templates?)
- Local definition of ASN.1 types
- Extended loops (e.g. seq2 := [f(x,y) for x, y in seq1 if g(x,y)])
- Signal *Fan-Out* > 1

Conclusions

- We applied *MBD* to a complex and heterogeneous system
- For modeling and deployment we successfully used TASTE:
  - Generates efficiently code, for relatively slow control systems
  - Robust editor, with some limits as the model grows
  - SDL modeling support may be improved
  - Some bugs identified (were fixed by TASTE's team)
  - **ARM support highly desired**, to target general industrial domains

Future work

- Validate more aggressively with testing (supported by TASTE)
- Run schedulability analysis (supported by TASTE)
- Apply FM: model checking, compositional reasoning, fault-extension and failure analysis

Thanks!

- For your attention
- To the TASTE's team for the great support they provided

Formal Methods in Industry?

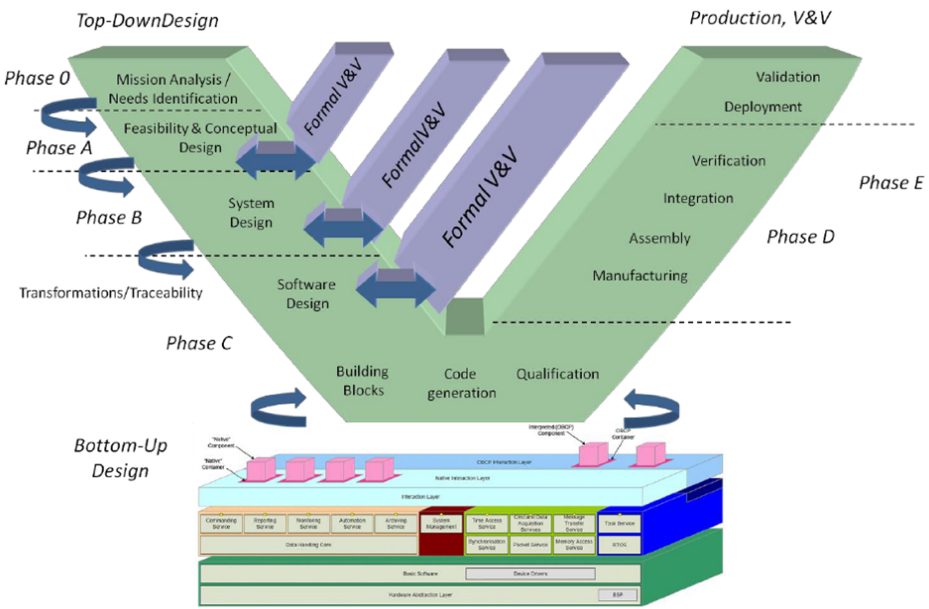
(Our vision at FBK)

Merge into the process, go beyond formal verification

- Untracked, wrong, ambiguous requirements and specifications: *Requirements Analysis*
- Need to tackle non-nominal conditions: *Safety Analysis*
- Stepwise approach needed: *Contract Based Design and Verification*
- Need Fault Detection and Isolation: apply dedicated techniques

TASTE

- TASTE is a Modeling tool made by ESA, devoted to deployment in ES
- In the design workflow, TASTE is collocated after the COMPASS toolset, which targets Model-based Dependability Engineering.
- TASTE and COMPASS share many concepts and specification language (AADL), but they are isolated



Taste Languages

AADL To describe the architecture in terms of function blocks, their input/output ports, and port connections

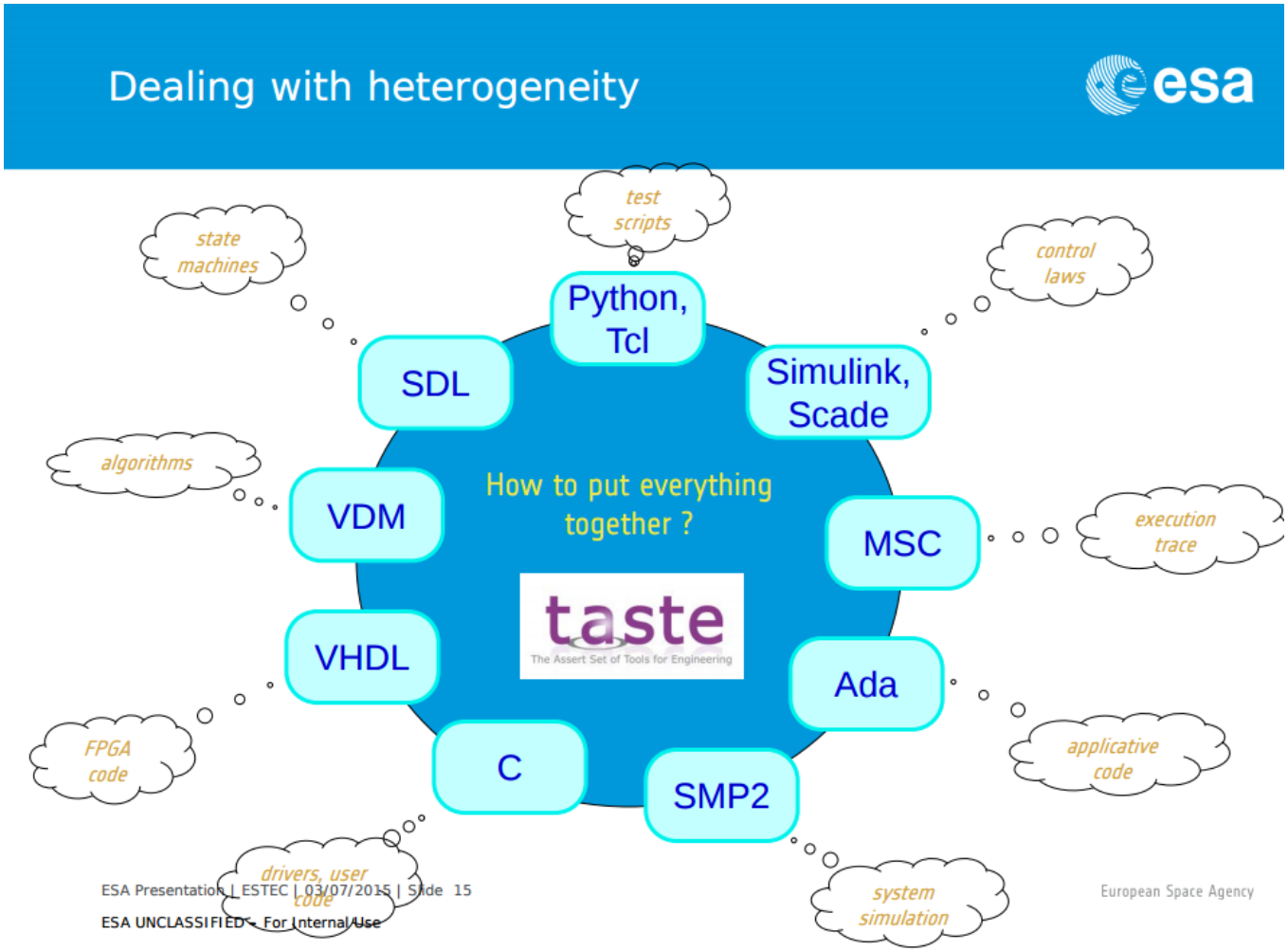
ASN.1 Standard notation to specify data types, along with their constraints and encoding. Types and values are then available at system and behavior level

SDL Formal specification language used in TASTE for modeling the behavior of the functional blocks

- FSMs described in SDL communicate asynchronously through events queued in channels
- Each FSM runs-to-completion, until stops waiting an event or timer

C/Ada For addressing low-level

Taste Languages (global vision)



Credits: *ESA*

TASTE pros against other tools

- Openness: TASTE is Open Source
- Produced by ESA, which is FBK's partner in several projects
- Very interesting research area to create links and continuity between COMPASS and TASTE
- Free as "Free beer"

Still, TASTE is a research tool, not currently qualified for safety critical domains. Estimated TRL: 4/5

Short tale of *MBD* at the ES Unit in FBK

- Until 2015, we addressed system V&V mostly at *engineering level*: COMPASS (ESA, FBK, RWTH Aachen) and FoReVer (ESA, FBK, Intecs, TAS) are good examples
- Our Unit was lacking the experience to cover the *deployment* stage
- In 2016 we joined efforts with the *Applied Research on Energy Systems* Unit in FBK, to develop innovative, complex and *safe* energy systems
- We searched for tools support, we jumped at TASTE as: OS, strengthening ESA partnership, willing to address industry, very interesting research area to create links and continuity between COMPASS and TASTE

This is the report of activities yielded by fortunate opening: ES Unit applying *MBD* to the *Contest* project, while learning and evaluating TASTE on the job.