New CCS technology protoyping

12th International Workshop on Simulation for European Space Programmes - SESP 2012

Claude Cazenave, Harald Eisenmann

Astrium Satellites

All the space you need

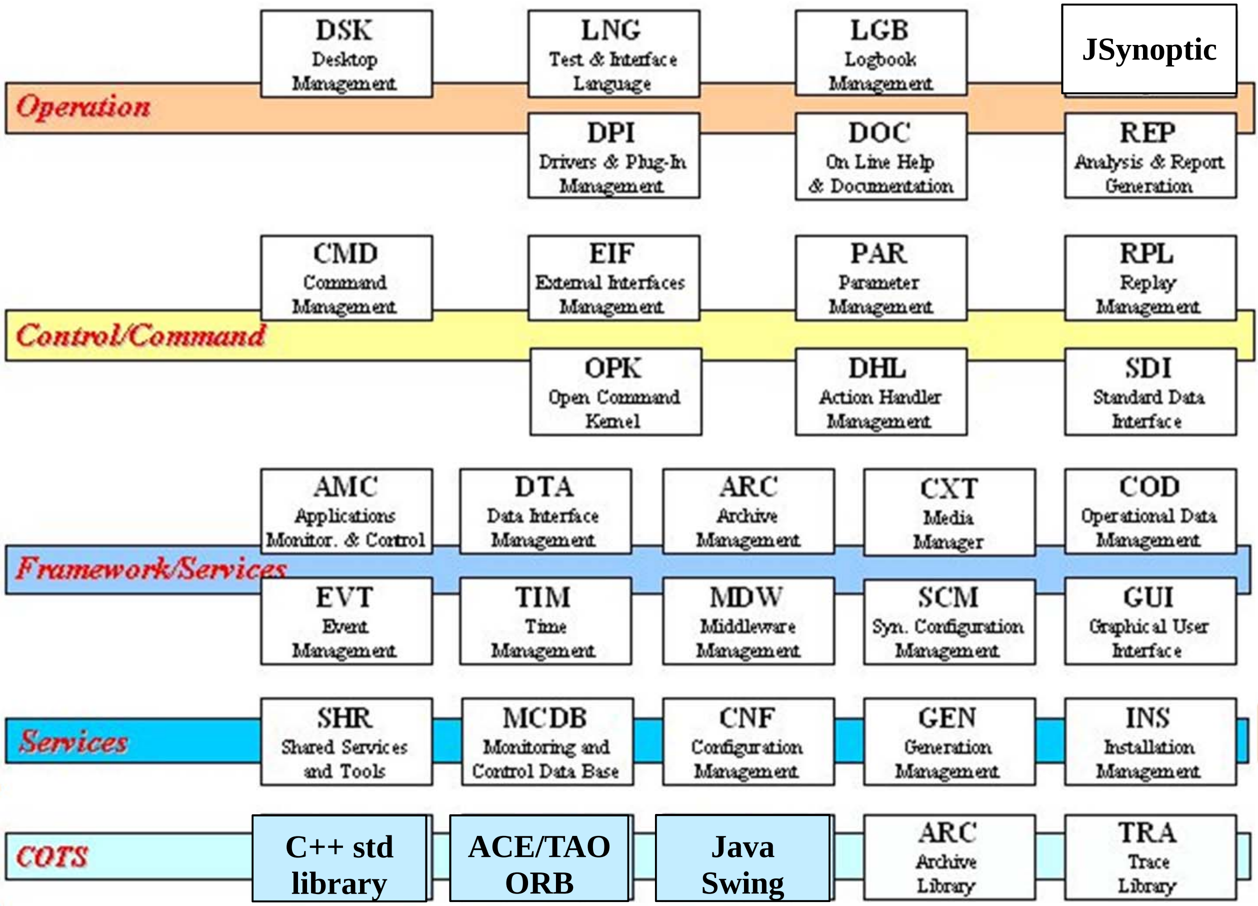
ASTRIUM
AN EADS COMPANY

Outline

- Astrium CCS background
- Lessons learnt
- Prototyping
  - Middleware : ZeroMQ
  - Data modelling : EMF
  - Procedure language : Java

Astrium background : Open Center

- Component framework designed for both EGSE CCS and Spacecraft Control Centers



Operation

DSK Desktop Management
LNG Test & Interface Language
LGB Logbook Management
JSynoptic
DPI Drivers & Plug-In Management
DOC On Line Help & Documentation
REP Analysis & Report Generation

Control/Command

CMD Command Management
EIF External Interfaces Management
PAR Parameter Management
RPL Replay Management
OPK Open Command Kernel
DHL Action Handler Management
SDI Standard Data Interface

Framework/Services

AMC Applications Monitor. & Control
DTA Data Interface Management
ARC Archive Management
CXT Media Manager
COD Operational Data Management
EVT Event Management
TIM Time Management
MDW Middleware Management
SCM Syn. Configuration Management
GUI Graphical User Interface

Services

SHR Shared Services and Tools
MCDB Monitoring and Control Data Base
CNF Configuration Management
GEN Generation Management
INS Installation Management

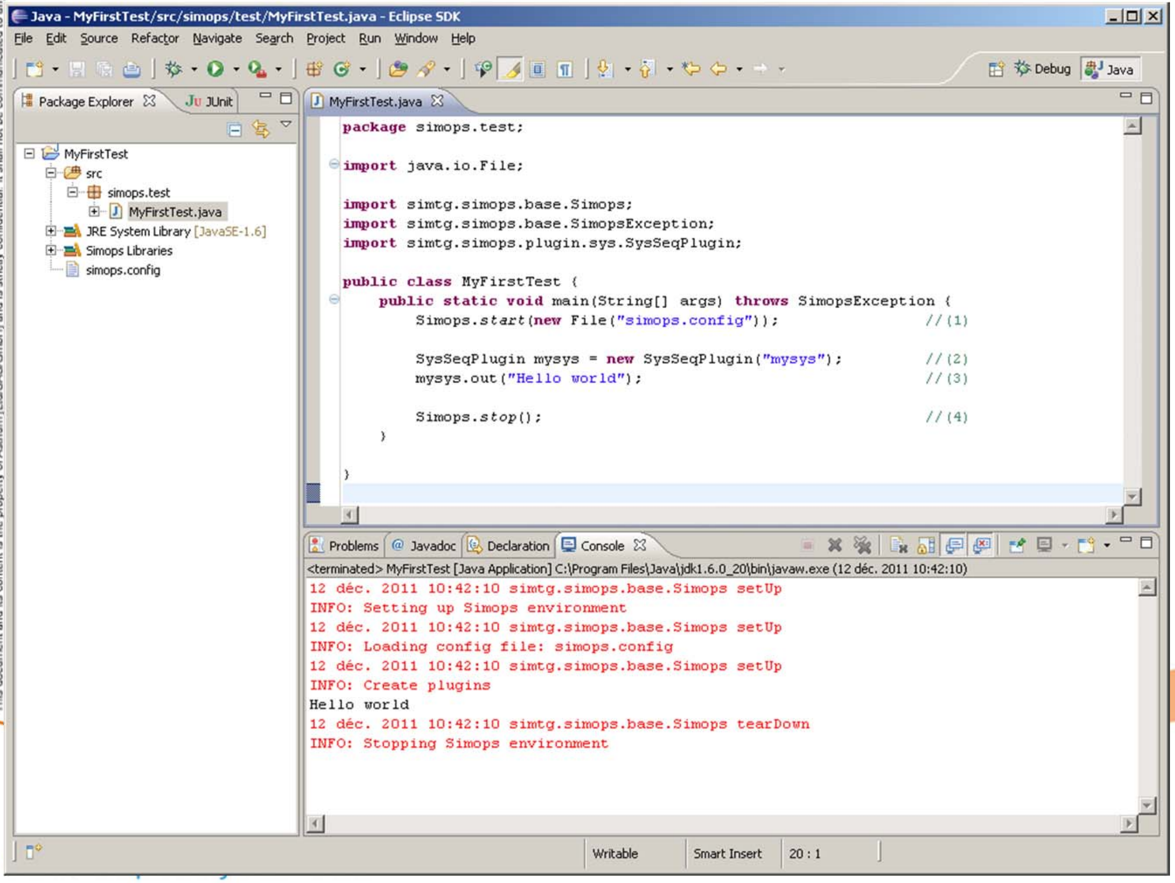
COTS

C++ std library
ACE/TAO ORB
Java Swing
ARC Archive Library
TRA Trace Library

- Designed for high scalability
- Typical configurations made of:
  - A server
  - A redundant server for Control Centers
  - 1 to many client workstations
- Generally more than 40 processes running on several Linux computers

Astrium background : SimOPS

- Lightweight monitoring and control system orginally designed for simulation verification but also used for On Board Software validation (SVF)



```java
package simops.test;

import java.io.File;

import simtg.simops.base.Simops;
import simtg.simops.base.SimopsException;
import simtg.simops.plugin.sys.SysSeqPlugin;

public class MyFirstTest {
    public static void main(String[] args) throws SimopsException {
        Simops.start(new File("simops.config"));                //(1)

        SysSeqPlugin mysys = new SysSeqPlugin("mysys");         //(2)
        mysys.out("Hello world");                               //(3)

        Simops.stop();                                          //(4)
    }

}
```

```
<terminated> MyFirstTest [Java Application] C:\Program Files\Java\jdk1.6.0_20\bin\javaw.exe (12 déc. 2011 10:42:10)
12 déc. 2011 10:42:10 simtg.simops.base.Simops setUp
INFO: Setting up Simops environment
12 déc. 2011 10:42:10 simtg.simops.base.Simops setUp
INFO: Loading config file: simops.config
12 déc. 2011 10:42:10 simtg.simops.base.Simops setUp
INFO: Create plugins
Hello world
12 déc. 2011 10:42:10 simtg.simops.base.Simops tearDown
INFO: Stopping Simops environment
```

- User MMI is also a server
  - One process for the server and one for each running procedure
- Java language for procedures
- Plugin/ OSGi architecture
  - JDT: java editor and debug
  - JSynoptic: user defined displays
  - SimMF: simulation modelling tool

ASTRIUM
AN EADS COMPANY

Lessons learnt (1/3)

- Complex systems are not only difficult to develop or to maintain, they are also difficult to configure by the end user
  - Select architecture enabling very simple but scalable configurations
  - Very simple configuration is a single PC with a minimum number of running processes : 1 or 2
  - Complex architecture is with a redundant server and with many client computers connected to the system

Lessons learnt (2/3)

- Use of de facto standards : some example
  - Procedure language
    - Takes benefits of well known and well supported language : not yet another language
    - OBSW team requires compiled and typed language
    - One solution is to use Java
  - Modelling language
    - UML2 for conceptual data model
    - Ecore (EMF) for technical data model

Lessons learnt (3/3)

- Real time performance is always an issue
  - Strong recommendation is to check if the most efficient solution fits with the other needs
  - A scalable solution shall take benefits of the various configurations
    - Inter thread communication for single process cases
    - Inter process communication for single computer cases

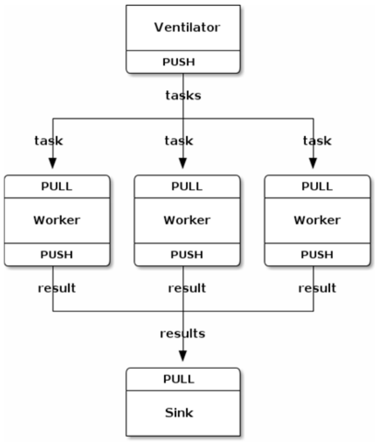
MOMs

- Message Oriented Middleware in place of remote procedure calls (CORBA, RMI, RPC,..)

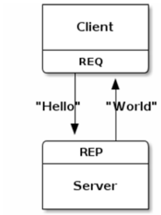
- Most popular solutions
  - RabbitMQ
    - Compliant with AMQP protocol
    - Broker achitecture
    - Easy to use and deploy but some latency due to the central broker
  - ActiveMQ
    - Compliant with JMS standard
    - Broker or P2P architecture
    - Very generic and easy to use but slower than RabbitMQ
  - ZeroMQ
    - Lightweight messaging system
    - Library architecture, no broker
    - More complex to use (low level) but faster than the other ones (takes benefits of the configuration and manages multicast TCP/IP)

ZeroMQ, parallelization

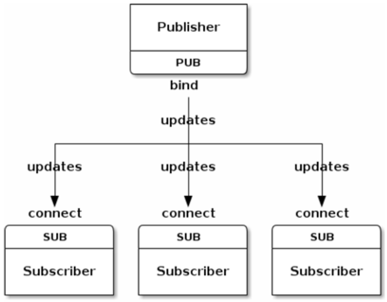
- A generic solution to parallelize computations, less taking care about synchronisation issues



Parallel pipeline with PUSH/PULL



Request and Reply

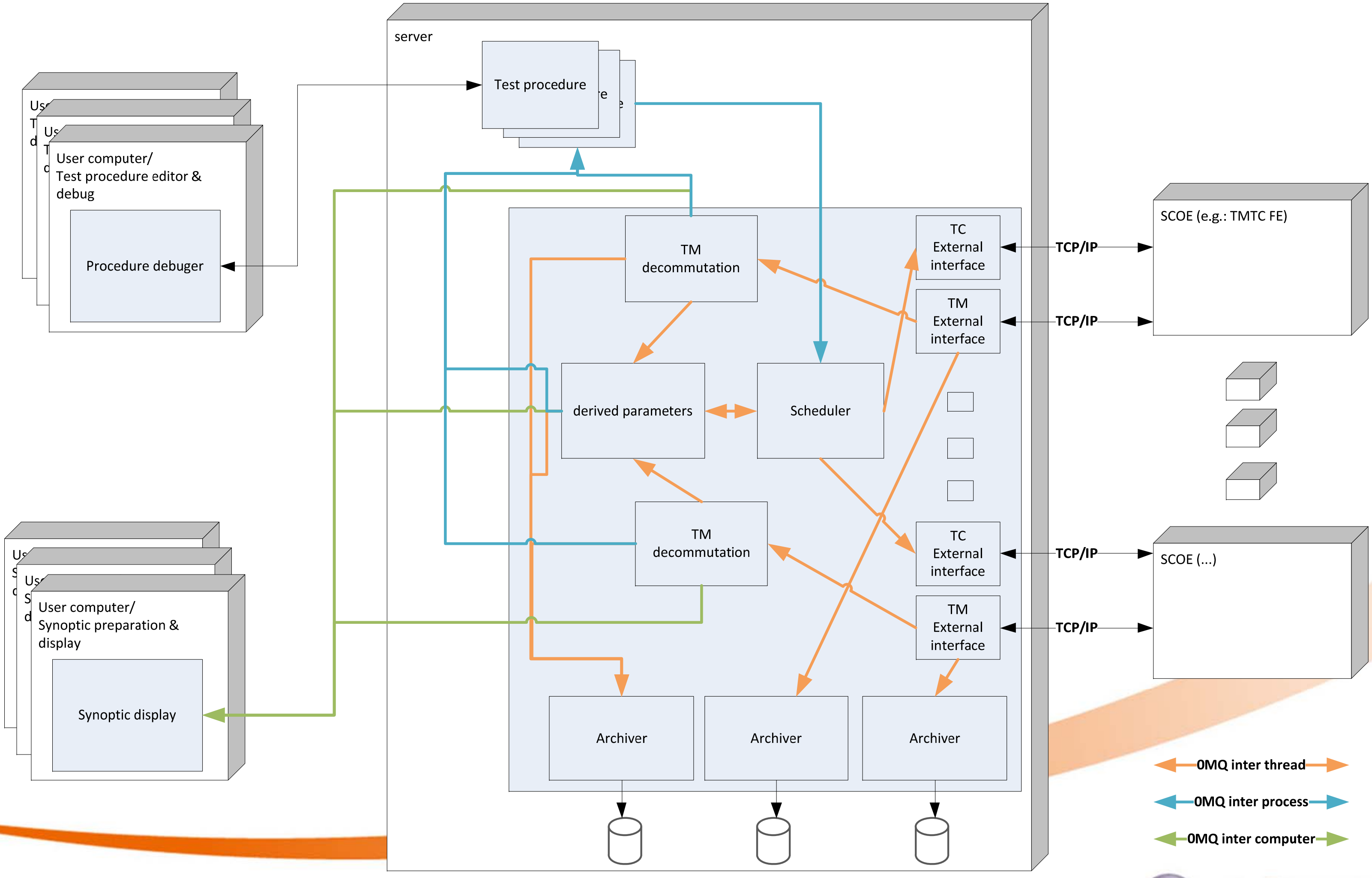


Publish-Subscribe

- To be combined with a marshalling solution where needed (e.g. not for TM packets)

This document and its content is the property of Astrium [Ltd/SAS/GmbH] and is strictly confidential. It shall not be communicated to any third party without the written consent of Astrium [Ltd/SAS/GmbH].

ZeroMQ, analysed architecture



All the space you need

Eclipse Modelling Framework and related technologies

- Application of proven technologies
  - From UML to java
    - UML to eCore
    - Technical data model
    - eCore to Java

      Data persistence : XMI (and binary format)

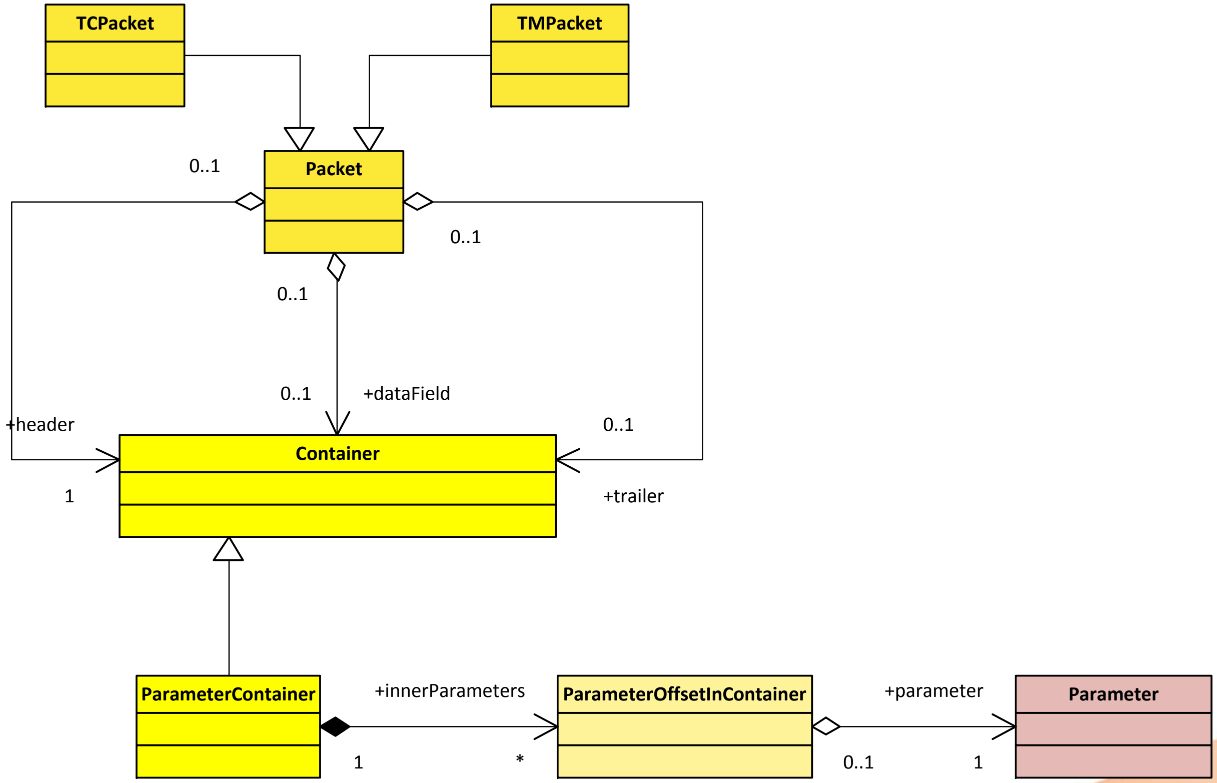
      Simple editors (need for a customized code generation)
  - Use of OCL to define constraints
    - Constraints are embeded in the metamodel to define their scope of application
  - Use of QVTO for model to model transformation
    - Example :

      UML metamodel for MIB files

      UML metamodel for data base

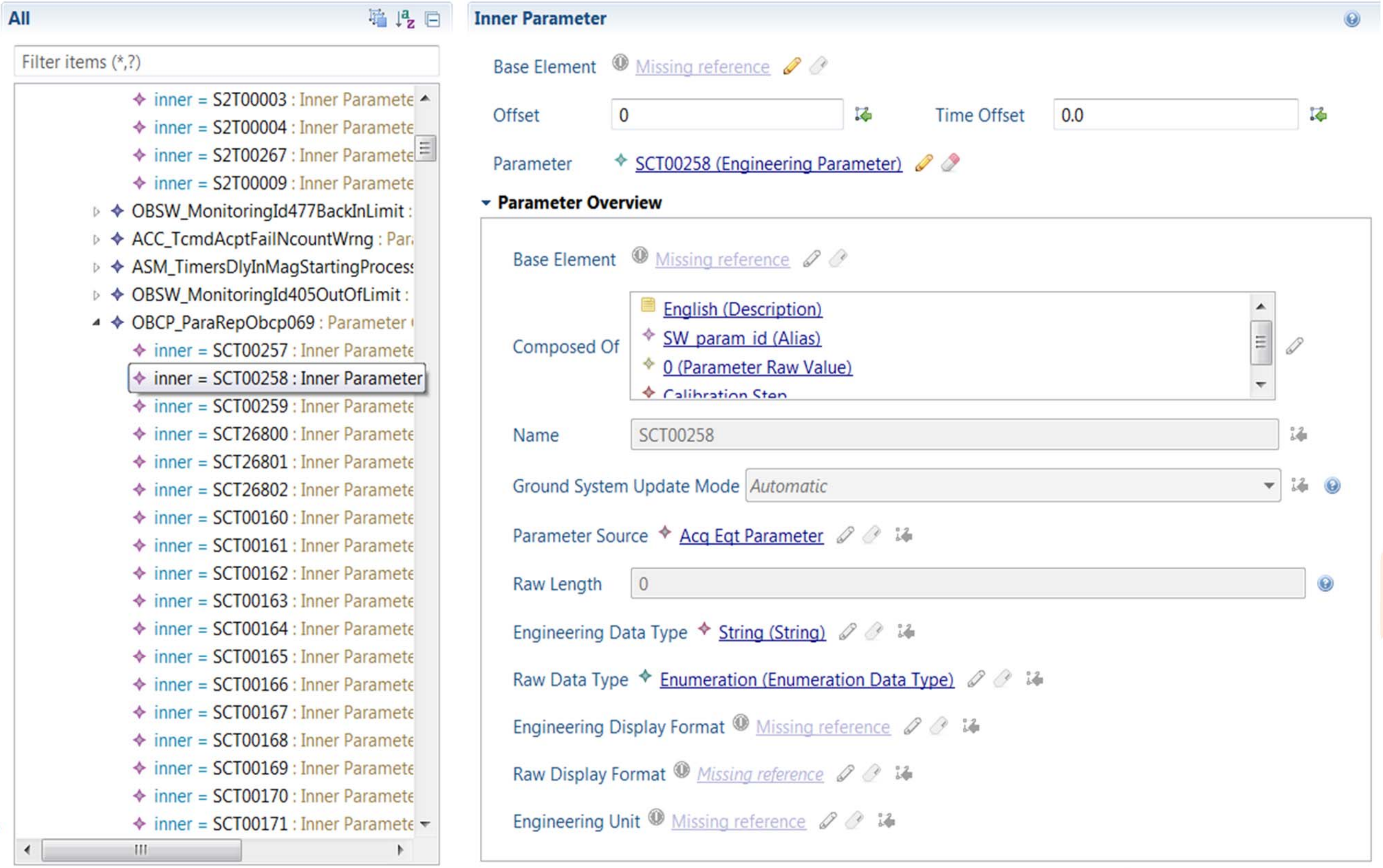
      QVTO to define mapping between the 2

UML TM/TC Meta model example



All the space you need

EMF TM/TC editor example



All

Filter items (*,?)

- inner = S2T00003 : Inner Paramete
- inner = S2T00004 : Inner Paramete
- inner = S2T00267 : Inner Paramete
- inner = S2T00009 : Inner Paramete
- OBSW_MonitoringId477BackInLimit :
- ACC_TcmdAcptFailNcountWrng : Para
- ASM_TimersDlyInMagStartingProcess
- OBSW_MonitoringId405OutOfLimit :
- OBCP_ParaRepObcp069 : Parameter
- inner = SCT00257 : Inner Paramete
- inner = SCT00258 : Inner Parameter
- inner = SCT00259 : Inner Paramete
- inner = SCT26800 : Inner Paramete
- inner = SCT26801 : Inner Paramete
- inner = SCT26802 : Inner Paramete
- inner = SCT00160 : Inner Paramete
- inner = SCT00161 : Inner Paramete
- inner = SCT00162 : Inner Paramete
- inner = SCT00163 : Inner Paramete
- inner = SCT00164 : Inner Paramete
- inner = SCT00165 : Inner Paramete
- inner = SCT00166 : Inner Paramete
- inner = SCT00167 : Inner Paramete
- inner = SCT00168 : Inner Paramete
- inner = SCT00169 : Inner Paramete
- inner = SCT00170 : Inner Paramete
- inner = SCT00171 : Inner Paramete

Inner Parameter

Base Element: Missing reference

Offset: 0

Time Offset: 0.0

Parameter: SCT00258 (Engineering Parameter)

Parameter Overview

Base Element: Missing reference

Composed Of:
- English (Description)
- SW_param_id (Alias)
- 0 (Parameter Raw Value)
- Calibration Step

Name: SCT00258

Ground System Update Mode: Automatic

Parameter Source: Acq Eqt Parameter

Raw Length: 0

Engineering Data Type: String (String)

Raw Data Type: Enumeration (Enumeration Data Type)

Engineering Display Format: Missing reference

Raw Display Format: Missing reference

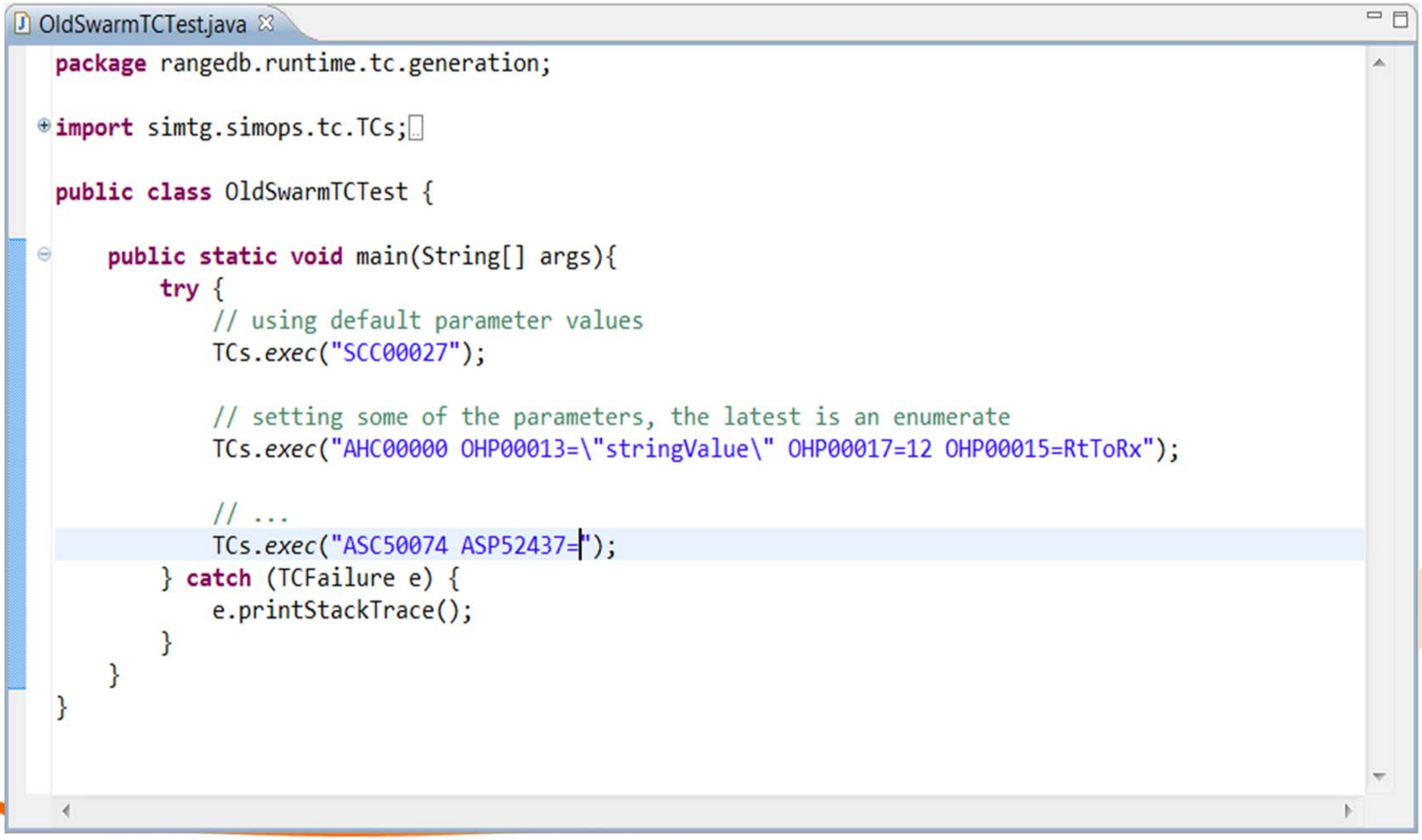
Engineering Unit: Missing reference

Java

- Current solution at Astrium Satellites to validate the On Board SoftWare
- Well known language, easy to learn, state of the art development environment (editor and debugger)
- Compliant with ECSS E70 32
  - Specific features are implemented using a base class
  - Compiled and typed
- Compliant with EGS-CC requirements for debugging and command retry
- And much more:
  - Scripting language interface
  - Annotations

Current approach: TCs are intrepreted

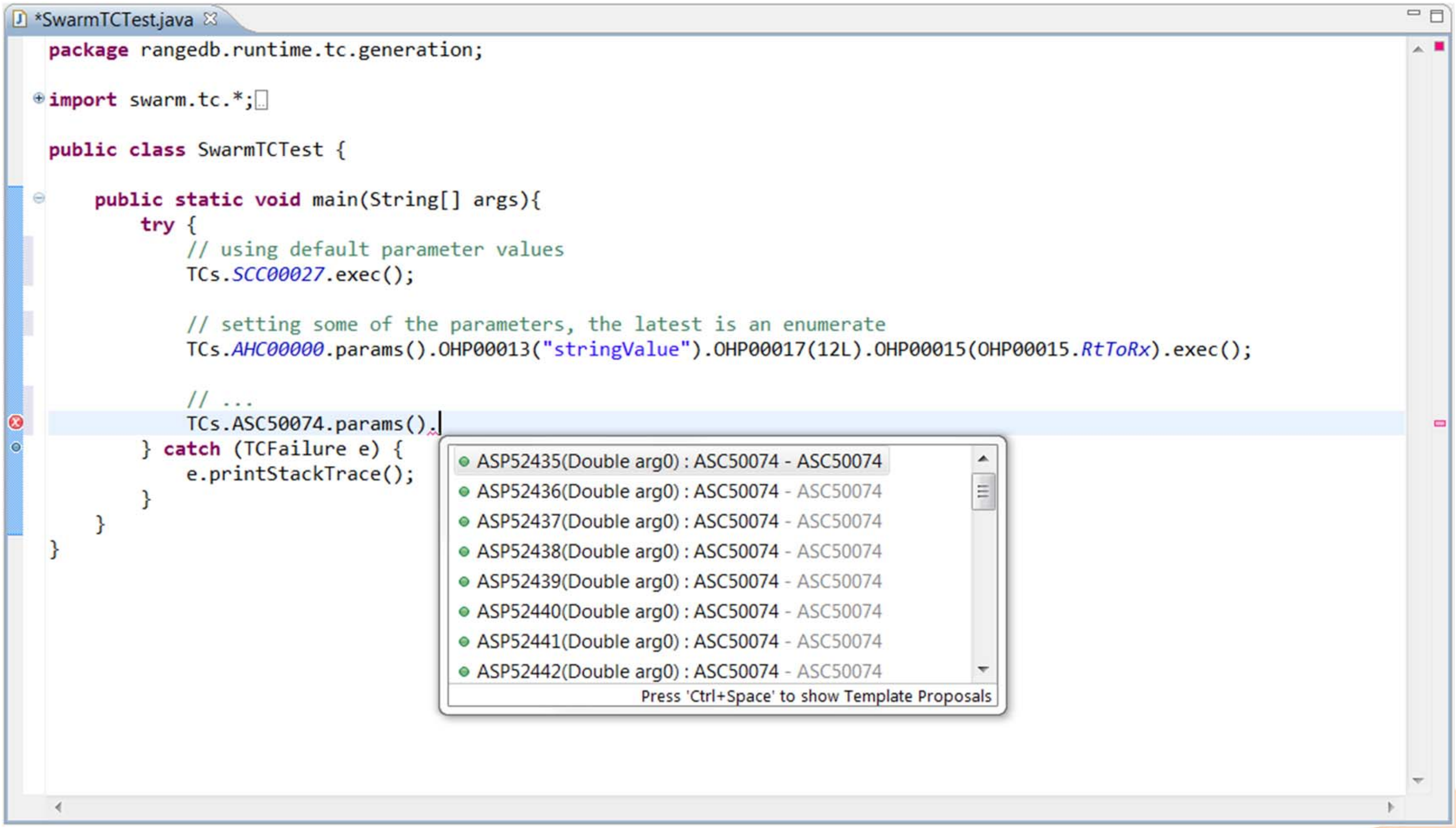
- Based on MIB files information



```java
package rangedb.runtime.tc.generation;

import simtg.simops.tc.TCs;

public class OldSwarmTCTest {

    public static void main(String[] args){
        try {
            // using default parameter values
            TCs.exec("SCC00027");

            // setting some of the parameters, the latest is an enumerate
            TCs.exec("AHC00000 OHP00013=\"stringValue\" OHP00017=12 OHP00015=RtToRx");

            // ...
            TCs.exec("ASC50074 ASP52437=");
        } catch (TCFailure e) {
            e.printStackTrace();
        }
    }
}
```

All the space you need

New approach, TCs interface is compiled



```
package rangedb.runtime.tc.generation;

import swarm.tc.*;

public class SwarmTCTest {

    public static void main(String[] args){
        try {
            // using default parameter values
            TCs.SCC00027.exec();

            // setting some of the parameters, the latest is an enumerate
            TCs.AHC00000.params().OHP00013("stringValue").OHP00017(12L).OHP00015(OHP00015.RtToRx).exec();

            // ...
            TCs.ASC50074.params().
        } catch (TCFailure e) {
            e.printStackTrace();
        }
    }
}
```

```
ASP52435(Double arg0) : ASC50074 - ASC50074
ASP52436(Double arg0) : ASC50074 - ASC50074
ASP52437(Double arg0) : ASC50074 - ASC50074
ASP52438(Double arg0) : ASC50074 - ASC50074
ASP52439(Double arg0) : ASC50074 - ASC50074
ASP52440(Double arg0) : ASC50074 - ASC50074
ASP52441(Double arg0) : ASC50074 - ASC50074
ASP52442(Double arg0) : ASC50074 - ASC50074
                         Press 'Ctrl+Space' to show Template Proposals
```

- Example of implementation using ASM byte code generator and RangeDB
  - <10 seconds to generate 3000 TCs
  - 3KB per TC (average)

All the space you need

ASTRIUM
AN EADS COMPANY

Conclusion

- Activities are to support Astrium activities in EGS-CC Systems Engineering Team
  - All results to be contributed to EGS-CC

- Prototypes confirm the benefits of these three technologies

- Potential next domains to be analyzed:
  - Component Framework (OSGI + ?)
  - Data archiving
  - User defined displays
  - Scripting language
  - Service integration platforms

Thanks for your attention

Claude Cazenave, Harald Eisenmann

ACE8

ASTRIUM Satellites

Email: claude.cazenave@astrium.eads.net

harald.eisenmann@astrium.eads.net

All the space you need