Application of modeling techniques for on-board satellite applications

Requirements management, design, validation and verification activities

Óscar Rodríguez Polo, Pablo Parra Espada, Aaron Montalvo

I-MDE-A: WORKSHOP ON MODEL-DRIVEN ENGINEERING AND ITS APPLICATIONS
IMDEA Software Institute
May 16, 2023

Universidad de Alcalá

SRG

Summary

Introduction

Space Research Group (SRG-UAH)

On-Board SW Development

2004

Nanosat 01

National Institute of *Aerospace Technology (Spain)*

2009

2010

Solar Orbiter

Introduction

Space Research Group (SRG-UAH)

On-Board SW Development

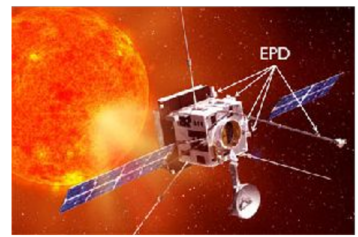
2004

Nanosat 01

2009

2010

Solar Orbiter



National Institute of *Aerospace Technology (Spain)*

Modelling Techniques: MBSE-> MDE

Introduction

Space Research Group (SRG-UAH)

On-Board SW Development

Embedded Software

Cross Compiled (Platform Config Control),
Low Memory Footprint (C or EC++)

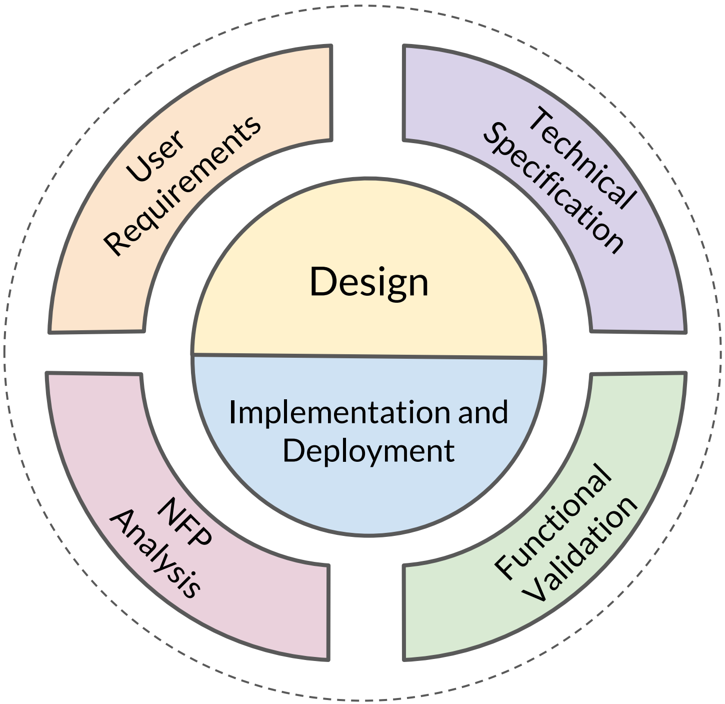
Real Time Software

Processor load, deadlines

Deterministic Software

No dynamic task creation or memory allocation
(Platform Config Control ->MICOBS)

Introduction



ECSS Standards
On-board software
ESA missions

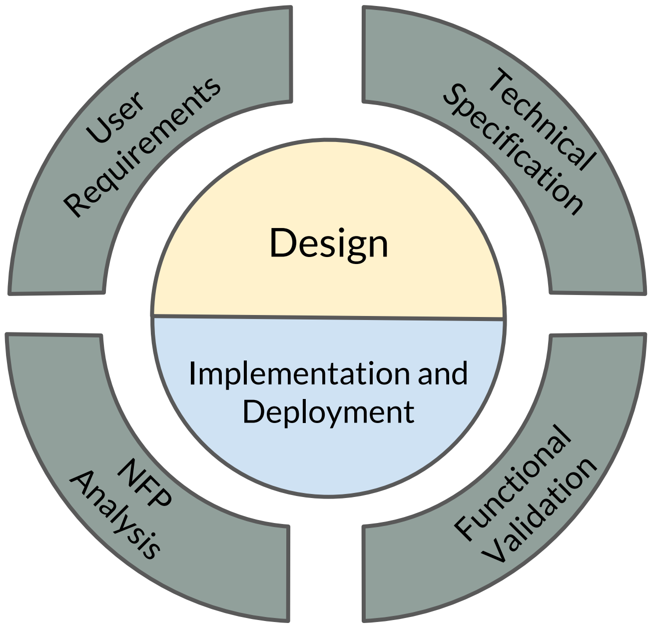
Verification

Component Based SW Design Modelling

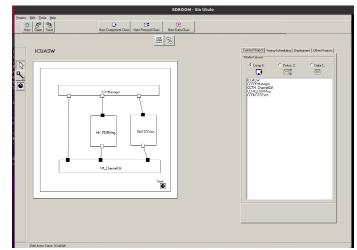
2001-**2004**

Nanosat 01

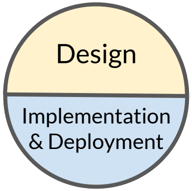
Component Based Design Modeling & Automatic Code Generation



EDROOM

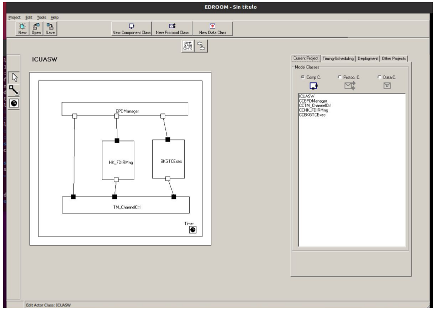


Component Based SW Design Modelling

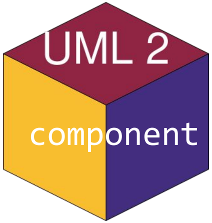


Component Based Design Modeling & Automatic Code Generation

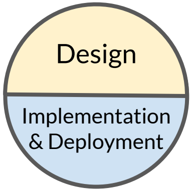
EDROOM



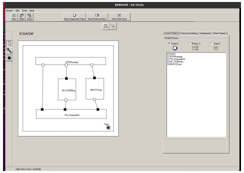
Based on ROOM
(Real-Time Object-Oriented Modelling)
Bran Selic



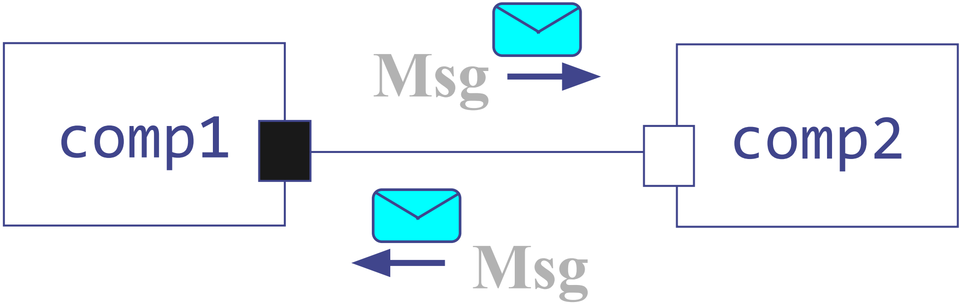
Component Based SW Design Modelling



EDROOM

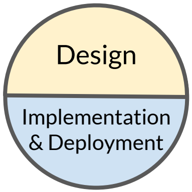


Communication based on message passing through ports



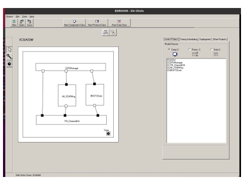
ROOM Computational Model

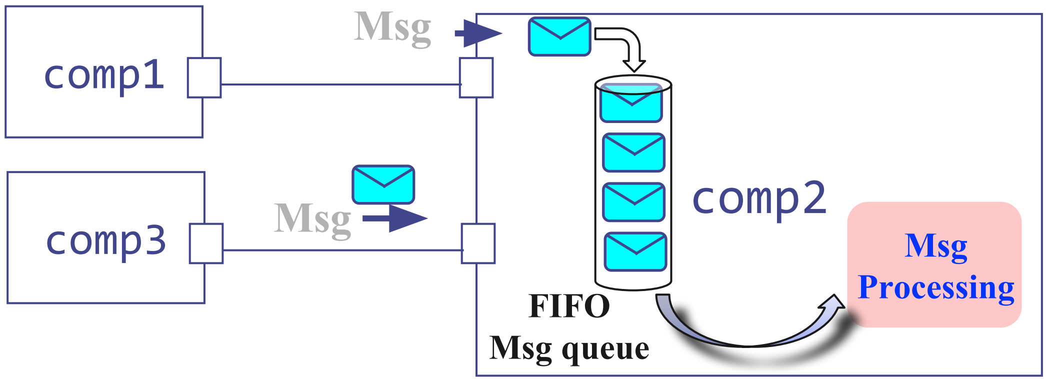
Component Based SW Design Modelling



Component = Message Processor

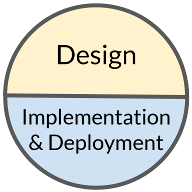
EDROOM



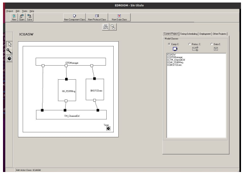


ROOM Computational Model

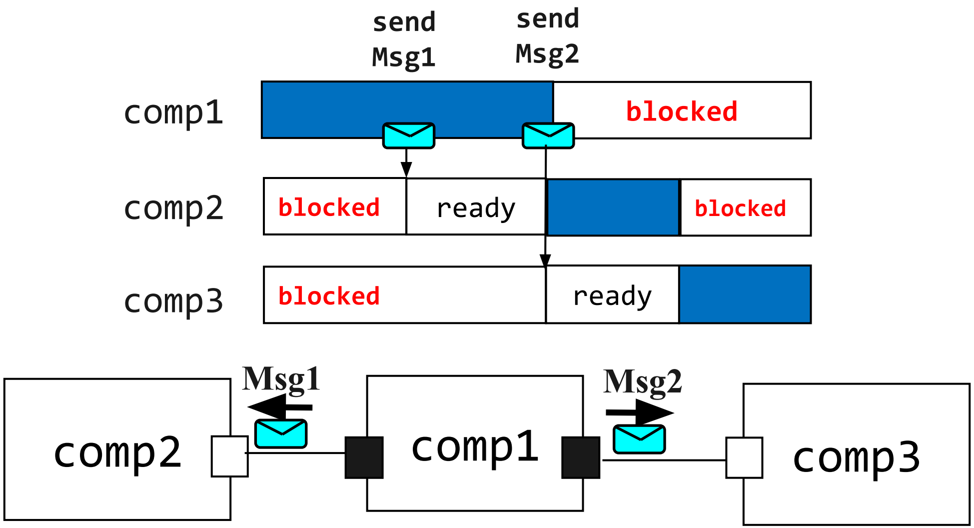
Component Based SW Design Modelling



EDROOM

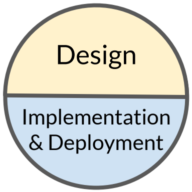


Asynchronous Communication

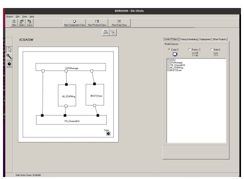


ROOM Computational Model

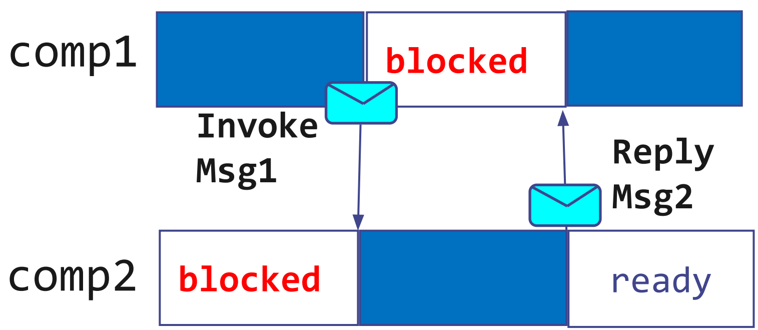
Component Based SW Design Modelling



EDROOM

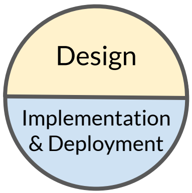


Synchronous Communication

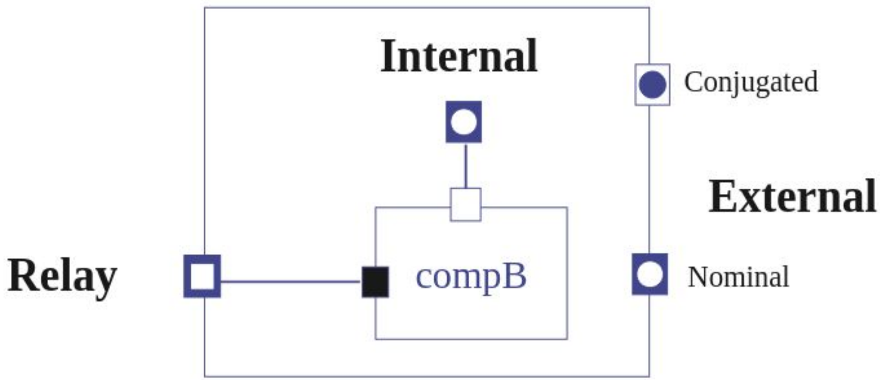


ROOM Computational Model

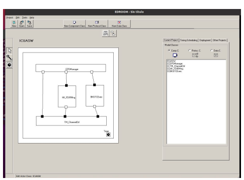
Component Based SW Design Modelling



Hierarchical Structure definition

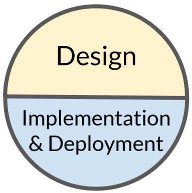


EDROOM

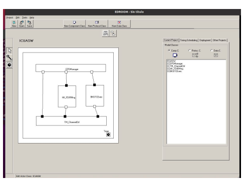


ROOM Computational Model

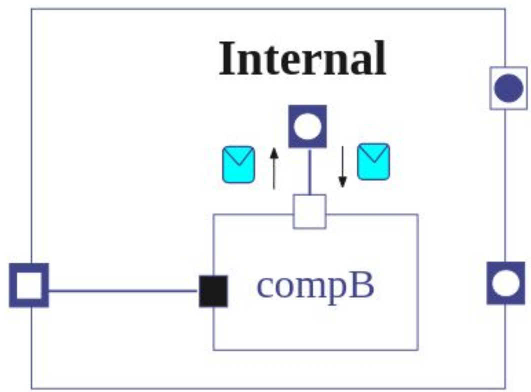
Component Based SW Design Modelling



EDROOM

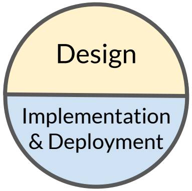


Communication with Sub-Components

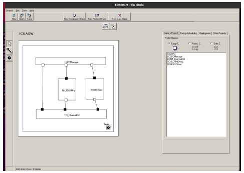


ROOM Computational Model

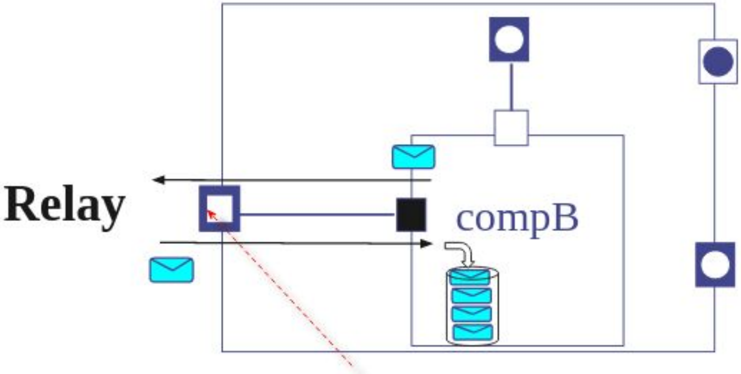
Component Based SW Design Modelling



EDROOM



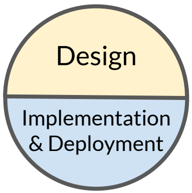
Exporting Sub-Components Ports



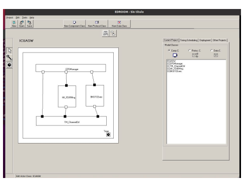
* Relay ports are distinguished graphically because they have a square inner edge

ROOM Computational Model

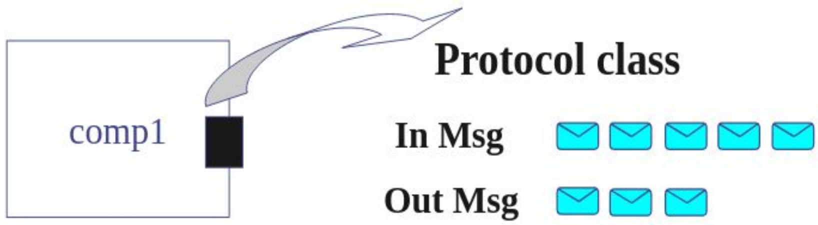
Component Based SW Design Modelling



EDROOM

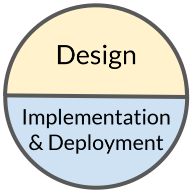


Communication Protocols



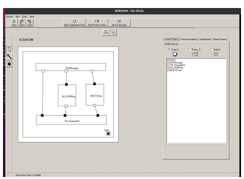
ROOM Computational Model

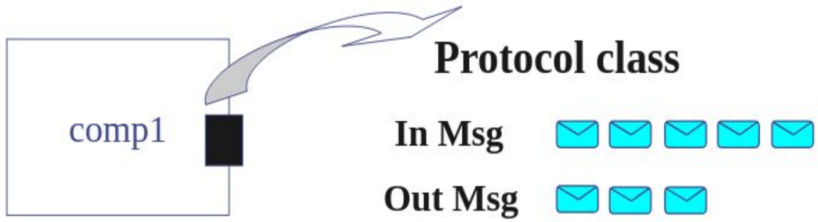
Component Based SW Design Modelling



Communication Protocols

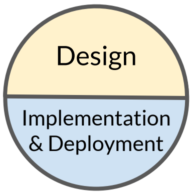
EDROOM



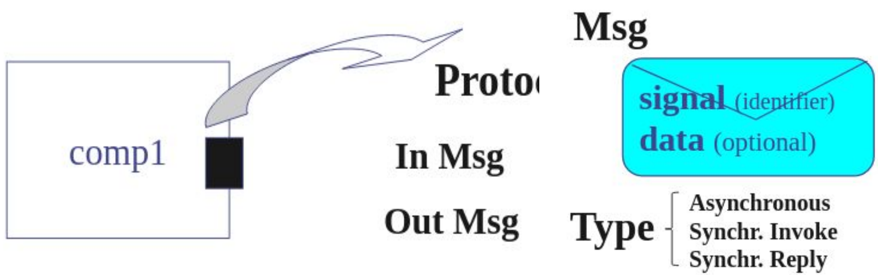


ROOM Computational Model

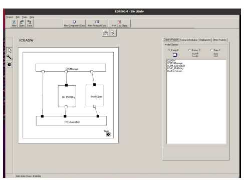
Component Based SW Design Modelling



Communication Protocols

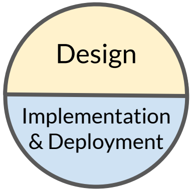


EDROOM

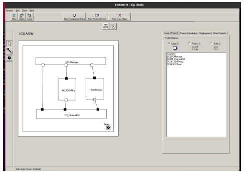


ROOM Computational Model

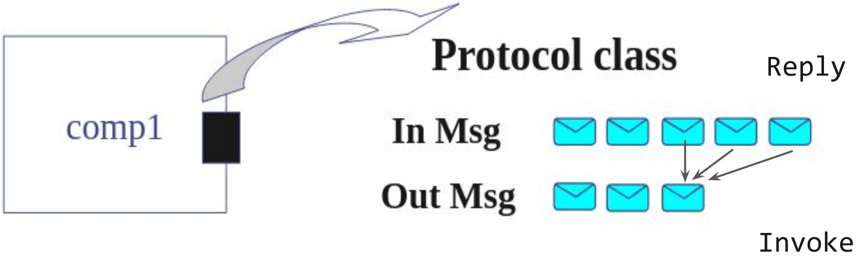
Component Based SW Design Modelling



EDROOM

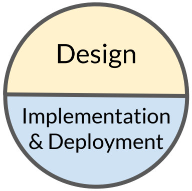


Communication Protocols

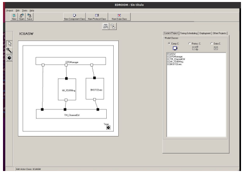


ROOM Computational Model

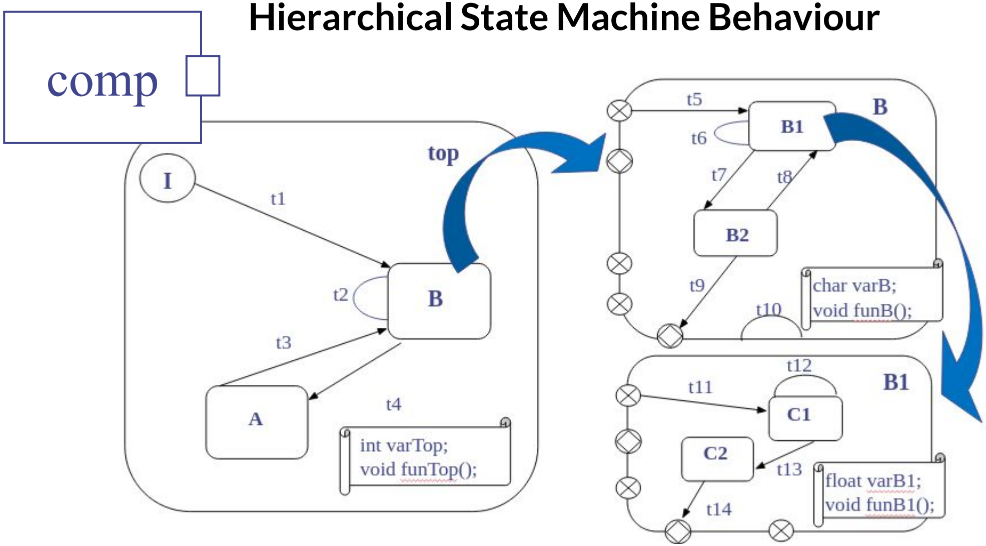
Component Based SW Design Modelling



EDROOM

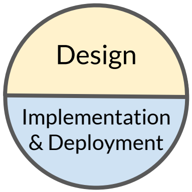


Hierarchical State Machine Behaviour

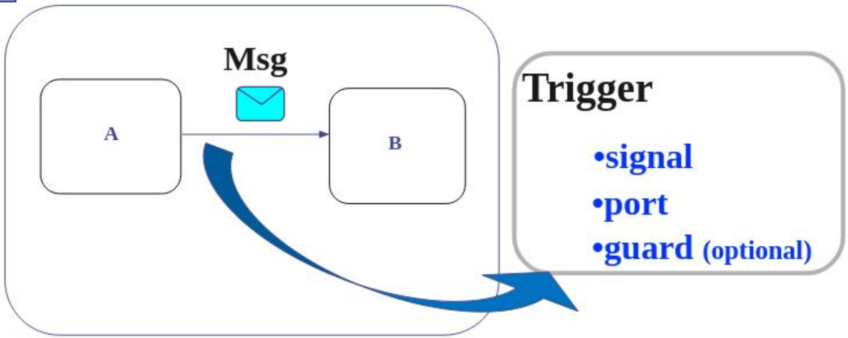


ROOM Computational Model

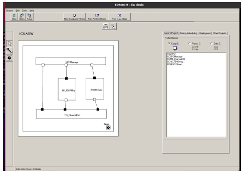
Component Based SW Design Modelling



Component Reactive behaviour

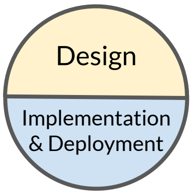


EDROOM



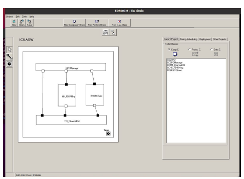
ROOM Computational Model

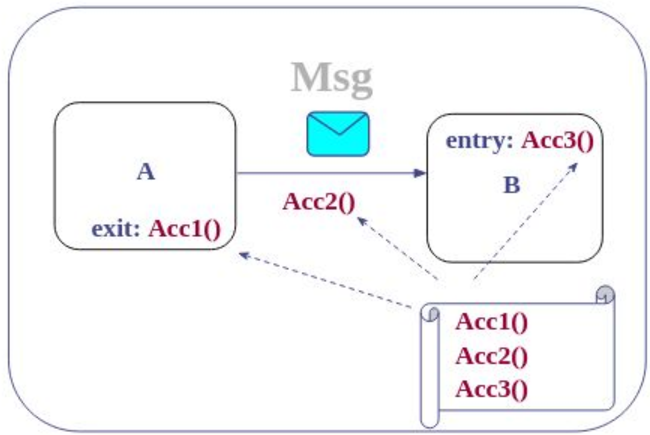
Component Based SW Design Modelling



Component Reactive behaviour

EDROOM

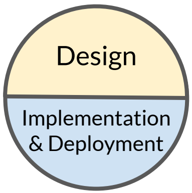




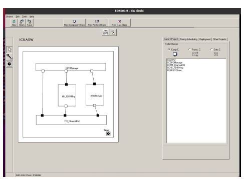

```
-send
-invoke
-reply
```

ROOM Computational Model

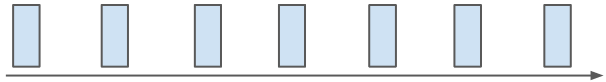
Component Based SW Design Modelling

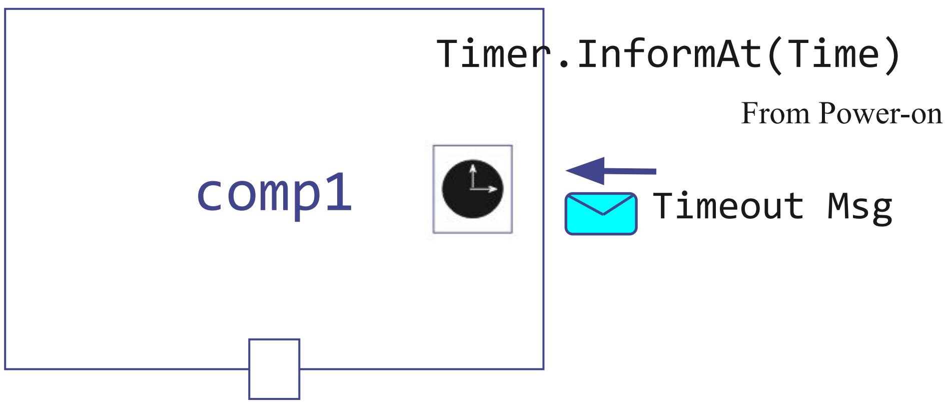


EDROOM



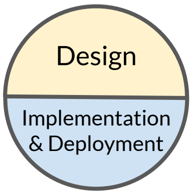
Task Periodic execution



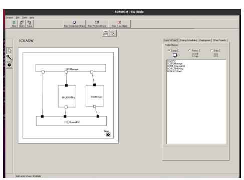


ROOM Computational Model

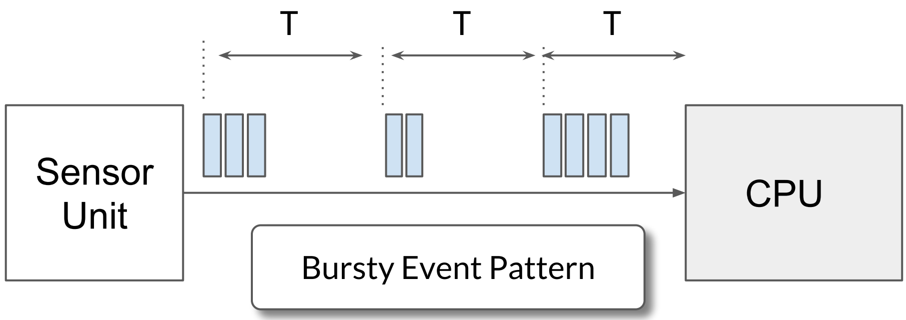
Component Based SW Design Modelling



EDROOM

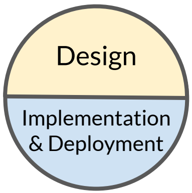


Events bounded but not periodic

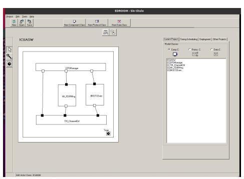


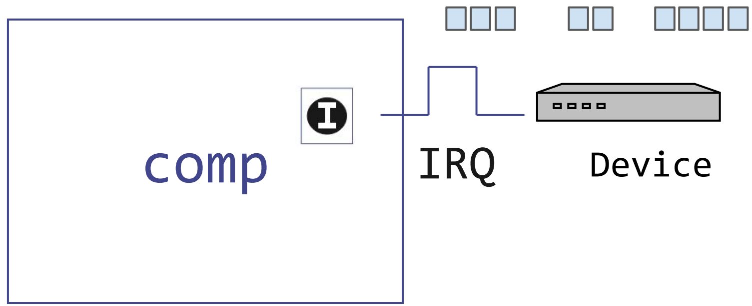
Not in ROOM Computational Model

Component Based SW Design Modelling



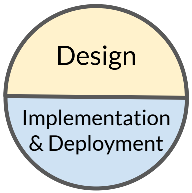
EDROOM



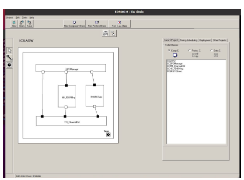


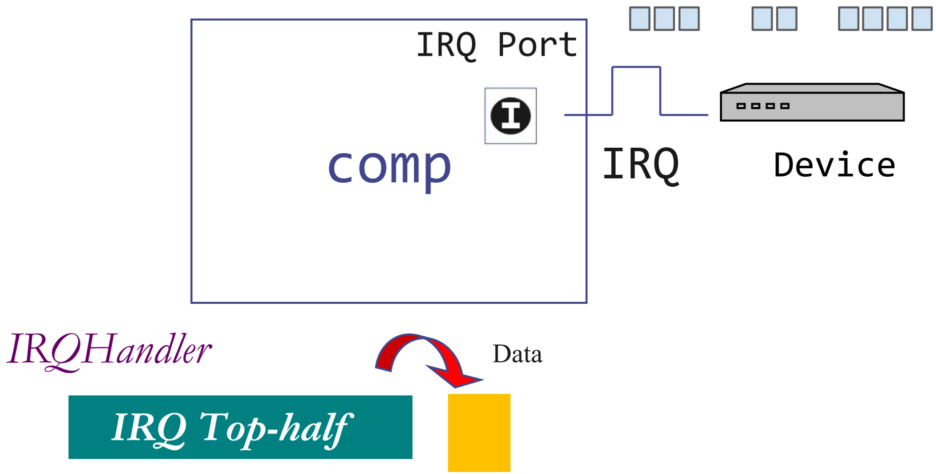
Computational Model Extension

Component Based SW Design Modelling



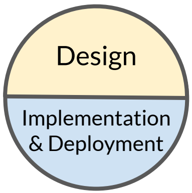
EDROOM



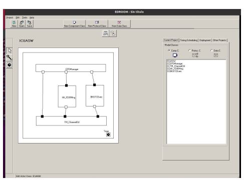


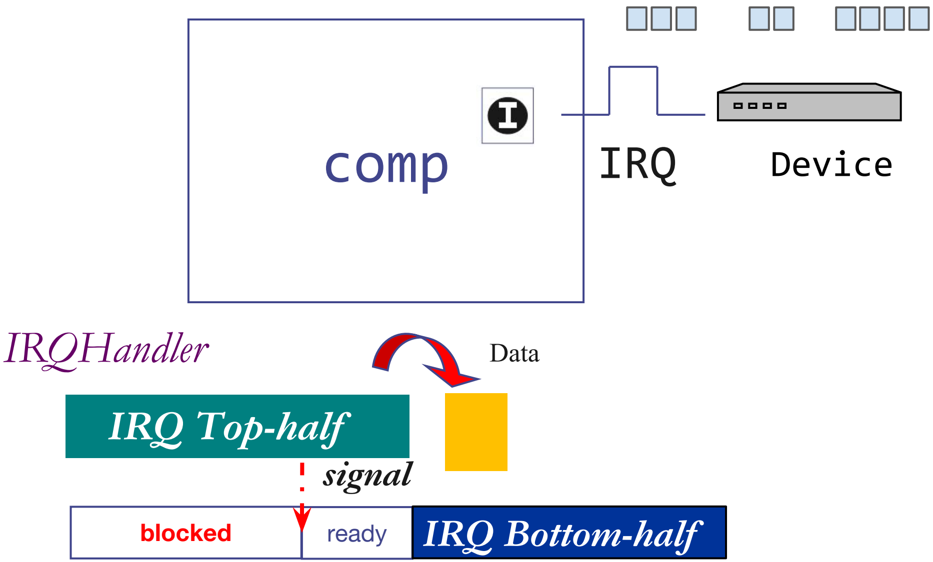
Computational Model Extension

Component Based SW Design Modelling



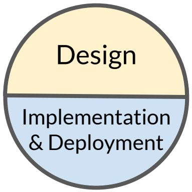
EDROOM



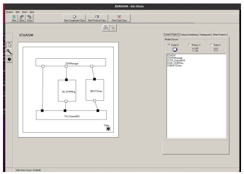


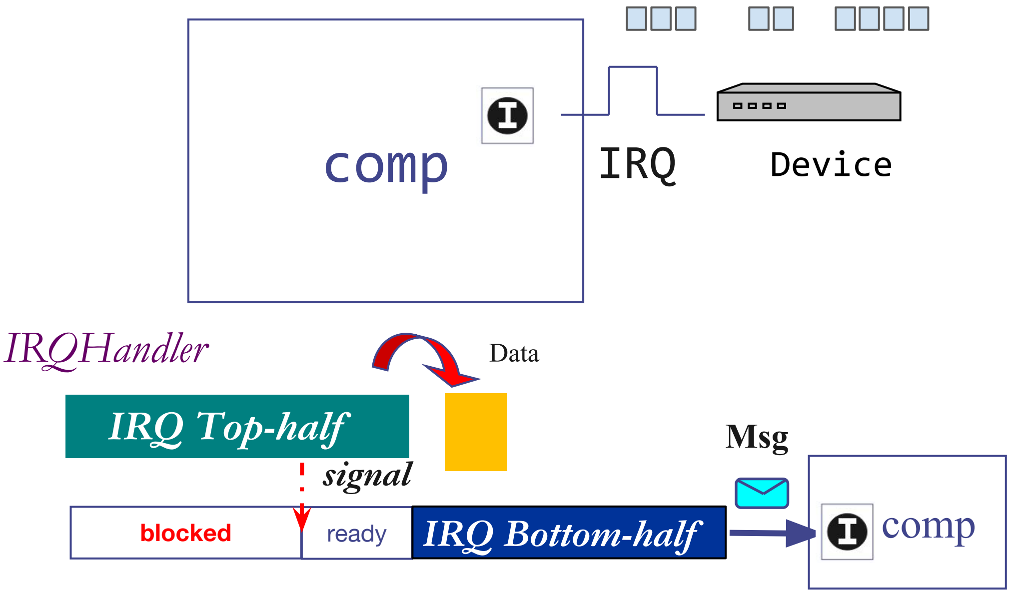
Computational Model Extension

Component Based SW Design Modelling



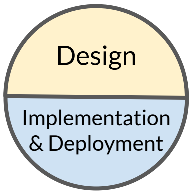
EDROOM



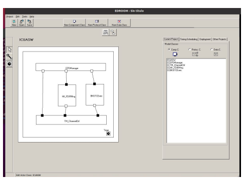


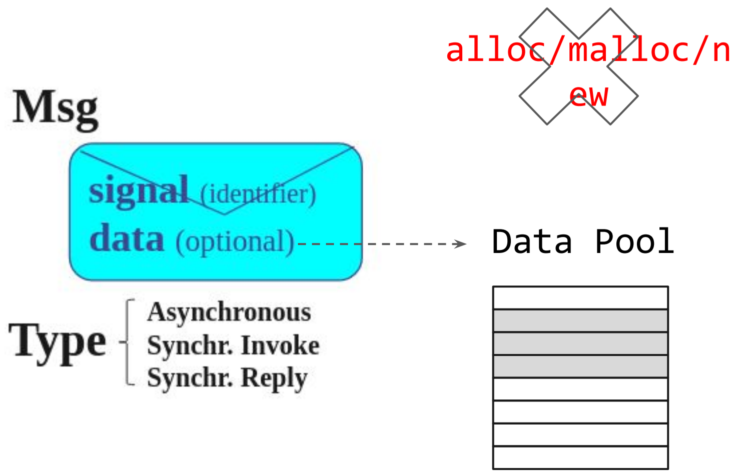
Computational Model Extension

Component Based SW Design Modelling



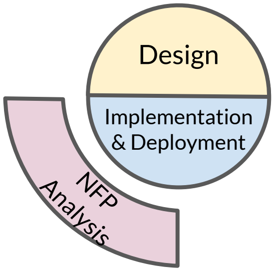
EDROOM



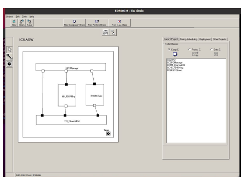


Computational Model Extension

Component Based SW Design Modelling

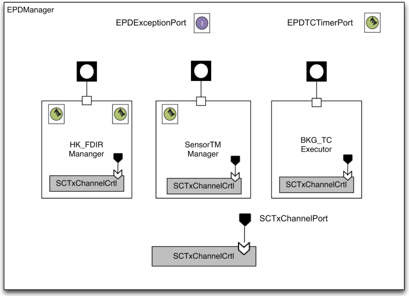


EDROOM

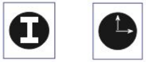


Computational Model Key Aspect

EDROOM Model

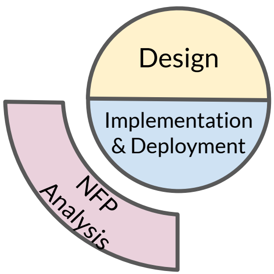


Source of Events

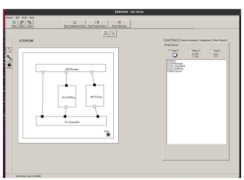


Bounded Response to Events

Component Based SW Design Modelling

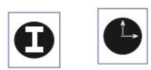


EDROOM



Computational Model Key Aspect

Source of Events



EDROOM Model

MAST Model

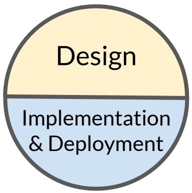
Bounded Response to Events

MAST: **Modeling and Analysis Suite for Real Time Applications**

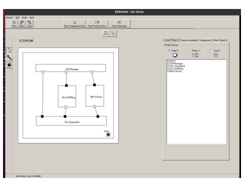
http://mast.unican.es/

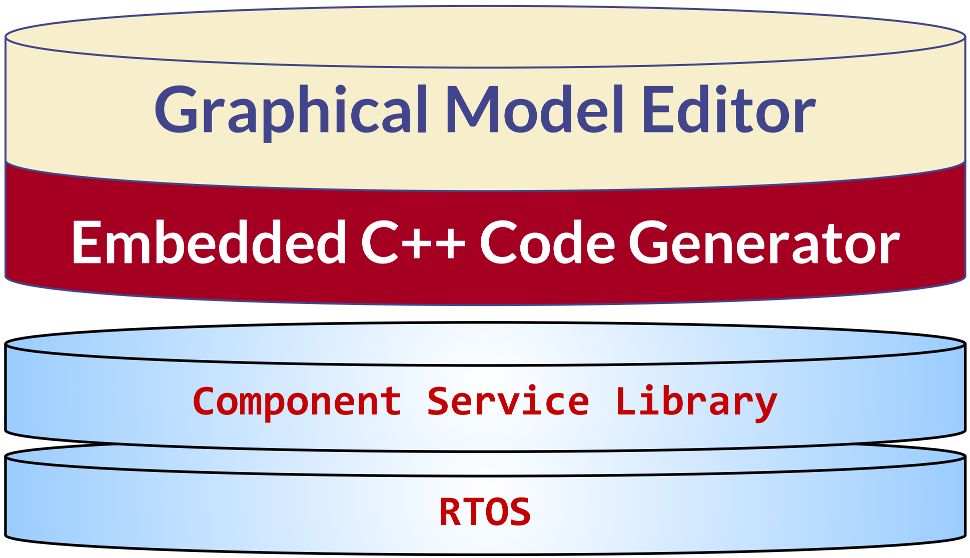
Schedulability Analysis!

Component Based SW Design Modelling



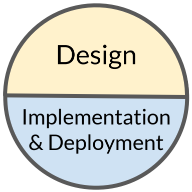
EDROOM





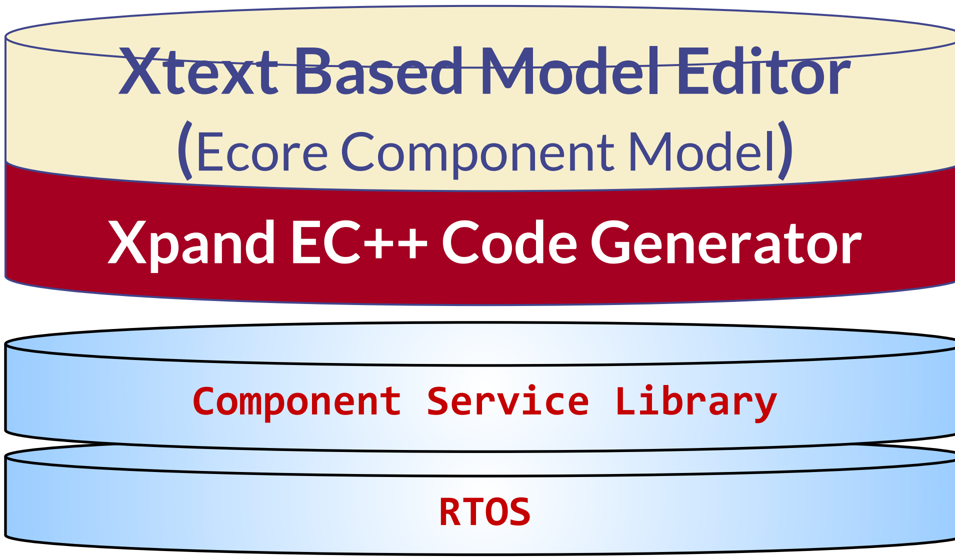
Embedded C++ Automatic Code Generation

Component Based SW Design Modelling



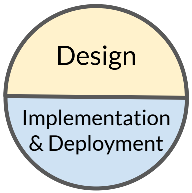
Xtext EDROOM Component Model

```
transitions{
  transition Init{
    id := 0;
    source := I;
    sink := Ready;
    asynchronous trigger {
       port := EDROOMsl;
       message := EDROOMStart;
       };
};
```

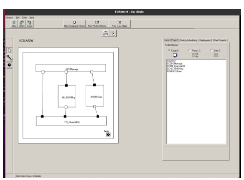


Embedded C++ Automatic Code Generation

Component Based SW Design Modelling



EDROOM



Components Embedded C++ generated Code
(platform independent code + Flat structure)

Component Service Library
Communication Timing
Memory-Control Interrupts Task-Frame
(platform independent code)

Pr_Task Pr_Mutex, Pr_Semaphore Pr_Time
Pr_IRQManager Pr_Kernel
RTOS_AL

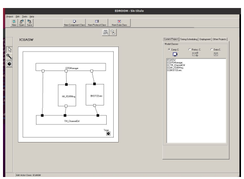
RTOS

Embedded C++ Automatic Code Generation

Component Based SW Design Modelling

Nanosat 01

EDROOM



Components Embedded C++ generated Code
(platform independent code)

Component Service Library
Communication Timing
Memory-Control Interrupts Task-Frame
(platform independent code)

Pr_Task Pr_Mutex, Pr_Semaphore Pr_Time
Pr_IRQManager Pr_Kernel
RTOS_AL

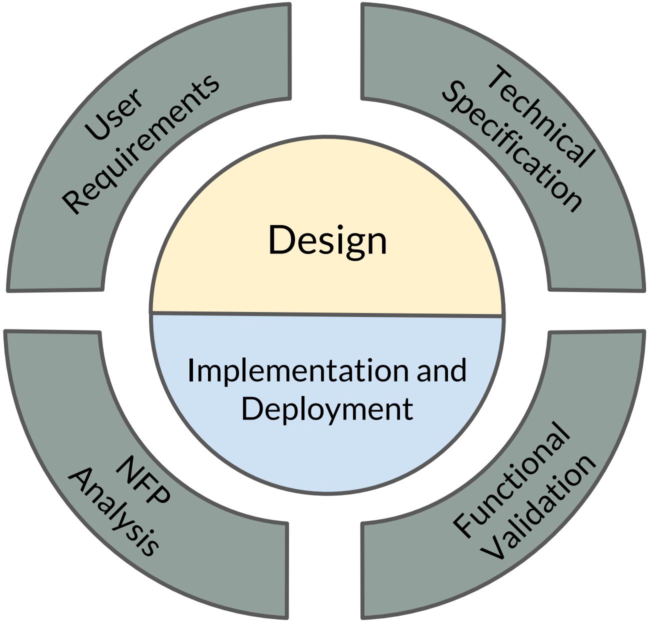
RTOS=CMX

Embedded C++ Automatic Code Generation

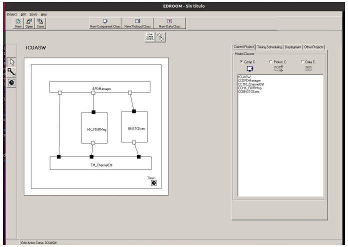
Component Based SW Design Modelling

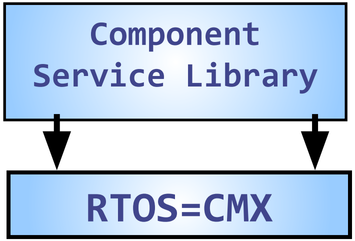
2005-**2009**

Component Based Design Modeling & Automatic Code Generation



EDROOM

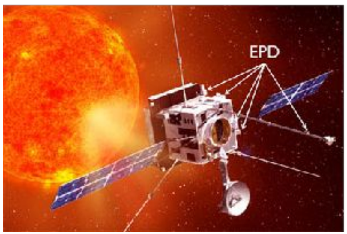




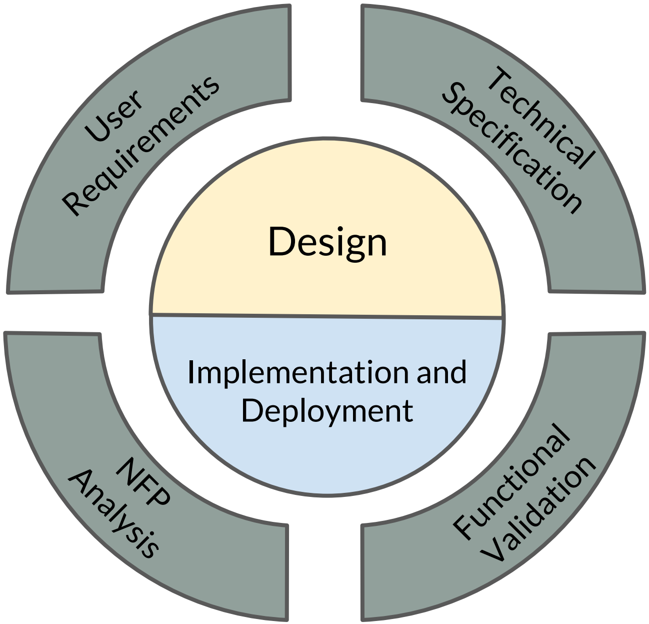
Component Based SW Design Modelling

2009-**2020**

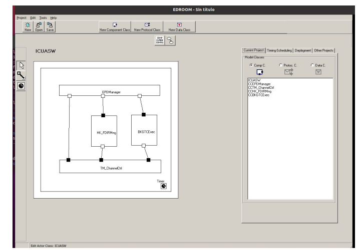
Solar Orbiter

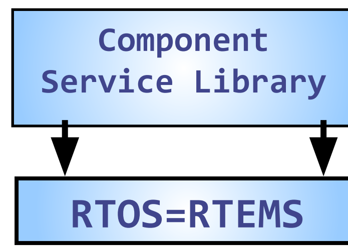


Component Based Design Modeling & Automatic Code Generation



EDROOM



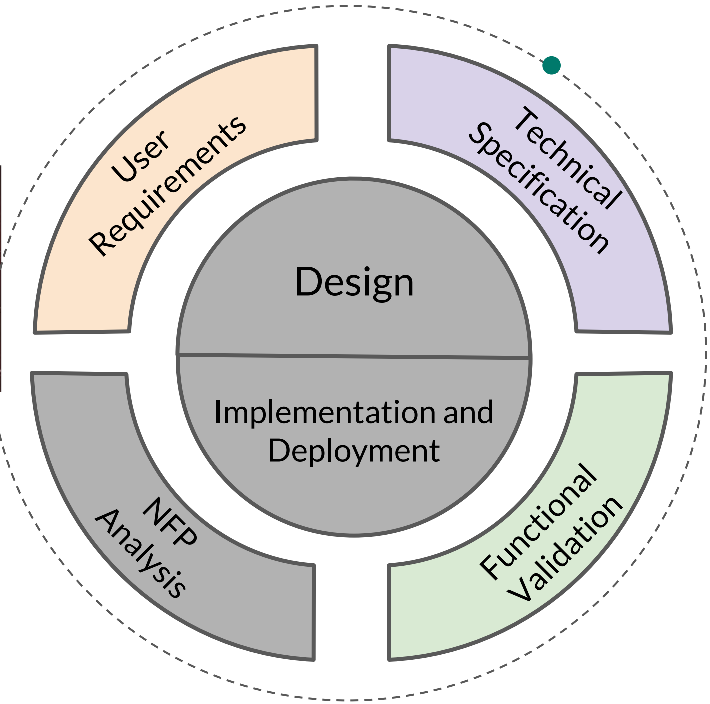


MDE Software Validation and Verification process

2009-**2020**

Solar Orbiter

MDE Software Validation and Verification (software system-level)



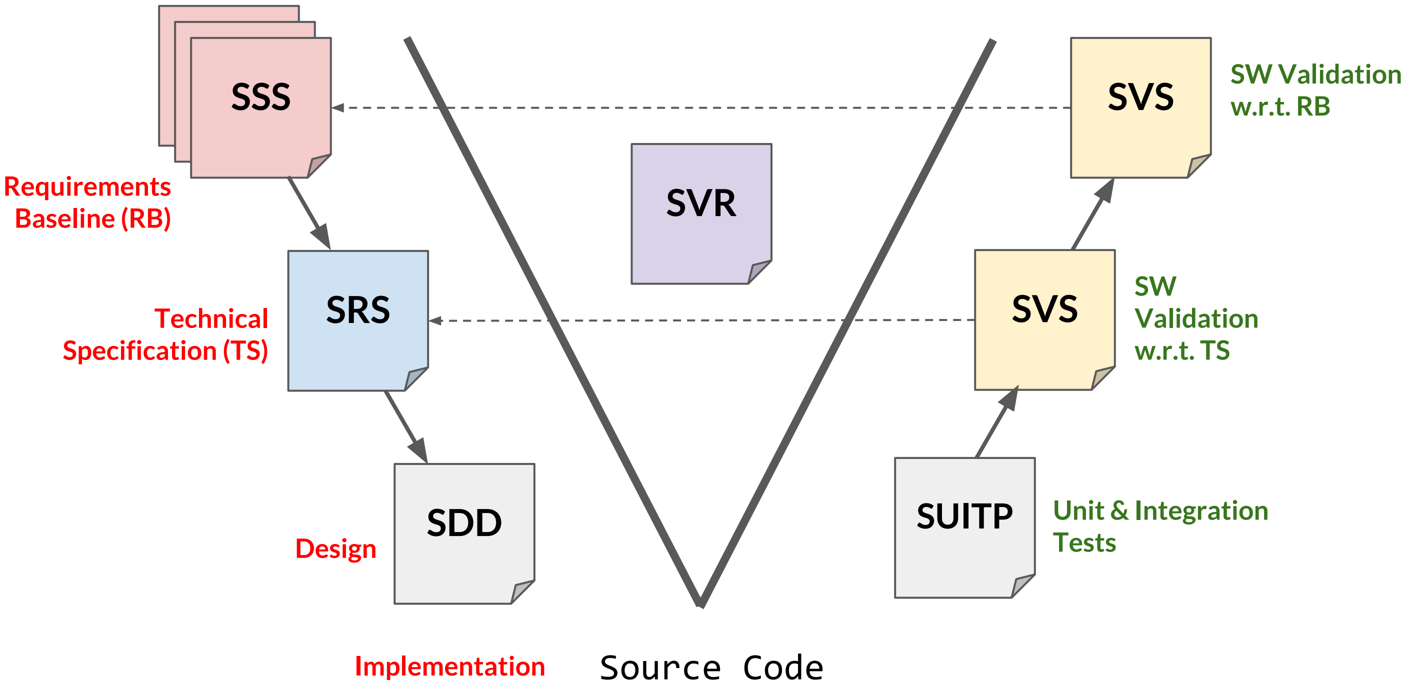
ECSS Standards
On-board software
ESA missions

Verification

MDE Software Validation and Verification process

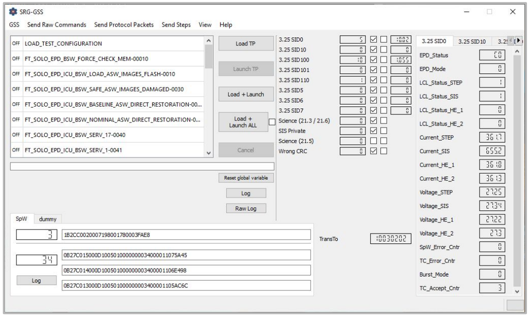
- Software system-level validation and verification process for space applications
  - Validation: “*are we building the right product?*”
  - Verification: “*are we building the product right?*”
- European Cooperation for Space Standardization (ECSS)
  - Defines standards that apply to every engineering process involved in space missions
  - Supported by the European Space Agency (ESA)
  - Software development: ECSS-E-ST-40C (March 2009)
  - Product assurance: ECSS-Q-ST-80C Rev.1 (February 2017)
  - Packet Utilization Standard: ECSS-E-ST-70-41C (April 2016)

Software Validation and Verification process



Mission-specific elements

- Telemetry/Telecommand information
  - Provided as a database
  - Follows the ESA's Satellite Control and Operation System 2000 (SCOS-2000) and the **ECSS PUS Standard** (Packet Utilization Standard)
- Ground Support Equipment (GSE)
  - Test harness that provides the required HW interfaces to the on-board processor
  - Emulates the flying environment
  - Enables the execution of the system-level validation tests
  - Managed using the Ground Support Software (GSS)



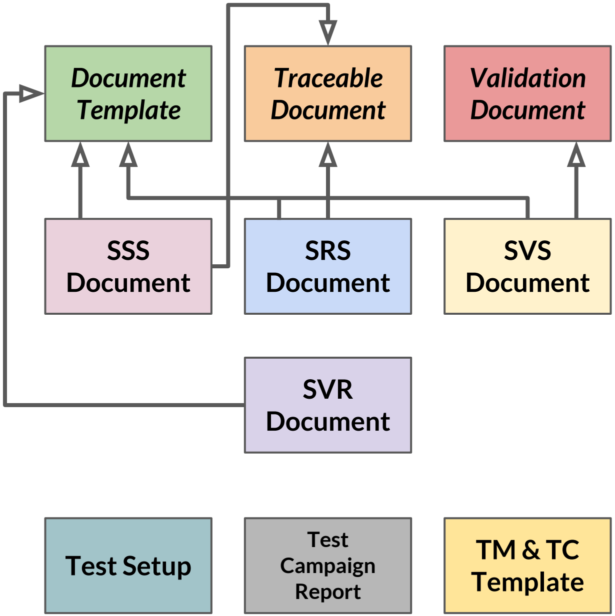
Model-driven approach

Automate the system-level validation and verification process for space software applications under the standards ECSS-E-ST-40 and ECSS-Q-ST-80 following a model-driven engineering approach

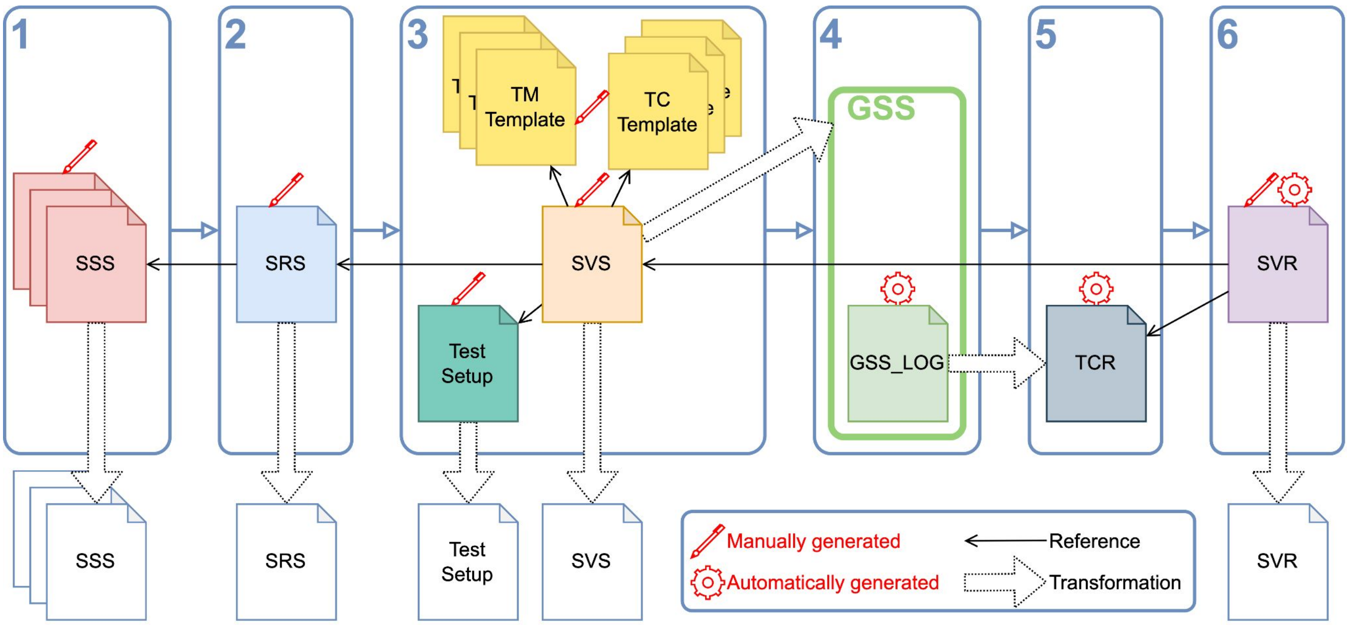
- Replace all the documents of the software V&V process with models
- Define model-to-model and model-to-text transformations to:
  - Generate the final deliverable documents
  - Generate the input files of the GSS with the test procedures
  - Incorporate the result of the test reports

Process Models

- **Common generic models**:
  - Document Template: models the contents of a text-based formatted document
  - Traceable Document: allows defining traceable items within a document
  - Validation Document: models documents that validate the traceable items included in other documents
- **Final document models**:
  - Requirements, validation and report models
  - Contain the same structure as the original documents
- **Other support models**:
  - Test Setup: contain the description and configuration of the test scenarios
  - Test Campaign Report: contains the results of the validation test campaign
  - TC and TM Templates: templates needed to define the TM/TC packets



Model-driven Validation and Verification process



Proof of concept

- Selection of a **subset of the requirements and validation tests** of the on-board software of the control unit of the Energetic Particle Detector (EPD) instrument on-board Solar Orbiter
- Generation of the documents in OOXML from the **SSS, SRS, SVS, and Test Setup models**
- **Generation of the same validation tests** that were used for the original qualification process
- **Integration** all **telemetry and telecommand** information from the EPD database
- The test results and the output documents were compared
  - The test log reports were the same as those created manually during the original test campaign
  - The automatically created documents were similar to the manually created ones
  - All **traceability matrices were obtained automatically => SVR**

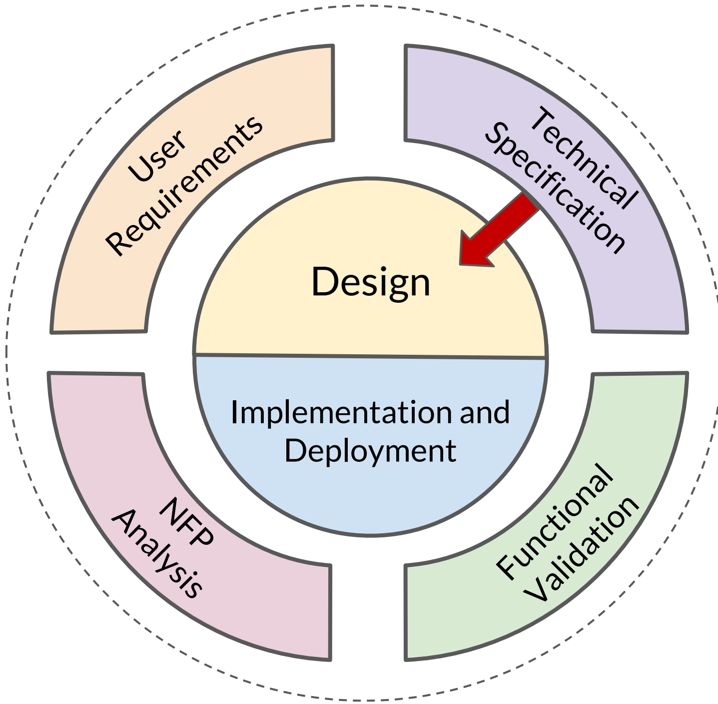
Conclusions

- Component Based Software Design Modelling has been successfully applied in three different missions.
- ROOM (UML2) Computational Model has been extended to handle specific on-board software requirements (Bursty Event Pattern)
- Embedded C++ Automatic Code Generation has been used in these missions with a low footprint
  - Nanosat-01 & Nanosat-1B used CMX as RTOS
  - Solar Orbiter EPD ICU used RTEMS (Edisoft) as RTOS

Conclusions (2)

- A model-driven engineering approach to the validation and verification process for space software applications has been also developed
- The solution follows the standards ECSS-E-ST-40 and ECSS-Q-ST-80 that are applicable in space software development
- The approach incorporates model-driven engineering techniques that maximize the automation of the different products required during validation and verification
- A complete proof of concept has been given corresponding to the development of the on-board software of the instrument control unit of EPD
- The resulting documents and reports have been compared successfully with the original ones

Future Work



From Technical Specification to Design

Verification

Thank you very much for your attention
Any questions?