HISTORY AND STATE OF THE PRACTICE OF MODEL-BASED SOFTWARE ENGINEERING IN THALES ALENIA SPACE

REGIS DE FERLUC, MARCO PANUNZIO

MBSE 2020

Date: 28/09/2020
Ref: 0005-0012366892
Template: 83230834-ACQ-TAS-EN-006

EMERGENCE OF MODELING FOR ON-BOARD SOFTWARE

/// Adoption of Model-based Software Engineering in Thales Alenia Space

- **EMERGENCE OF MODELING FOR ON-BOARD SOFTWARE**
- **CONSOLIDATION AND MATURITY**
- **APPLICATION TO PAYLOAD SOFTWARE DEVELOPMENT**
- **LINK TO OTHER DISCIPLINES**

/// Initial conclusions

/// Challenges for the near future

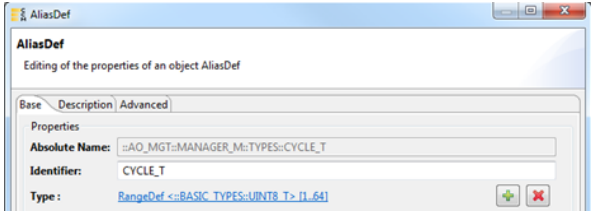
[Some satellite artist's views and photos in this presentation are courtesy of ESA, CNES]

EMERGENCE OF MODELING FOR ON-BOARD SOFTWARE

/// Operational use of MBSE for on-board software in Thales Alenia Space starts in 2009

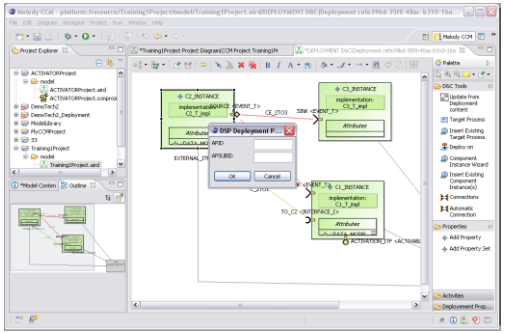
KEY DESIRED ELEMENTS OF MODELING LANGUAGE AND MBSE APPROACH

- **Precise data type modeling**
  - mirroring the full expressiveness of the Ada language




```
-- from ::AO_MGT::MANAGER_M::TYPES::CYCLE_T
-- AOCS elementary cycle number type
subtype CYCLE_T is BASIC_TYPES.UINT8_T range 1 .. 64;
```

- **Support for PUS modeling in the design space**
  - APID, Service, Subservice, ParamIDs, etc… as first class modeling entities



- **Target runtime adapted for embedded space software**

EMERGENCE OF MODELING FOR ON-BOARD SOFTWARE

- **NO "OFF-THE-SHELF" APPROACH WAS CONSIDERED SUITABLE AT THE TIME**
- **WITH THALES GROUP, WE DEVELOPED OUR OWN DOMAIN-SPECIFIC LANGUAGE AND ECLIPSE-BASED MODELING ENVIRONMENT: CCM4SPACE**

Starting from an existing OMG standard: Lightweight CCM (LwCCM)

- Modeling environment
- Code generators
  - Interface and component code
  - Skeletons for business code
- Document generators
- Targeting TAS reference software architecture and run-time

- **KEY DECISION: TO DEVELOP AND MAINTAIN IN-HOUSE THE MODELING TOOLCHAIN**
  - All space-specific extensions and generators mastered in-house by a team in close relationship with projects
  - Knowledge of space-specific needs and desired code patterns has been essential for success

EMERGENCE OF MODELING FOR ON-BOARD SOFTWARE

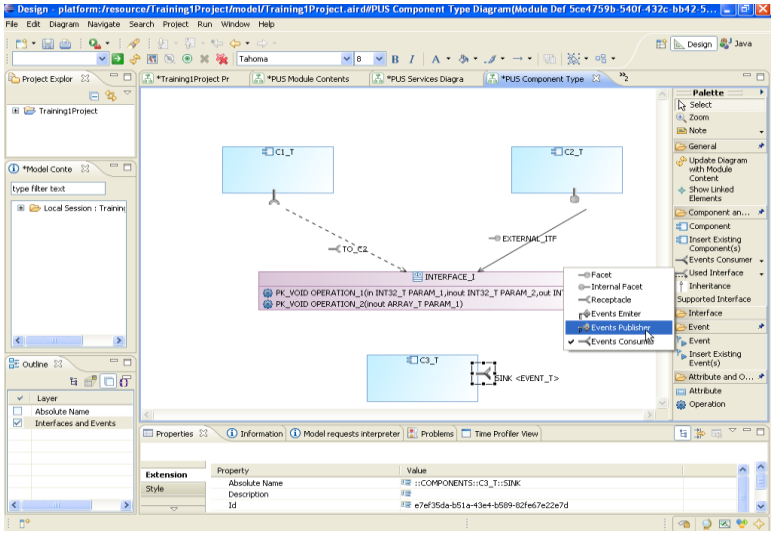
/// Major achievements in 2009-2012

- Successful compromise between modeling abstraction and productivity found
- **=> Model software architecture, not the source code!**

- Automated and coherent generation of document output (ICD, SDD)
- Automated and coherent generation of output for the spacecraft database
- **=> Eliminate error prone phases from backlog of engineers**

- Exhaustive code generation for all code related to data types, component architecture and component interfaces for satellite platform software
- **=> No more SW-SW ICD problems**

- Code generation for PUS TM / TC encoding / decoding
- **=> Let developer focus on business code. The design environment manages PUS-related aspects.**



/// First successful operational deployment on ESA missions

- **Sentinel 3**

CONSOLIDATION AND MATURITY

/// From 2012 to 2016 the major focus has been to consolidate the approach and increase its maturity

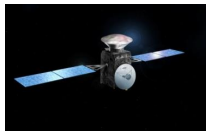
- **Quickly respond to specific project needs (adaptations and performance optimisations)**
- **Extend the toolset with additional features**

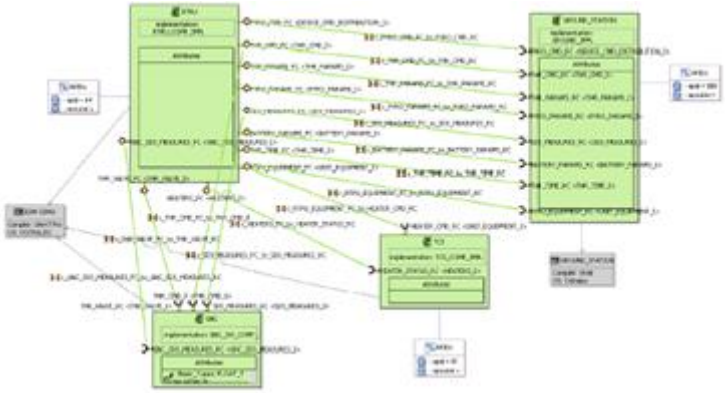
ADDITIONAL FEATURES AND IMPROVEMENTS

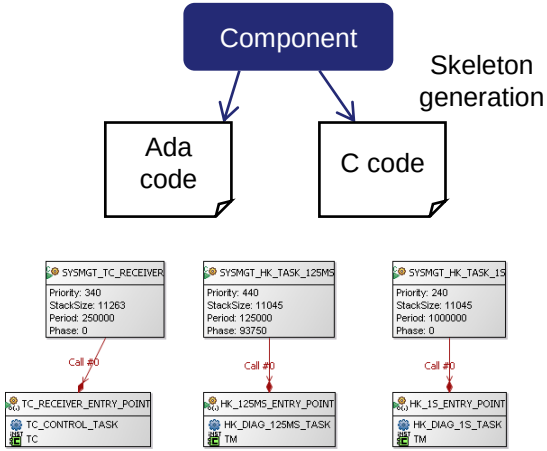
- Exhaustive modeling and code generation for all component interactions
- Flexibility for component implementation
  - => Use either Ada or C as implementation language for components
- Modeling and generation of tasking and semaphores
  - => Master the complexity of concurrency aspects and keep coherent the code with the intended real-time architecture
- Software “missionisation” from spacecraft data base, including automated testing

/// Successful operational deployments

- **Iridium Next (66-satellite Telecom constellation)**
- **Exomars 2016**







APPLICATION TO PAYLOAD SOFTWARE DEVELOPMENT

/// Whilst devised primarily for platform software development, TAS' MBSE approach has been adapted and deployed on selected payload software developments, starting from 2015

INCREASED COMPLEXITY OF PAYLOAD SOFTWARE

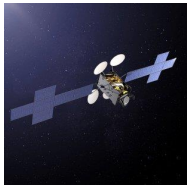
- => software architecting has increased importance
- => greater impact of concurrency and real-time considerations
- => greater resource management constraints
- => greater use of complex Monitoring and Control protocols (e.g., PUS, or payload-specific)

APPLICATION OF THE MBSE APPROACH TO A SUBSET (COMMAND AND DATA HANDLING) OF PAYLOAD SOFTWARE CONSIDERED A VALUE ADDED IN SOME CASES

- Could reuse solutions already devised for platform software for many aspects
- Required adaptations
  - different HW platforms and (partly) software practices
  - emergence of a "payload reference software execution platform"

/// Successful operational deployments

- **Meteosat Third Generation (MTG), FCI and IRS ICU common software execution platform**
- **Sizeable telecom payloads**



BEYOND ON-BOARD SOFTWARE DEVELOPMENT

/// Starting from 2015, it became evident that a coherent modeling approach would need to address not only on-board software, but also all related disciplines

=> EMERGENCE OF THE "MODEL-BASED SYSTEM AND SOFTWARE FACTORY"

MAJOR GOALS

- **Global optimisation of the whole process** rather than local efficiency only
- Bridge with approaches (or modeling approaches) of other disciplines, while preserving discipline know-how and practices
- Support for co-engineering between disciplines

KEY CHALLENGES

- Major paradigm shift and overall process refinement
- Alignment of tools and technologies
- Coordinated configuration management

MODEL-BASED SYSTEM & SOFTWARE FACTORY

Model-Based System & Software Factory

Capella — **Avionics Modeling** (2015)

V&V test campaign and scenarios modeling (2018) — VIVO

Sys2Soft

Avionics to Software transition Modeling of on-board devices and generation of device software managers (2016)

Operation Preparation Environment : SCOPE (2015)

SDB-NEXT (2018)

MATLAB & SIMULINK

Full GNC modeling and code generation + integration with OBSW modeling (2017)

CCMSpace

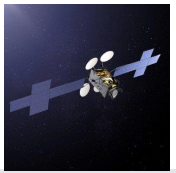
On-board software modeling and code generation (2009)

Model-based FDIR Design (2018)

SCOPE-FDIR

/// Successfully deployed incrementally on

- SWOT (CNES) [Avionics modeling, Sys2Soft, FDIR design only]
- Spacebus NEO telecom product line (NEOSAT)



(in parenthesis, year of first operational deployment)

INITIAL CONCLUSIONS

/// Factors that led to success of modeling (in our company experience)

- **INVESTMENT AS SELF-FUNDED R&D AND PARTICIPATION TO R&D PROJECTS FUNDED BY AGENCIES AND EU**
  - Important to exchange ideas, implement initial concepts, prioritise value added elements
    - => focus on improving productivity directly (e.g., automation) or indirectly (e.g., improved coherence, absence of defects)
- **STRONG IN-HOUSE SKILLS IN MODELING AND TOOL DEVELOPMENT**
  - Considered essential to perform incremental development of approach and customisations
- **A BOTTOM-UP AND INCREMENTAL APPROACH**
  - Have clear goals from the onset of what you want to achieve in terms of modeling products (e.g., code, documents) or process improvement
  - Bottom-up approach helped, as the modeling “stop criteria” is evident and tangible
    - => we do not model for the sake of modeling, but **to increase productivity and value added**
  - Deployment of full approach on a long time scale not only because of development and maturation time, but also because it required a cultural mindset change
- **MANAGEMENT WILLINGNESS TO INDUSTRIALISE AND DEPLOY OPERATIONALLY PROMISING R&D RESULTS**
  - Decisiveness is important to shift from R&D to operational use
    - => avoid the “eternal prototype” phase

/// Key messages

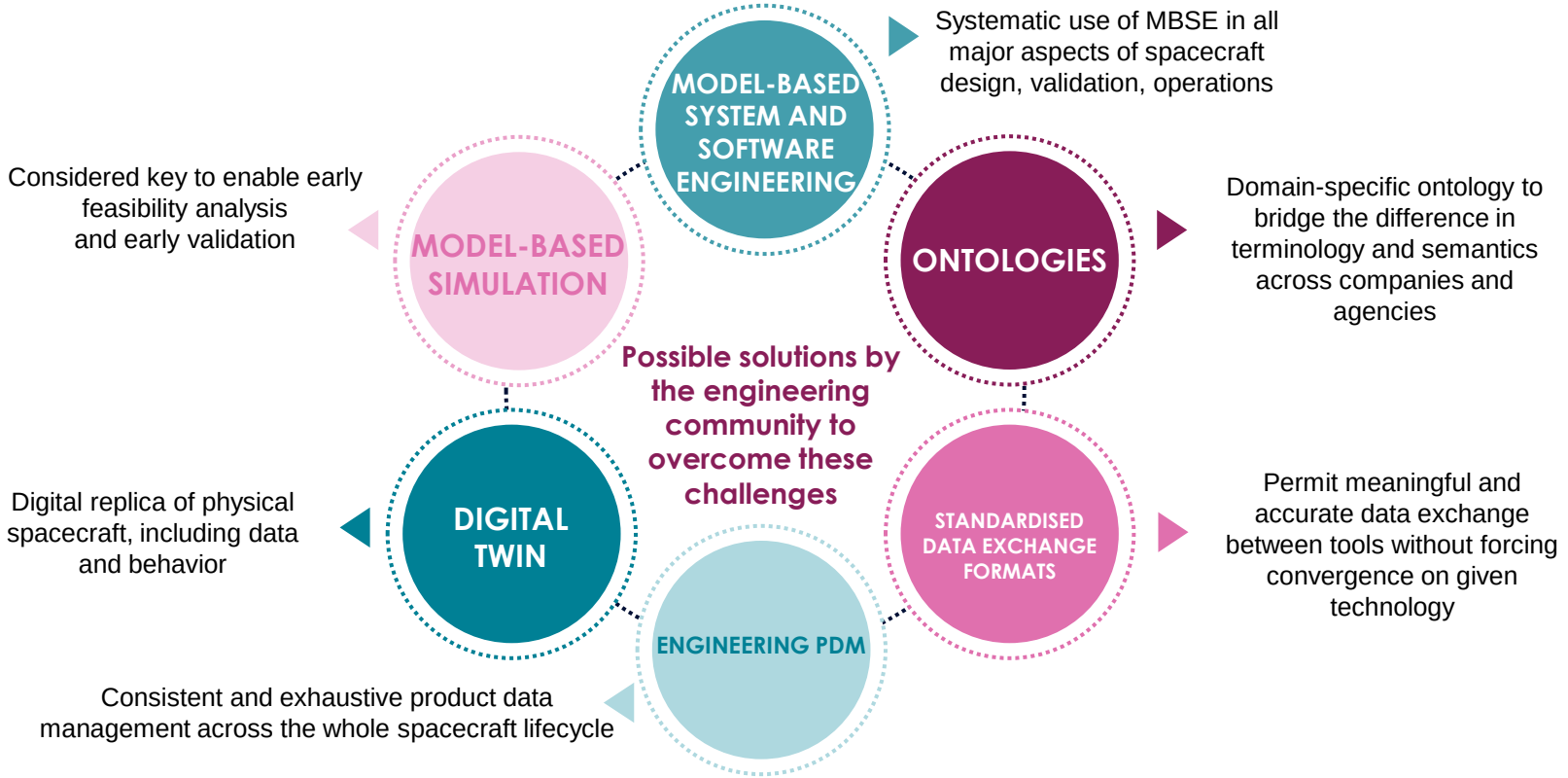
- **MBSE / MODELING IS VERY MUCH ALIVE, AND ACTIVELY USED EVERY DAY**
- **MODELING HAS LED TO A POSITIVE, PROFOUND AND LONG-LASTING TRANSFORMATION IN THE NATURE OF ENGINEERING PROCESSES AND ACTIVITIES FOR SOFTWARE DEVELOPMENT AND BEYOND**

CHALLENGES FOR THE NEAR FUTURE

/// Three major trends

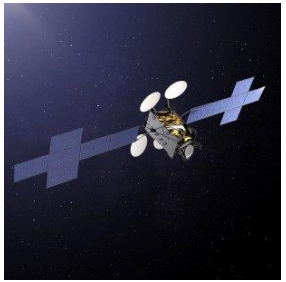
- **A SYSTEMATIC SEARCH FOR A SOLUTION TO THE "DIGITAL CONTINUITY" CHALLENGE**
  - the capability of *meaningfully* model a **full system** from the early design phases (i.e., 0, A/B1)
  - and to transition modeling data in the detailed design and implementation phases (B2, C, D) and later into operations
    - without loss of data
    - adjusting the abstraction of representation to the level meaningful to the actual phase of development
- **THE RECONCILIATION OF HETEROGENEITY INTERNAL TO THE PROJECT**
  - approaches, methodologies and technologies best suited for individual disciplines which need to be able to fit together
    - e.g., System, structure, AOCS, thermal, avionics / SW, etc. for **development**, **simulation** and **validation** aspects.
  - free and meaningful circulation of data between and beyond disciplines, throughout the whole development lifecycle
- **THE RECONCILIATION OF HETEROGENEITY BROUGHT BY COLLABORATION OF SEVERAL COMPANIES (FULL CONSORTIUM), OR SEVERAL AGENCIES**
  - Additional challenges on top of those above
    - Heterogeneity of terminology, standards, methods and tools
    - Shared concept of product baseline
    - Cybersecurity, confidentiality, IP protection, export control

CHALLENGES FOR THE NEAR FUTURE

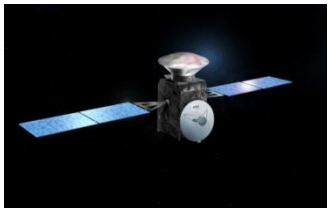


THANK YOU FOR YOUR ATTENTION

Questions?



Model-Based System & Software Factory



CCMSpace