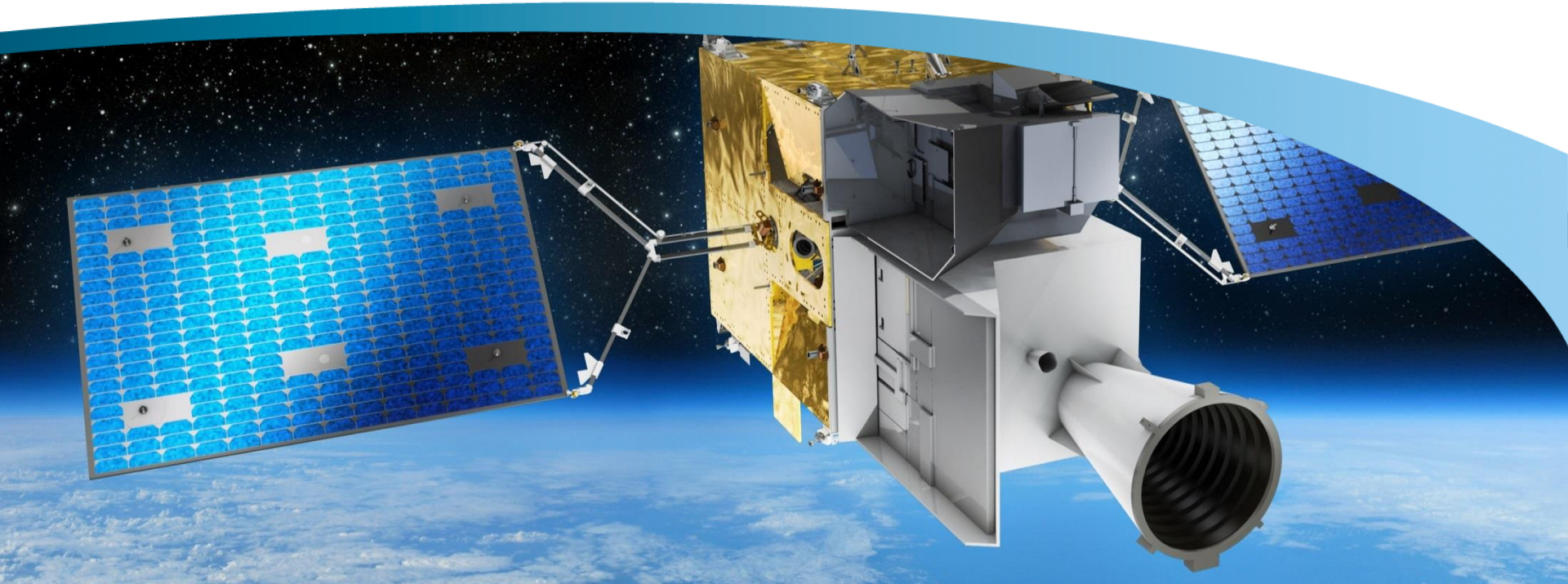


OHB System AG
Andreas Wortmann
08.12.2016, MBSSE Future Directions



SPACE SYSTEMS

A Model Based and Domain Specific Development Environment

We. Create. Space.

Agenda

- What are the Needs?
- What do we do?
- A tool to consider ...

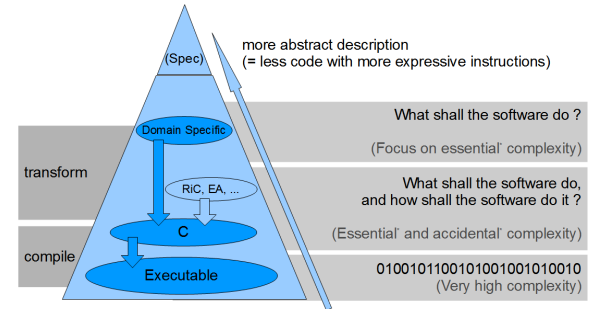
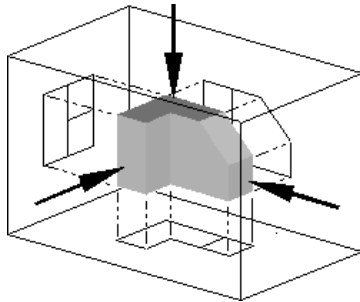
Needs in System and Software Engineering

- Faster, Later, Softer
 - ... increase productivity ...
 - ... increase reactivity ...
 - ... increase flexibility ...

- Where do we come from ?
 - Paper-based Engineering (requirements, hardware, interfaces ...)
 - Programming (C, Ada, VHDL, Verilog ...)

- What do we do to improve ?
 - Model Based Engineering
 - Domain Specific Engineering

Methodologies & Techniques



* Essential and accidental complexity as identified by Frederick P. Brooks in 1986, 'No Silver Bullet - Essence and Accidents of Software Engineering'

Model Based Engineering

- (Abstract) description of the system
- Consistency (Single Source Paradigm)
- Different Viewpoints
- Early Analyses (Formal Checks)
- Early Tradeoffs
- Code, Document Generation
- graphical CASE tools
- General Purpose Languages (UML, SysML ...)

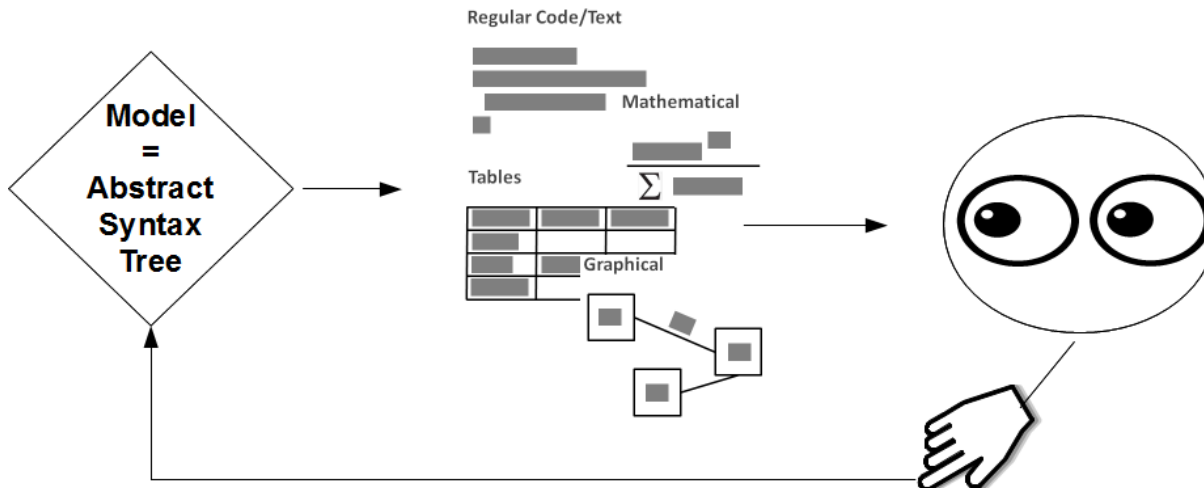
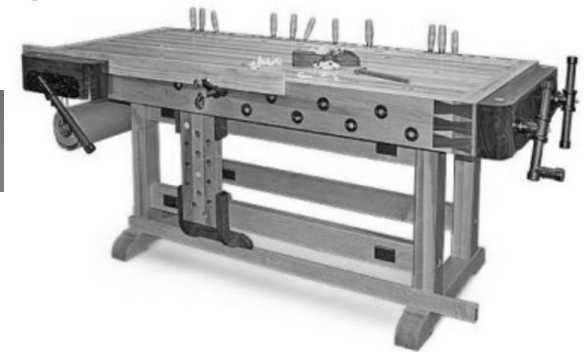
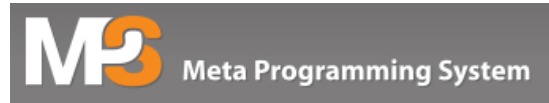
Domain Specific Engineering

- Raise level of abstraction
- More efficient description of a solution/design
- Shift domain knowledge into tool and languages (not everything can be expressed in any tool/language)
- Domain Specific Languages (AADL, ASN.1 + ACN, SDL ...)



Implement models using Domain Specific Languages

- Composable models based on a set of Meta-Models (= Languages)
- Language Workbench (IDE for Language Engineering)
 - JetBrains MPS
 - Open Source
 - www.jetbrains.com/mps
- Projectional Editing (no parser needed)



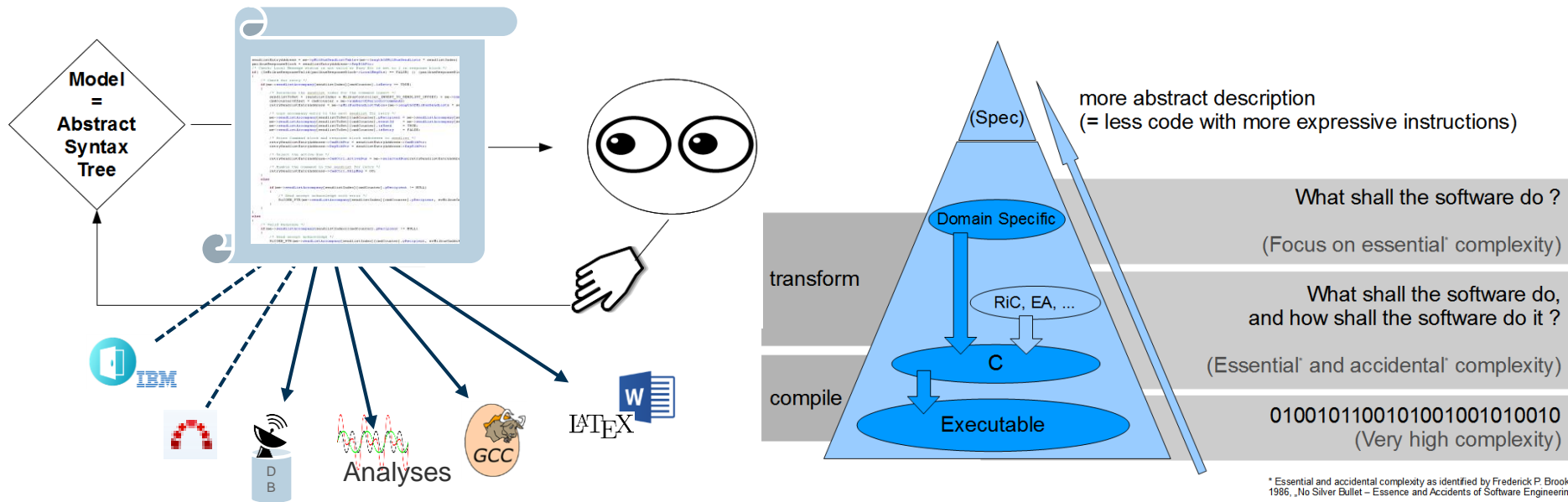
Languages are composable
Models “live” is a common habitat

- Integrate well with each
- May refer each other
- Can extend each other

Extend the model to cover the C language



- C Language is implemented with a model based technology
 - A Domain Specific Language that looks like C and behaves like C
 - Is extended with higher-level abstractions: state machines, physical units ...
 - Is extended with space domain specific aspects: PUS, FDIR ... (see DASIA 2016 paper)
 - Is extended with analyses and checks: MISRA, variable initialization, ..., schedulability ...
- Blur the border between modeling and implementation **(Model == Implementation)**



* Essential and accidental complexity as identified by Frederick P. Brooks in 1996, „No Silver Bullet – Essence and Accidents of Software Engineering“

Benefits – some examples

- Ground Station Configuration: 100% generated (no extra effort, correct by construction)
- Computational Model for Schedulability Analysis
- Efficient Programming: PUS 3 Housekeeping Service → implemented in a few hours
- Different targets, same source: Leon2+RTEMS (Satellite), Intel+Linux (Desktop PC)

Activity GenerateOneShotHousekeepingReports with Numeric Id 27 is commandable by TC(3,27)

Short Description: Generate a single Housekeeping Parameter Report

Description: A one shot report for the addresses housekeeping parameter report structure is generated. This report is a TM(3,27) message type.

Constraints:

0: HKMODE.inMode(ACTIVE) // Housekeeping Service must be active

In-Parameter:

```
VA_SID sidList: constrained : sidList.length > 0 // Variable Array of SIDs
{
```

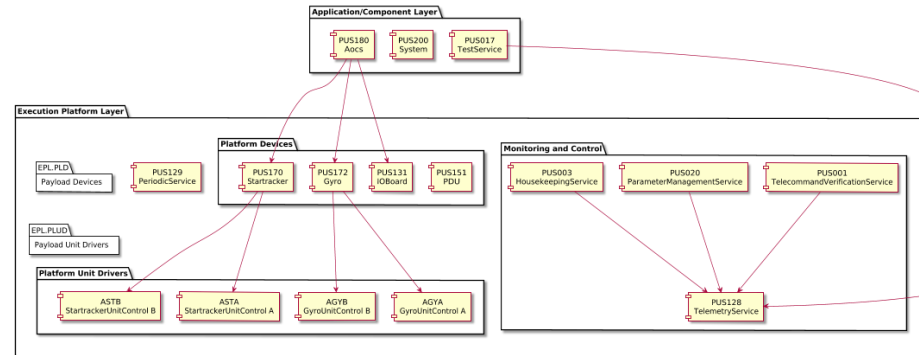
```
for (idx ++ in [0..sidList.length]) {
  UI32 sid = sidList.data[idx];
  if (HKREPORTS[sid].valid) {
    VA_PID structure = HKREPORTS[sid].parameterList;
    TELEMETRY (3,25)
```

Description: Housekeeping Parameter Report Structure

```
[SID : UI32 = sid // SID
Parameter Set structure ]
```

```
} if
} for
}
```

- Documentation



Further Extensions / Languages are Feasible

- Onboard Software Reference Architecture (OSRA)
- System Specification and Requirements Engineering
 - Model Satellite Hardware
 - Deployment of Functions/Software to Processing Nodes
- Simulation Models (SVF)
- Test Scripts
- Automation (OPS, AIT)
- Onboard Control Procedures

- ... what ever you come up with ...

Thank You !

Questions ?

SPACE SYSTEMS

A Model Based and Domain Specific Development Environment

We. Create. Space.