

STANDARD-BASED AUTOMATION: SCALABILITY, FLEXIBILITY AND EXCHANGE FOR LONG TERM MISSIONS

Nieves Salor Moral - Vitrociset Belgium

SESP 2015 – Noordwijk, March 2015



Why?

WHY?

- Communication GS
 - Required for
 - Operating the spacecraft, performing tasks
 - Sending TCs
 - Receiving TMs
 - Critical
 - Failures have grave consequences
 - Complexity increases exponentially.
 - Technological gap in SW technologies and architectures

Re-engineer & Upgrade SW

WHY?

- Communication GS
- Increase of involved systems
 - Reuse/Tailoring of systems
 - Licensing problems/ tight relation with specific requirements.
 - Heritage/ commercial aspects
 - Reinventing the wheel for each mission
 - Lack of interoperability
- Development of EGS-CC

Generalize and Interface Systems

WHY?

- Communication GS
- Increase of involved systems
- Need for Automation capabilities
 - Use of Procedures
 - Syntax vs. functionality
 - Multiple applicable standards
 - Understanding of the standards
- Improve HMI and port them to new devices

**Formalize Procedure
Methods**

WHY?

- Communication GS
- Increase of involved systems
- Need for Automation capabilities



- Re-engineer automation systems → SOA, OSGI, HMIs
- Conceptualization of procedure requirements
- Develop of interfaced plug-play automation framework

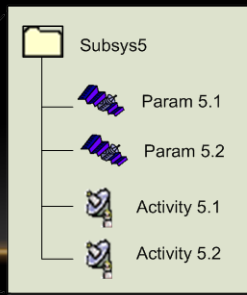
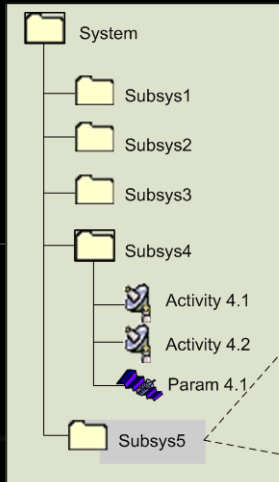
What?

WHAT? – INFORMATION TO BE HANDLED

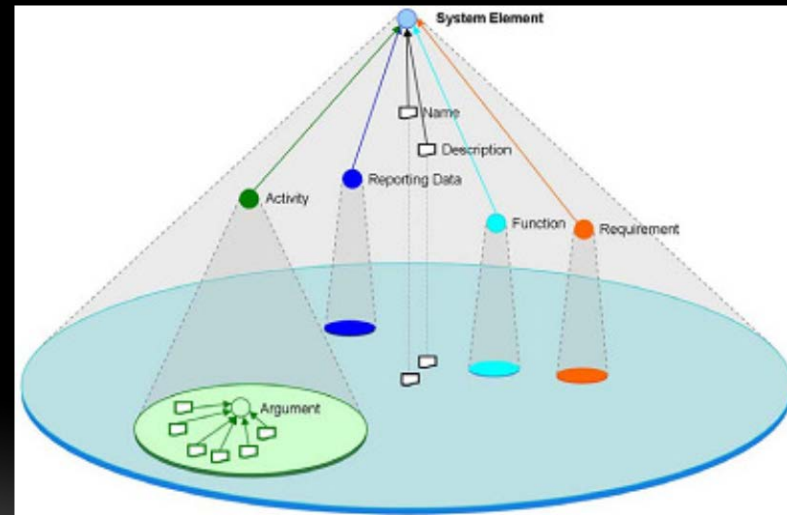
- What entails automation?
 - Execute simple/complex tasks without involving the user at every step obtaining deterministic behaviors.
 - Handle data and erroneous situations following a standardized approach
 - Scheduling tasks depending on data and/or time conditions.
- What is needed for automation?
 - Data Model to be handled
 - Automation Language (i.e. Procedure Language) Editor and compiler
 - Execution Engine

WHAT? – MAIN CONCEPTS (I)

- What entails automation?
- What concepts have to be handled?
 - ECSS-E-ST-70-31 (SSM)

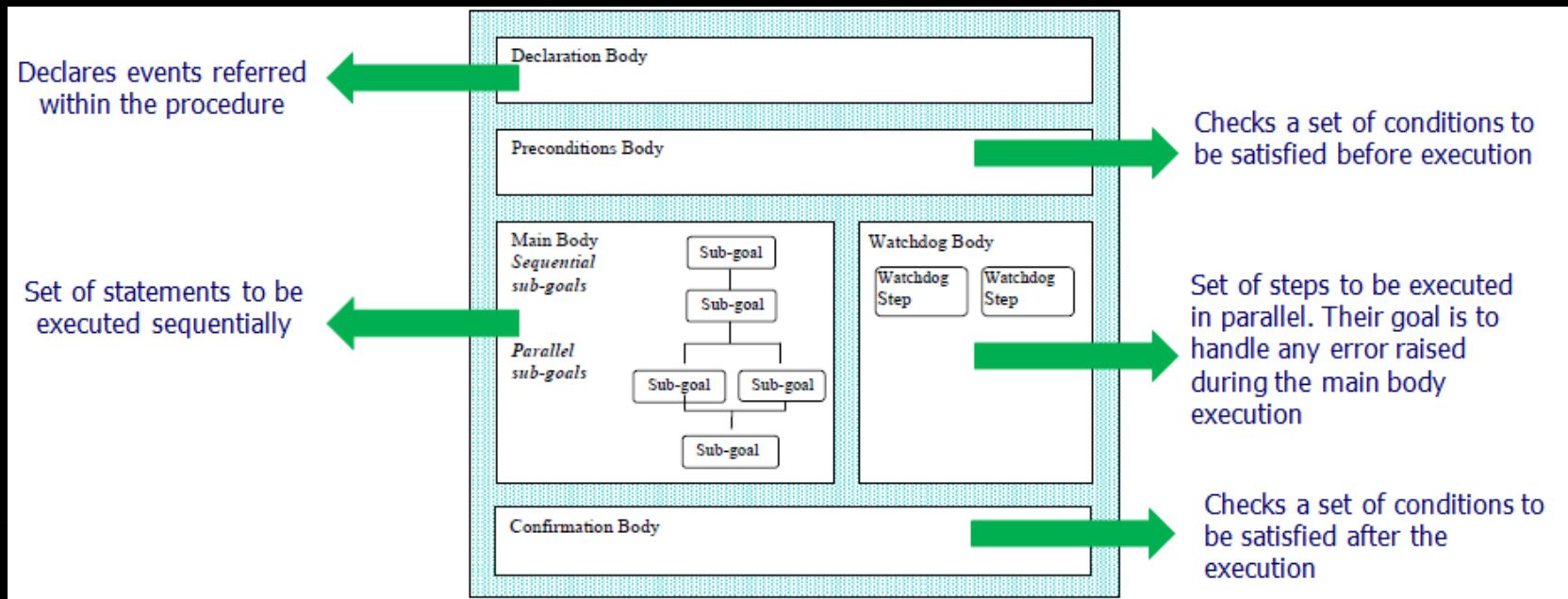


In EGS-CC:
SSM ↔ MCM



WHAT? – MAIN CONCEPTS (II)

- What entails automation?
- What concepts have to be handled?
 - ECSS-E-ST-70-32



WHAT? – MAIN CONCEPTS (III)

- What entails automation?
- What concepts have to be handled?
 - In EGS-CC, requirements of ECSS-E-ST-70-32



WHAT? – SCALABILITY PURPOSES

- What entails automation?
- What concepts have to be handled?
- What is the long-term durability to be provided?
 - Allow future updates in the standards
 - Allow new automation standards
 - Develop new graphical aspects
 - Easy portability of the existing features
 - Loose coupling of new features

How?

HOW? – REQUIREMENTS AND TARGET CLASSIFICATION



Business support

Reduce human errors

Facilitate standardization of procedures

Time saving

Increase the level of automation



User support

Interfaceable with legacy Systems

Execute all types of Activities

Graphical Editor Interface

Flexible



Distributed Architecture

Open source based application

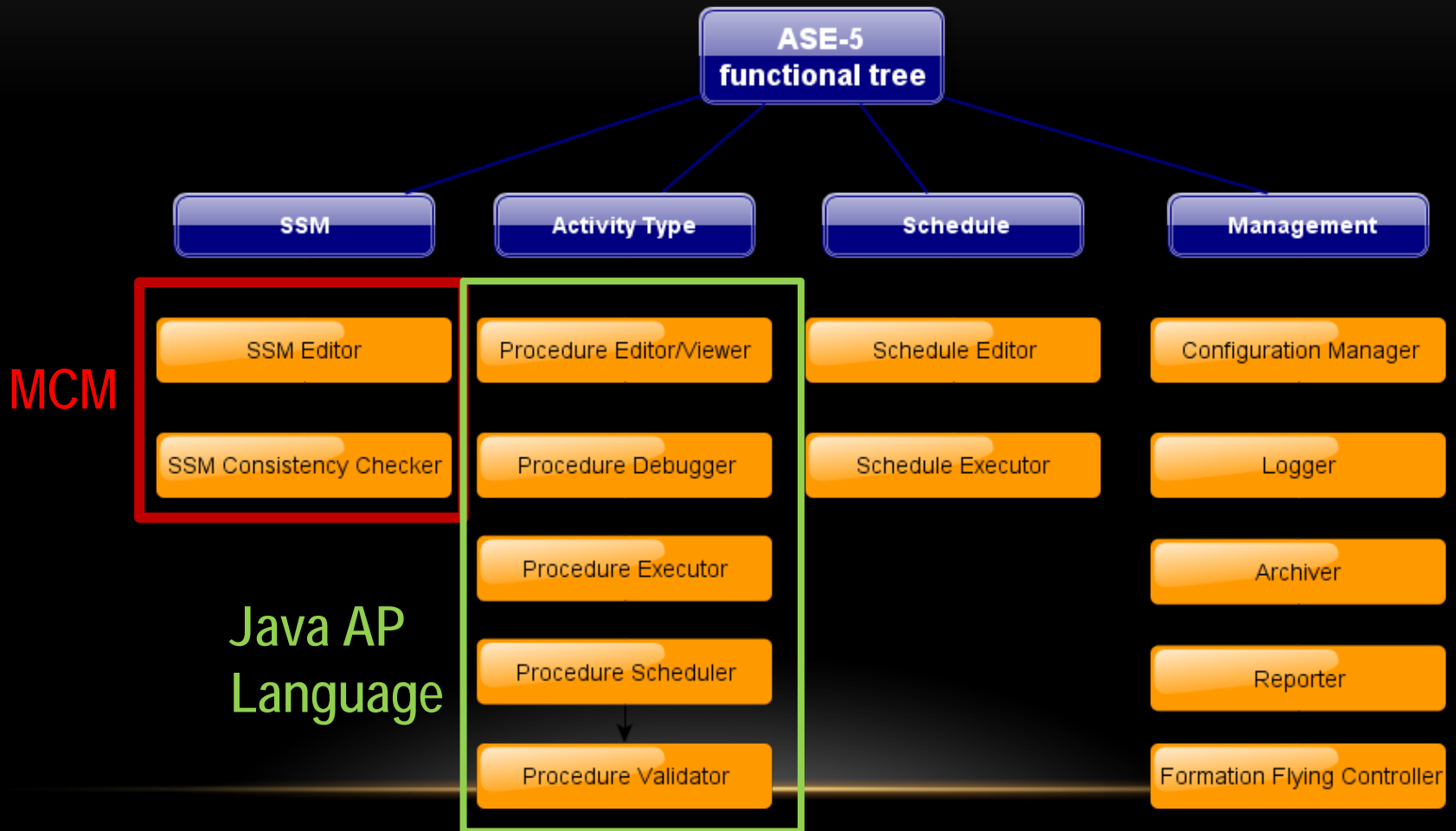
Eclipse/Java Developed

Platform Independent

Multi-domain

Service-based

HOW? – AUTOMATION FEATURE BRANCHES



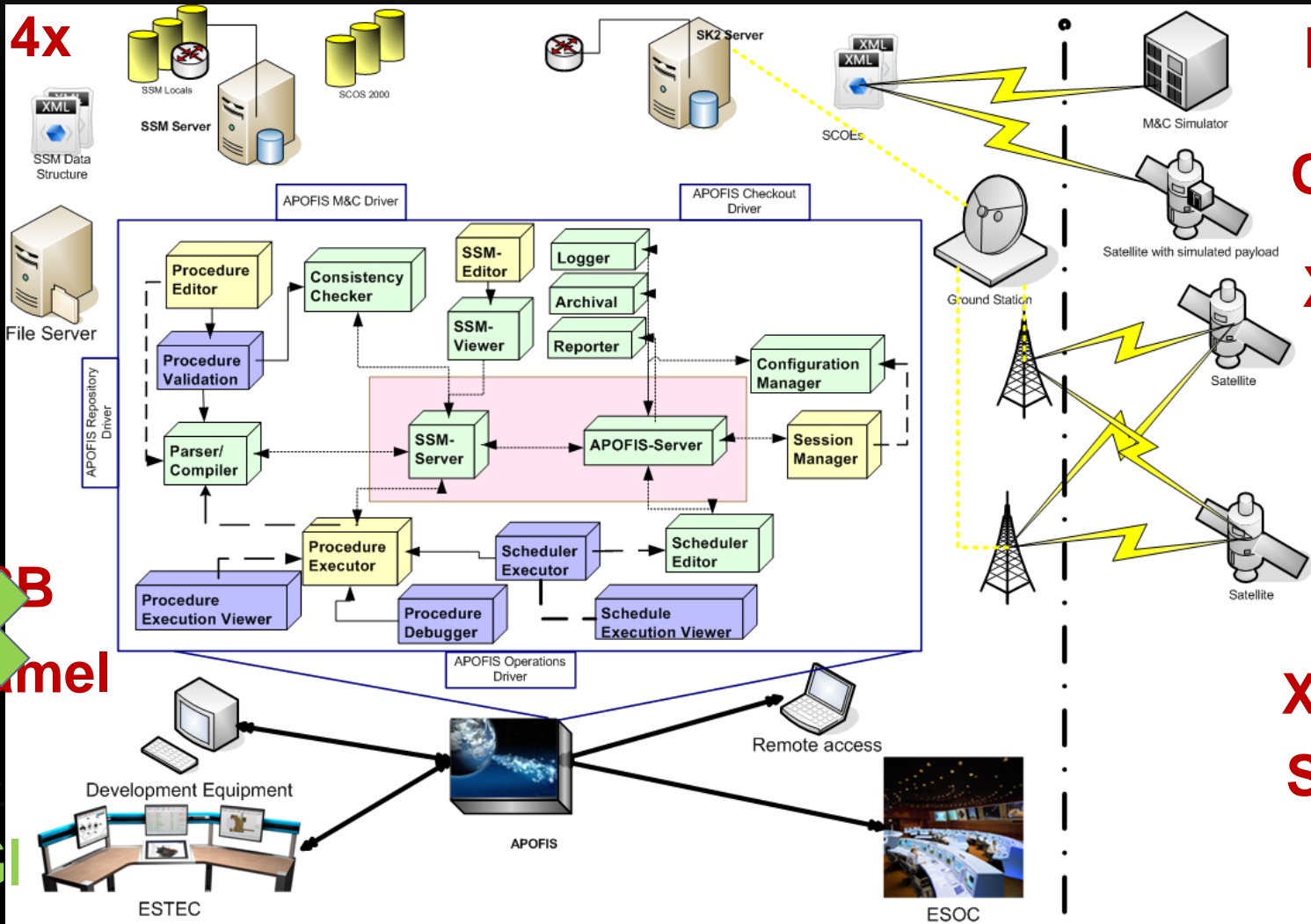
HOW? – ARCHITECTURE MODULES

Eclipse 4x

JAVA

SOA

~~ESB~~
Camel
↓
OSGI



EMF

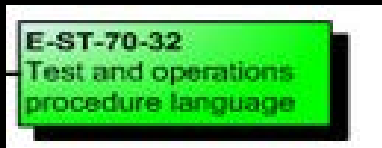
ORM

Xtext

XSD

SQL

HOW? – PROCEDURE HARMONIZATION (I)

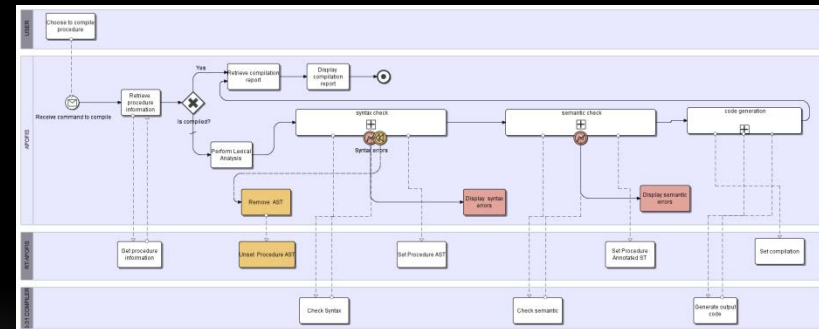
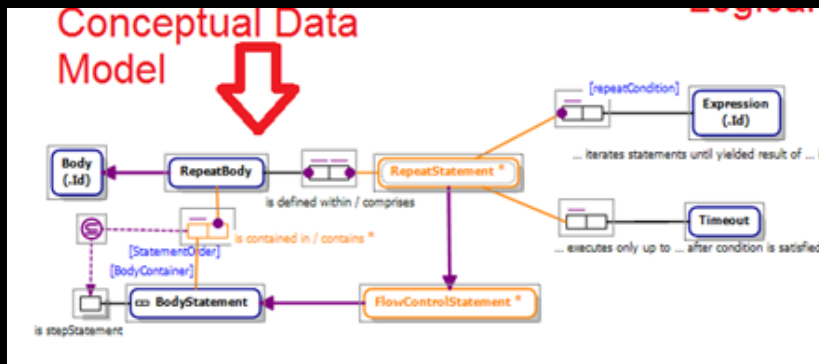


HOW? – PROCEDURE HARMONIZATION (II)

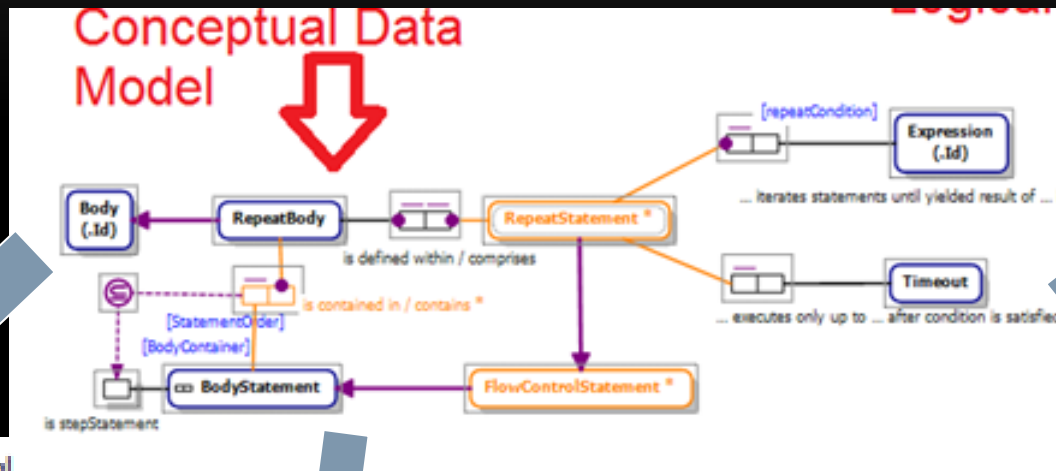


ORM

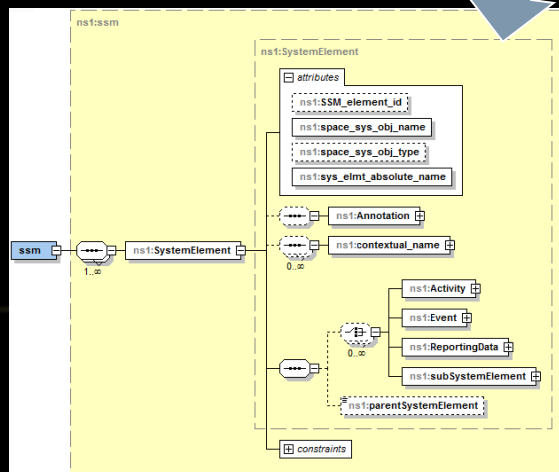
BPM



HOW? – PROCEDURE HARMONIZATION (III)

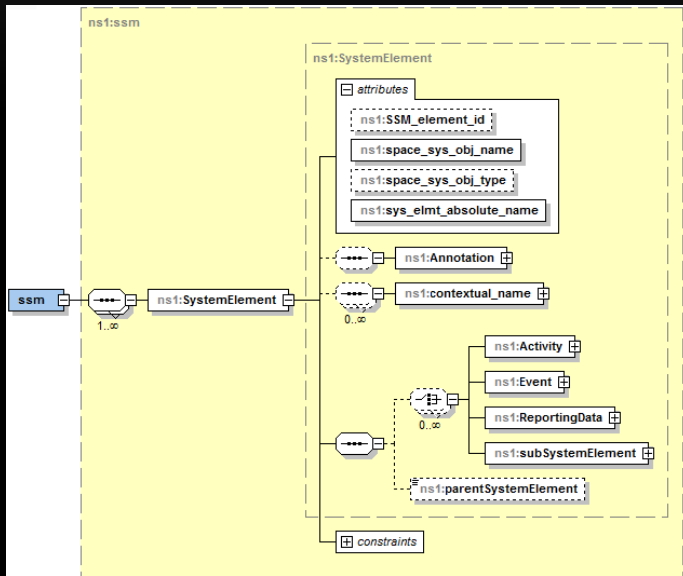


- 20130611-70-32 Formal.MySQL.sql
- 20130611-70-32 Formal.Oracle.sql
- 20130611-70-32 Formal.PostgreSQL.sql



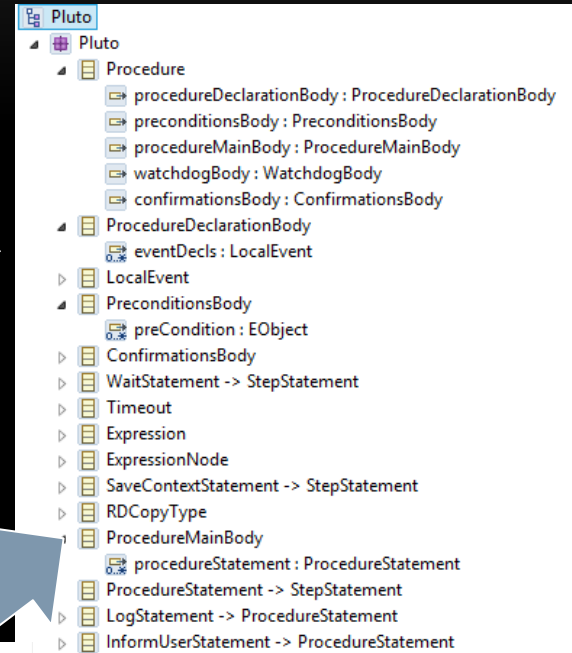
BodyStatement is stepStatement.
 In each population of BodyStatement is stepStatement, each BodyStatement occurs at most once.
 RepeatBody contains BodyStatement.
 Each RepeatBody contains some BodyStatement.
 Each BodyStatement is contained in at most one RepeatBody.
 It is possible that some RepeatBody contains more than one BodyStatement.
 Derivation Note: Fully derived from the entries in the repeatBody.
 If some BodyStatement is contained in some RepeatBody then that BodyStatement is stepStatement.
 Body is an entity type.
 Reference Scheme: Body has Body_Id.
 Reference Mode: .Id.

HOW? – PROCEDURE HARMONIZATION (IV)



EMF

Xtext



**Xmi
DB**

```

Procedure returns Procedure:
  '<PROCEDURE>' (procedureDeclarationBody = Procedure_declaration_body)? (precondition = Precondition)

Procedure_declaration_body returns ProcedureDeclarationBody:
  '<DECLARE>' (eventDecls += EventDeclaration)+ '</DECLARE>';

EventDeclaration returns LocalEvent:
  '<EVENT>' name = ID ('<DESCRIPTION>' eventDescr = STRING '</DESCRIPTION>')? '</EVENT>';

PreconditionsBody returns PreconditionsBody:
  '<PRECONDITIONS>' (preCondition += (Expression|Wait_statement))+ '</PRECONDITIONS>';

ConfirmationsBody: '<CONFIRM>' (confirmCondition += (Expression|Wait_statement))+ '</CONFIRM>';

Wait_statement returns WaitStatement:
  '<WAIT>' (('<EVENT>' event=[LocalEvent] '</EVENT>') | expression=Expression) (saveContextStmnt = SaveContextStatement)

Timeout returns Timeout:
  '<TIMEOUT>' expression=Expression ('<RAISE_EVENT>' event = [LocalEvent] '</RAISE_EVENT>')? '</TIMEOUT>';
    
```

Where?

WHERE? – SPACECRAFT INFORMATION (SSM)

The screenshot displays the Activity IDE interface for 'APPLICATION v1.0'. The main workspace is the SSM Editor, showing a tree view of the SSM structure. The selected object is '<activity> Procedure calculateStatus'. The Properties window on the right shows the following details:

| Property | Value |
|--------------------|---------------------------|
| Act Descr | calculate start-up values |
| Activity Phase | AIT, Operations |
| Act Version | 1 |
| Name | calculateStatus |
| Proc Script | testProcedure2.pluto |
| Space Sys Obj Type | activity |
| SSM Element Id | ASE5-19 |

At the bottom, the Console window is empty, and the Selected Object is '<activity> Procedure calculateStatus'. The SSM Server Status is also visible at the bottom right.

WHERE – PROCEDURE EDITORS AND COMPILERS

The screenshot displays the Activity IDE interface for 'Activity APPLICATION v1.0'. The main editor shows a procedure definition for 'testProcedure5.pluto' with the following structure:

```
<PROCEDURE>
  <DECLARE>
    <EVENT> antenaOff
      <DESCRIPTION> "describes the error of the antena" </DESCRIPTION>
    </EVENT>
  </DECLARE>
  <MAIN>
    <INIT_AND_CONFIRM_STEP> stepProblematic
      <STEP>
        <DECLARE>
          <VARIABLE> antenaStatus
            boolean
          </VARIABLE>
          <EVENT> signalReceived </EVENT>
        </DECLARE>
        <MAIN>
          <ASSIGN_STATEMENT>
            <VARIABLE> antenaStatus </VARIABLE>
            <EXPRESSION>
              <REL_EXPRESSION>
                <TERM>
                  <PRODUCT>
                    <FACTOR>
                      <CONSTANT> false </CONSTANT>

```

The Properties window on the right shows a tree view of the procedure structure, including 'Procedure Declaration Body', 'Local Event=antenaOff', 'Procedure Main Body', 'Initiate & confirm step stepProblematic', 'Step Declaration Body', 'Variables', 'Events', 'Step Main Body', 'Assignment', 'Condition', 'Log', 'Wait Statement', 'Waiting Events:signalReceived', 'Timeout', 'Watchdog Body', and 'Initiate & confirm step lookingForAntenaOff'.

The Console window at the bottom left shows the following output:

```
ActivityExecutorConsole
  procname:triggerWakeUp
creating PLUTO editor for: switchonRadar
SSM belongs to project:SessionProject1394203423790
short procname:testProcedure5.pluto
procname:switchonRadar
```

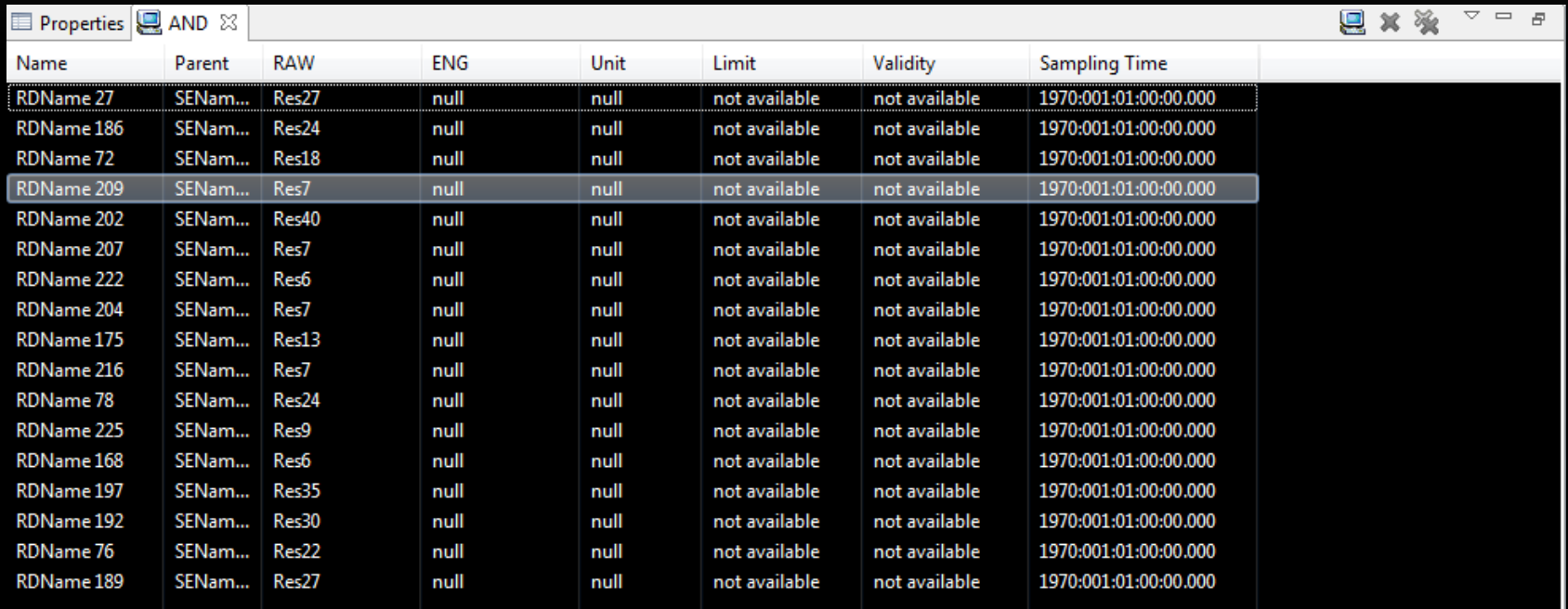
A code snippet at the bottom right shows a 'main' function with an error message:

```
main{
  initiate{
    Couldn't resolve reference to Activity 'sys1.sys2.Procedure2'.
  }end initiate;
  initiate and confirm{
    sys1.sys2.Procedure2
    refer by "asda" in case{
```

WHERE? – ACTIVITY EXECUTION ENGINE

| Statement | Execution Status | Confirmation Status | Initialization Time | Completion Time |
|--|------------------|---------------------|-----------------------|-----------------------|
| ▲ PROCEDURE | COMPLETED | CONFIRMED | 2014:066:13:43:02.341 | 2014:066:14:29:08.615 |
| PROC_DECLARATION_BODY | COMPLETED | CONFIRMED | 2014:066:13:43:02.352 | 2014:066:13:43:02.375 |
| ▶ PROC_MAIN_BODY | COMPLETED | CONFIRMED | 2014:066:13:43:02.421 | 2014:066:14:29:08.602 |
| ▶ PROCEDURE | COMPLETED | CONFIRMED | 2014:066:14:29:28.170 | 2014:066:14:36:17.806 |
| ▶ PROCEDURE | COMPLETED | CONFIRMED | 2014:066:14:36:59.023 | 2014:066:14:39:15.776 |
| ▲ PROCEDURE | COMPLETED | CONFIRMED | 2014:066:14:39:55.666 | 2014:066:14:41:55.175 |
| PROC_DECLARATION_BODY | COMPLETED | CONFIRMED | 2014:066:14:39:55.667 | 2014:066:14:39:55.675 |
| ▲ PROC_MAIN_BODY | COMPLETED | CONFIRMED | 2014:066:14:39:55.680 | 2014:066:14:41:55.171 |
| ▲ INIT_AND_CONFIRM_STEP_STMT stepProblemat | COMPLETED | CONFIRMED | 2014:066:14:39:55.686 | 2014:066:14:41:55.165 |
| STEP_DECL_BODY | COMPLETED | CONFIRMED | 2014:066:14:39:55.689 | 2014:066:14:39:55.696 |
| ▲ STEP_MAIN_BODY | COMPLETED | NOT_CONFIRMED | 2014:066:14:39:55.710 | 2014:066:14:41:55.139 |
| ASSIGN_STMT | COMPLETED | CONFIRMED | 2014:066:14:39:55.728 | 2014:066:14:39:55.741 |
| LOG_STMT | COMPLETED | CONFIRMED | 2014:066:14:39:55.747 | 2014:066:14:39:55.756 |
| ▲ WAIT_STMT | COMPLETED | NOT_CONFIRMED | 2014:066:14:39:55.761 | 2014:066:14:41:55.087 |
| SAVE_CONTEXT_STMT | NOT_INITIALISED | NOT_AVAILABLE | 1970:001:01:00:00.000 | 1970:001:01:00:00.000 |
| LOG_STMT | COMPLETED | CONFIRMED | 2014:066:14:41:55.128 | 2014:066:14:41:55.135 |
| ▶ WATCHDOG_BODY | COMPLETED | CONFIRMED | 2014:066:14:39:55.708 | 2014:066:14:41:55.152 |
| ▲ PROCEDURE | RUNNING | NOT_AVAILABLE | 2014:066:14:54:44.278 | 1970:001:01:00:00.000 |
| PROC_DECLARATION_BODY | COMPLETED | CONFIRMED | 2014:066:14:54:44.279 | 2014:066:14:54:44.284 |
| ▲ PROC_MAIN_BODY | RUNNING | NOT_AVAILABLE | 2014:066:14:54:44.288 | 1970:001:01:00:00.000 |
| ▲ INIT_AND_CONFIRM_STEP_STMT stepProblemat | RUNNING | NOT_AVAILABLE | 2014:066:14:54:44.297 | 1970:001:01:00:00.000 |
| STEP_DECL_BODY | COMPLETED | CONFIRMED | 2014:066:14:54:44.310 | 2014:066:14:54:44.323 |
| ▶ STEP_MAIN_BODY | RUNNING | NOT_AVAILABLE | 2014:066:14:54:44.335 | 1970:001:01:00:00.000 |
| ▶ WATCHDOG_BODY | RUNNING | NOT_AVAILABLE | 2014:066:14:54:44.336 | 1970:001:01:00:00.000 |

WHERE? – MONITORING TELEMETRY DISPLAY



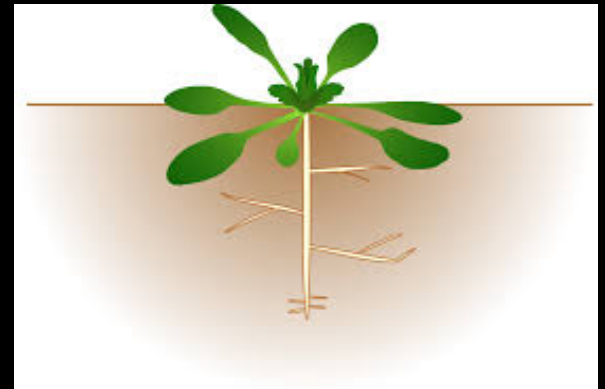
The screenshot shows a software window titled 'Properties' with a search filter 'AND'. It displays a table of telemetry data. The table has the following columns: Name, Parent, RAW, ENG, Unit, Limit, Validity, and Sampling Time. The row for 'RDName 209' is highlighted in blue.

| Name | Parent | RAW | ENG | Unit | Limit | Validity | Sampling Time |
|------------|----------|-------|------|------|---------------|---------------|-----------------------|
| RDName 27 | SENam... | Res27 | null | null | not available | not available | 1970:001:01:00:00.000 |
| RDName 186 | SENam... | Res24 | null | null | not available | not available | 1970:001:01:00:00.000 |
| RDName 72 | SENam... | Res18 | null | null | not available | not available | 1970:001:01:00:00.000 |
| RDName 209 | SENam... | Res7 | null | null | not available | not available | 1970:001:01:00:00.000 |
| RDName 202 | SENam... | Res40 | null | null | not available | not available | 1970:001:01:00:00.000 |
| RDName 207 | SENam... | Res7 | null | null | not available | not available | 1970:001:01:00:00.000 |
| RDName 222 | SENam... | Res6 | null | null | not available | not available | 1970:001:01:00:00.000 |
| RDName 204 | SENam... | Res7 | null | null | not available | not available | 1970:001:01:00:00.000 |
| RDName 175 | SENam... | Res13 | null | null | not available | not available | 1970:001:01:00:00.000 |
| RDName 216 | SENam... | Res7 | null | null | not available | not available | 1970:001:01:00:00.000 |
| RDName 78 | SENam... | Res24 | null | null | not available | not available | 1970:001:01:00:00.000 |
| RDName 225 | SENam... | Res9 | null | null | not available | not available | 1970:001:01:00:00.000 |
| RDName 168 | SENam... | Res6 | null | null | not available | not available | 1970:001:01:00:00.000 |
| RDName 197 | SENam... | Res35 | null | null | not available | not available | 1970:001:01:00:00.000 |
| RDName 192 | SENam... | Res30 | null | null | not available | not available | 1970:001:01:00:00.000 |
| RDName 76 | SENam... | Res22 | null | null | not available | not available | 1970:001:01:00:00.000 |
| RDName 189 | SENam... | Res27 | null | null | not available | not available | 1970:001:01:00:00.000 |

So...what now?

SO...FUTURE WORK

- Support for non-space automation standards
- Support for programming language scripts
- Developing a space HMI library
- Increasing the set of features
 - Scheduling
 - User Management
 - Security Management



SO...FUTURE WORK....TOWARDS THE EGS-CC

- Adaptation between the ASE5 and EGS-CC:
 - DSLs to/from EGS-CC AP Format
 - Driver between ASE-5 and EGS-CC engines
 - Router between SSM and MCM models
 - Transformation from SOA WS to OSGI Services

Any Question?...