

Leveraging the Eclipse Modeling Framework (EMF) to work with Electronic Datasheets (EDS)

Lukas Petersson, ESA/ESTEC,

David Perillo, ESA/ESTEC

ESA-MBSE2021 | September
2021 | Slide 1



Study Context



Implementation guidelines



Validation of Electronic Datasheets



Conclusions and Future Work

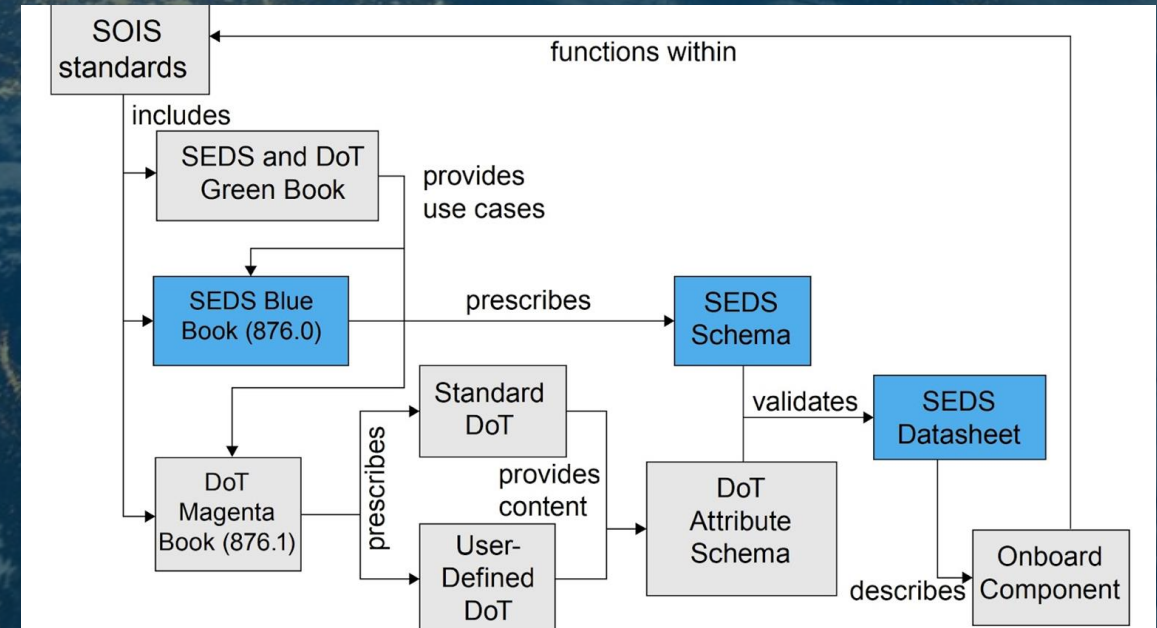
Problems:

- Organize physical documents
- Hard to retrieve relevant information
- Cannot generate artefacts

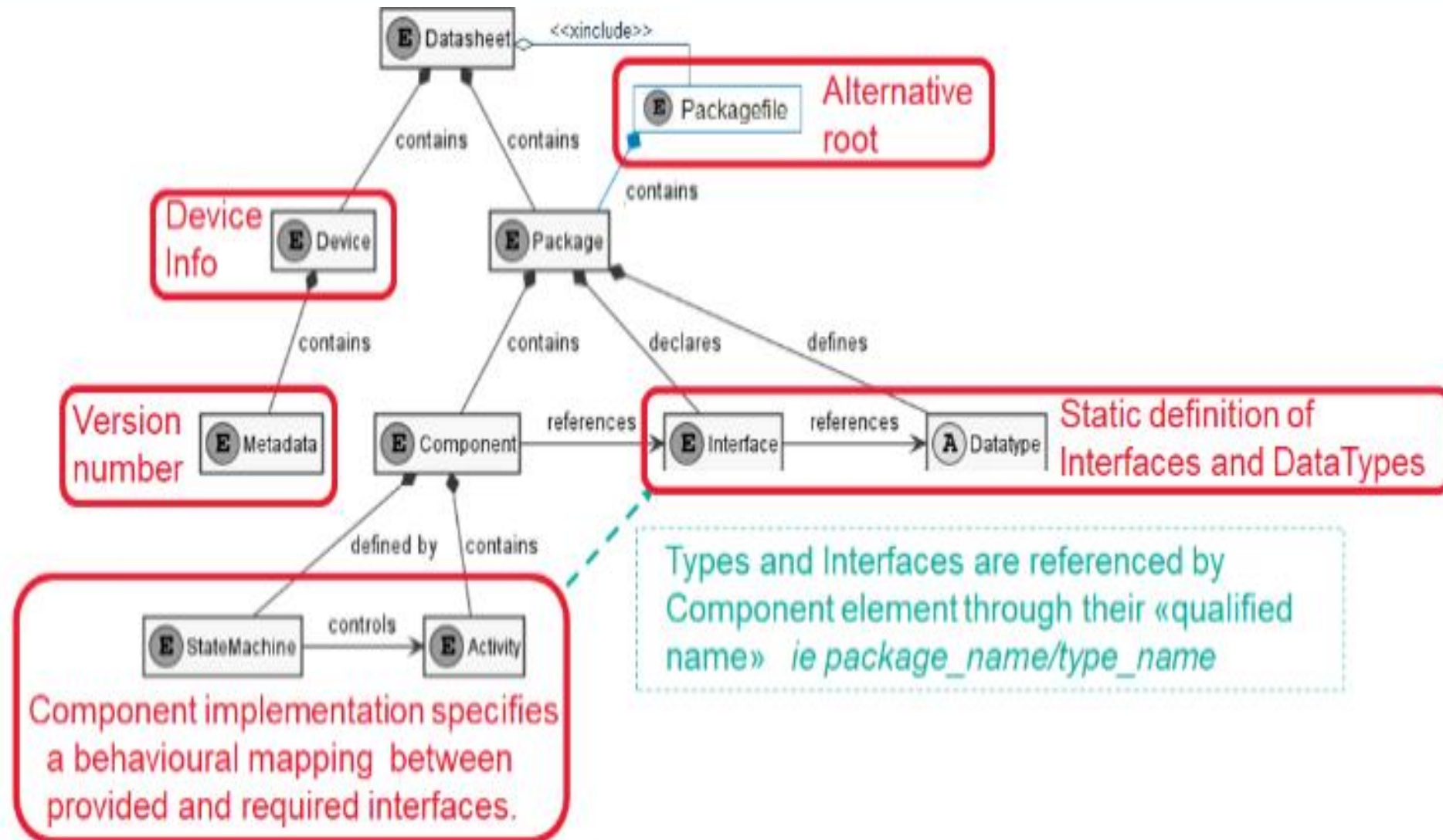


Solution:

Digitalize ICD by means of
SOIS Electronic Datasheets



Structure of a SOIS EDS (xml) file



SOIS EDS schema
(xml schema definition)

compliant to

SOIS EDS model (xml)

```
<StateMachine defaultEntryState="Idle" name="ASYNC" shortDescription="State machine for pr
<State name="Idle"/>
<Transition fromState="Idle" name="RECV_TM_MDB" toState="Idle">
  <OnCommandPrimitive command="receive" interface="subnetworkPS">
    <ArgumentValue name="data" outputVariableRef="working_tm_mdb"/>
  </OnCommandPrimitive>
  <Guard>
    <TypeCondition>
      <FirstOperand variableRef="data"/>
      <TypeOperand>TM_MDB</TypeOperand>
    </TypeCondition>
  </Guard>
  <Do activity="update_TM_MDB"/>
</Transition>
<Transition fromState="Idle" name="RECV_TM_EV_SELFTEST_INPROGRESS" toState="Idle">
  <OnCommandPrimitive command="receive" interface="subnetworkPS">
    <ArgumentValue name="data" outputVariableRef="working_tm_ev_selftest_inprogress"/>
  </OnCommandPrimitive>
  <Guard>
    <TypeCondition>
      <FirstOperand variableRef="data"/>
```

Example EDS file: Jena star tracker

Lines of (xml) code: 7358

Imports 5 EDS Package files (types, definitions, PUS services, ...)

Content:

- ~ 150 data types
- 7 interfaces
- 7 state machines
- 99 activities

Complex to visualise, edit and validate without a proper authoring tool.

EMF generated Tree Editor to Author EDS (xml files)



SOIS EDS schema
(xml schema definition)

consistent with

SOIS EDS ecore metamodel
(generated by SOIS EDS schema)

compliant to

SOIS EDS model (xml)

```
<StateMachine defaultEntryState="Idle" name="ASYNC" shortDescription="State machine for p
<State name="Idle"/>
<Transition fromState="Idle" name="RECV_TM_MDB" toState="Idle">
  <OnCommandPrimitive command="receive" interface="subnetworkPS">
    <ArgumentValue name="data" outputVariableRef="working_tm_mdb"/>
  </OnCommandPrimitive>
  <Guard>
    <TypeCondition>
      <FirstOperand variableRef="data"/>
      <TypeOperand>TM_MDB</TypeOperand>
    </TypeCondition>
  </Guard>
  <Do activity="update_TM_MDB"/>
</Transition>
<Transition fromState="Idle" name="RECV_TM_EV_SELFTEST_INPROGRESS" toState="Idle">
  <OnCommandPrimitive command="receive" interface="subnetworkPS">
    <ArgumentValue name="data" outputVariableRef="working_tm_ev_selftest_inprogress"/>
  </OnCommandPrimitive>
  <Guard>
    <TypeCondition>
      <FirstOperand variableRef="data"/>
```

Output file



compliant to

SOIS EDS Tree Editor

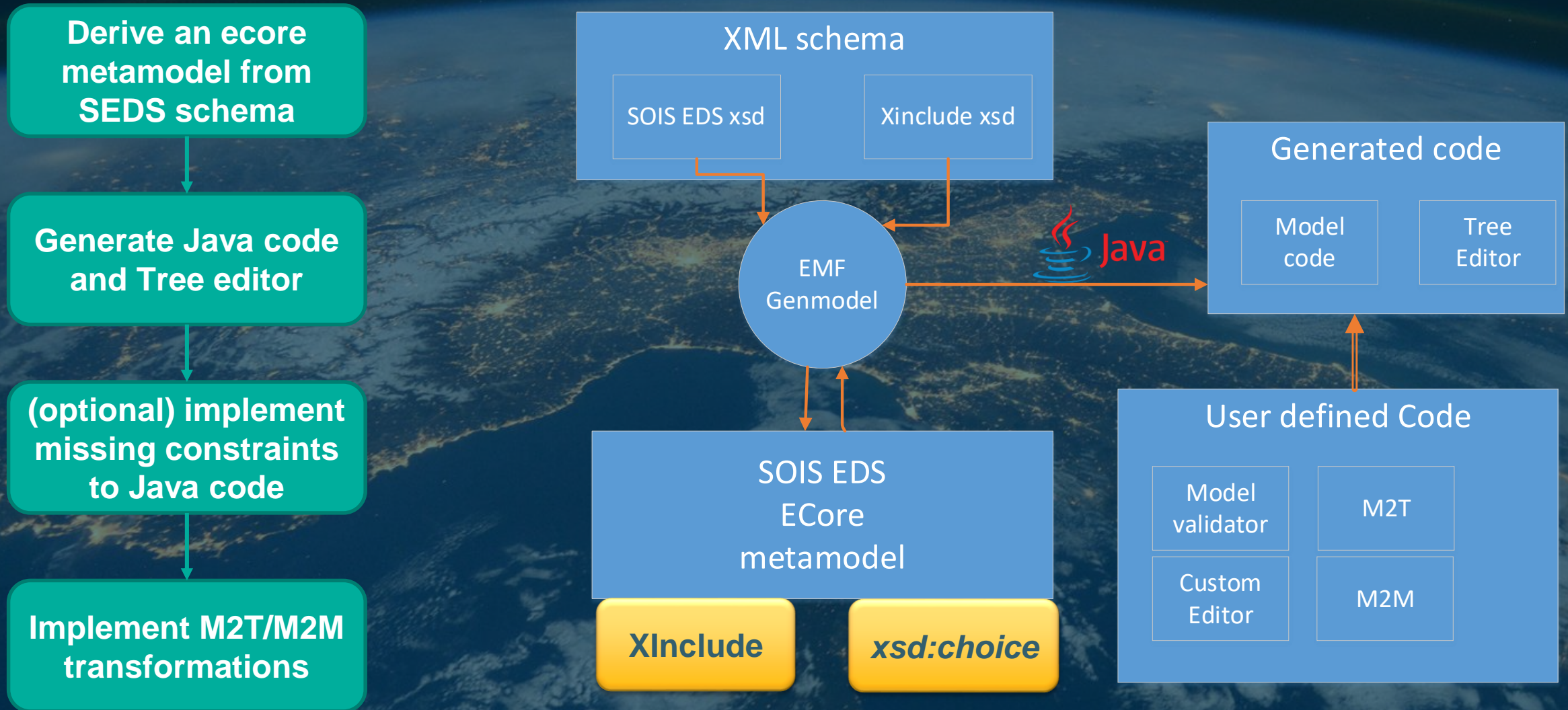
File Edit Seds Editor Window Help

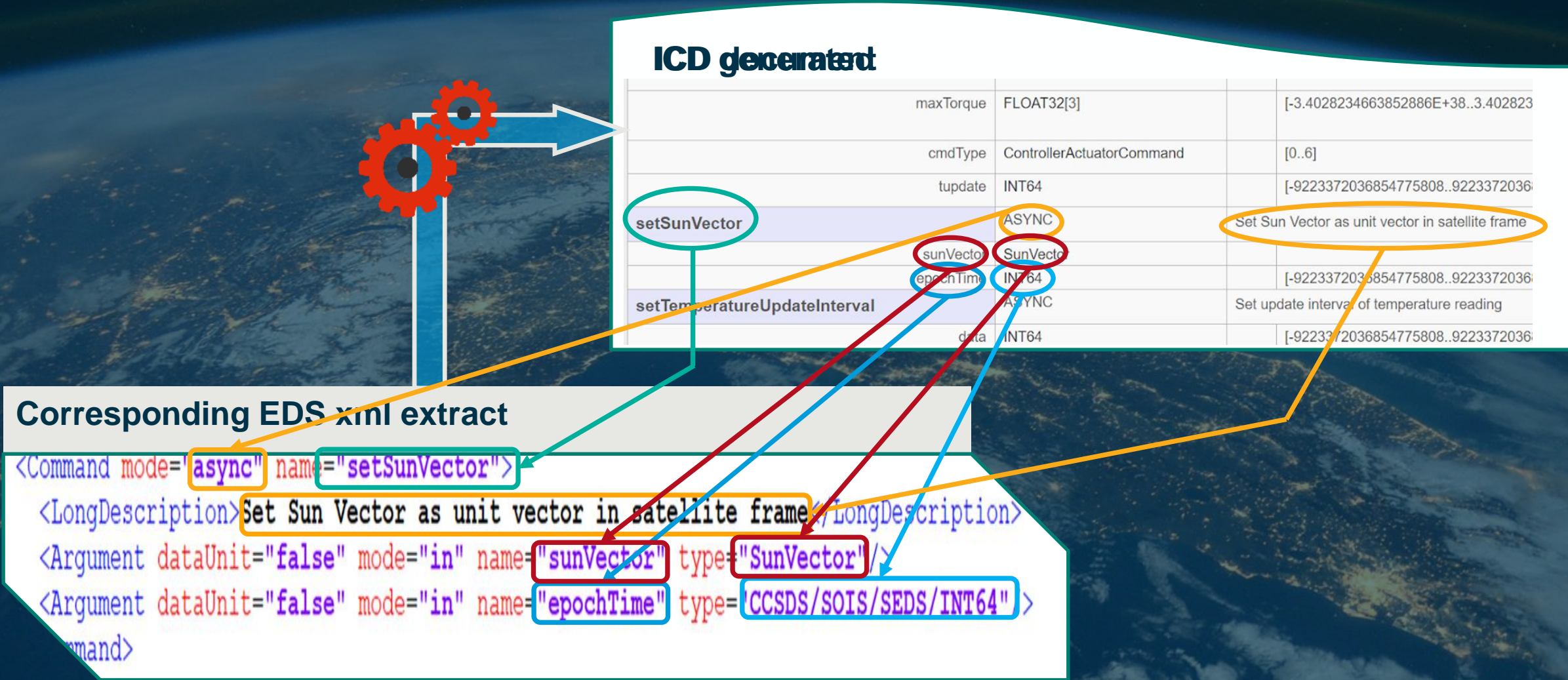
jena_star_tracker.xml

Resource Set

- Interface Set Type
- Interface Set Type
- Component Implementation Type
- Variable Set Type
- Activity Set Type
- State Machine Set Type
 - State Machine Type ASYNC
 - <state> State Type Idle
 - <transition> Transition Type RECV_TM_MDB
 - On Command Primitive Type false
 - Sink Argument Value Type data
 - Boolean Expression Type
 - Type Check Type TM_MDB
 - Variable Ref Operand Type data
 - Activity Invocation Type
 - <transition> Transition Type RECV_TM_EV_SELFTEST_I
 - <transition> Transition Type RECV_TM_EV_OPMODE_!
 - <transition> Transition Type RECV_TM_EV_TIME_SYNC
 - <transition> Transition Type RECV_TM_SIDSTATES DU

Property	Value
Command	receive
Failed	false
Interface	subnetworkPS
Transaction	





Model to Text Transformation example

```
<ContainerDataType
name="TelemetrySet2Type">
  <EntryList>
    <Entry name="hk3" type="HKType"/>
    <Entry name="hk4" type="HKType"/>
  </EntryList>
</ContainerDataType>
```

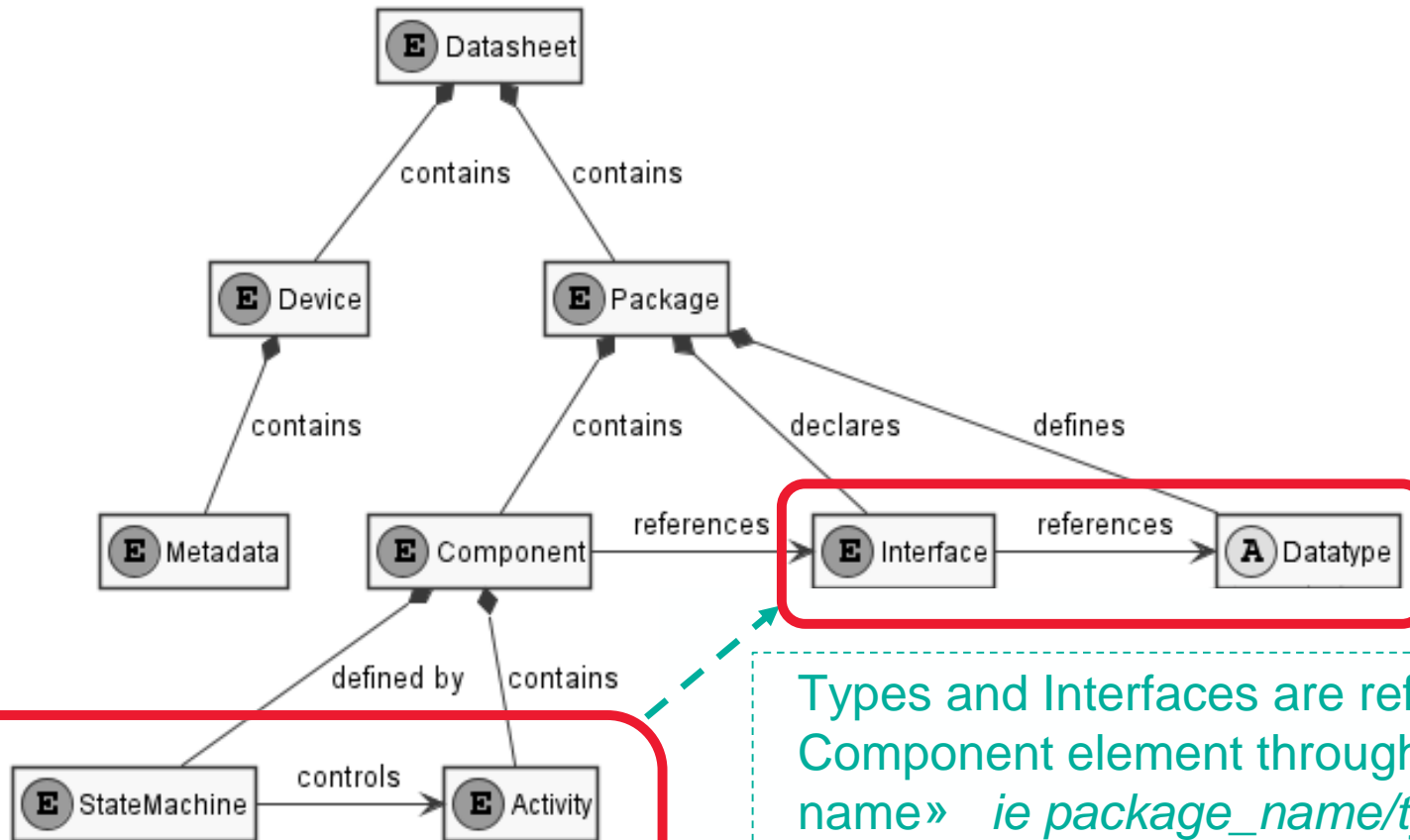
```
<IntegerDataType name="HKType">
  <IntegerDataEncoding encoding="unsigned"
sizeInBits="8"/>
  <Range>
    <MinMaxRange min="0" max="255"
rangeType="inclusiveMinInclusiveMax"/>
  </Range>
</IntegerDataType>
```



Package: Demo

Container Name	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
StatusType	StatusElz	StatusElz														
TelemetrySet1Type				hk1									hk2			
TelemetrySet2Type				hk3									hk4			
Eventtype				type									data			
ExtendedStatusOrModeType				queryCount									type			
ExtendedStatusUnionType				status												
ExtendedModeUnionType				mode												
TelecommandType				type												
TelecommandModeType				mode												
TelecommandUserData	userData															
TelemetryType				type												
TelemetryAckType								status								
Telemetry1Type	telemetr															
Telemetry2Type	telemetr															
TelemetryEventType				eventType									eventData			

The Need for Validation



Component implementation specifies a behavioural mapping between provided and required interfaces.

Types and Interfaces are referenced by Component element through their «qualified name» ie *package_name/type_name*

selected object

qualified name: DEMO_PRES/DAS_COMP/DAS

entity type: Interface

model instance: NA

description

selected object

qualified name: DEMO_PRES/DAS_COMP

entity type: Component

model instance: Demo_Presentation.xml

all genericTypeMapSet

object	name	info
Interface	DAS	(type) "DEMO_PRES/DAS_TYPE"

selected object

qualified name: DEMO_PRES/DAS_TYPE

entity type: InterfaceDeclaration

model instance: NA

all baseInterfaceSet commandSet genericTypeSet parameterSet

object	name
InterfaceDeclaration	DAS_TYPE
ParameterSet	Score
CommandSet	Goal
InterfaceCommand	Team
CommandArgument	

selected object

qualified name: DEMO_PRES/DAS_COMP

entity type: Component

model instance: Demo_Presentation.xml

all dataTypeSet declaredInterfaceSet implementation providedInterfaceSet required

object	name
Component	DAS_COMP
ComponentImplementation	
VariableSet	
ActivitySet	
Activity	send_goal_tc
Activity	forward_score
Body	
Assignment	
Assignment	
SendParameterPrimitive	
Operand	
VariableRefOperand	
SendCommandPrimitive	
NamedArgumentValue	Team
VariableRefOperand	
ValueOperand	
StateMachineSet	

Syntax

3.7.3 A BooleanDataEncoding element shall carry a sizeInBits attribute which specifies the size, in bits, of the encoded data as a positive integer

Semantics

3.15.19 Each ArgumentValue child element of a SendCommandPrimitive element shall carry a name attribute identifying the command argument with which this value is associated.

Property	Type	Value
Long Description	String	
Short Description	String	
Failed	boolean	false
Transaction	TransactionRef	
Command	CommandRef	Goal
Interface	InterfaceRef	DAS

tooltip
No tooltip for SendCommandPrimitiveTypeImpl

SEDS are machine-readable interface specifications, standardized under SOIS
(syntactic definitions of SEDS is available on the CCSDS website)

Discussed a procedure to create SOIS-EDS using the Eclipse Modeling Framework,

CONs: limitations of ecore with regards to very specific XSD constructs

PROs: use EMF for generating a number of spacecraft engineering artefacts

Use Case example using Acceleo M2T

Discussed the need for semantic validation of SEDS.

A detailed identification of semantic errors and how to detect them in SEDS will be subject of a future work.

Thank you for listening!
Questions?