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

Lukas Petersson, ESA/ESTEC,

David Perillo, ESA/ESTEC

Agenda

- Study Context
- Implementation guidelines
- Validation of Electronic Datasheets
- Conclusions and Future Work

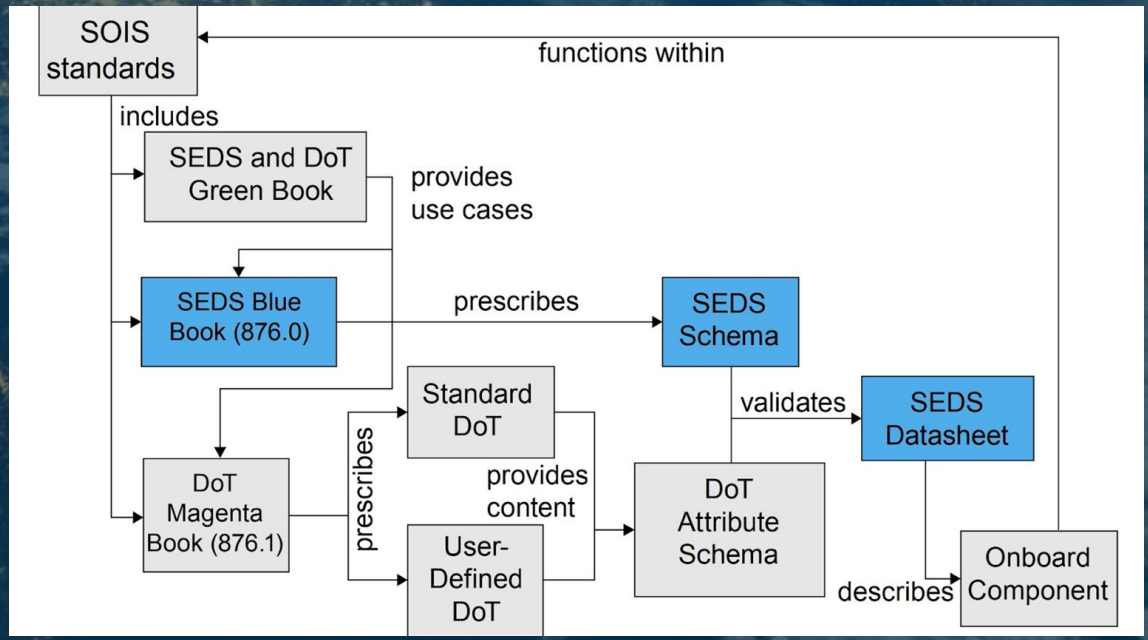
Study Context, working with paper-based ICDs

Problems:

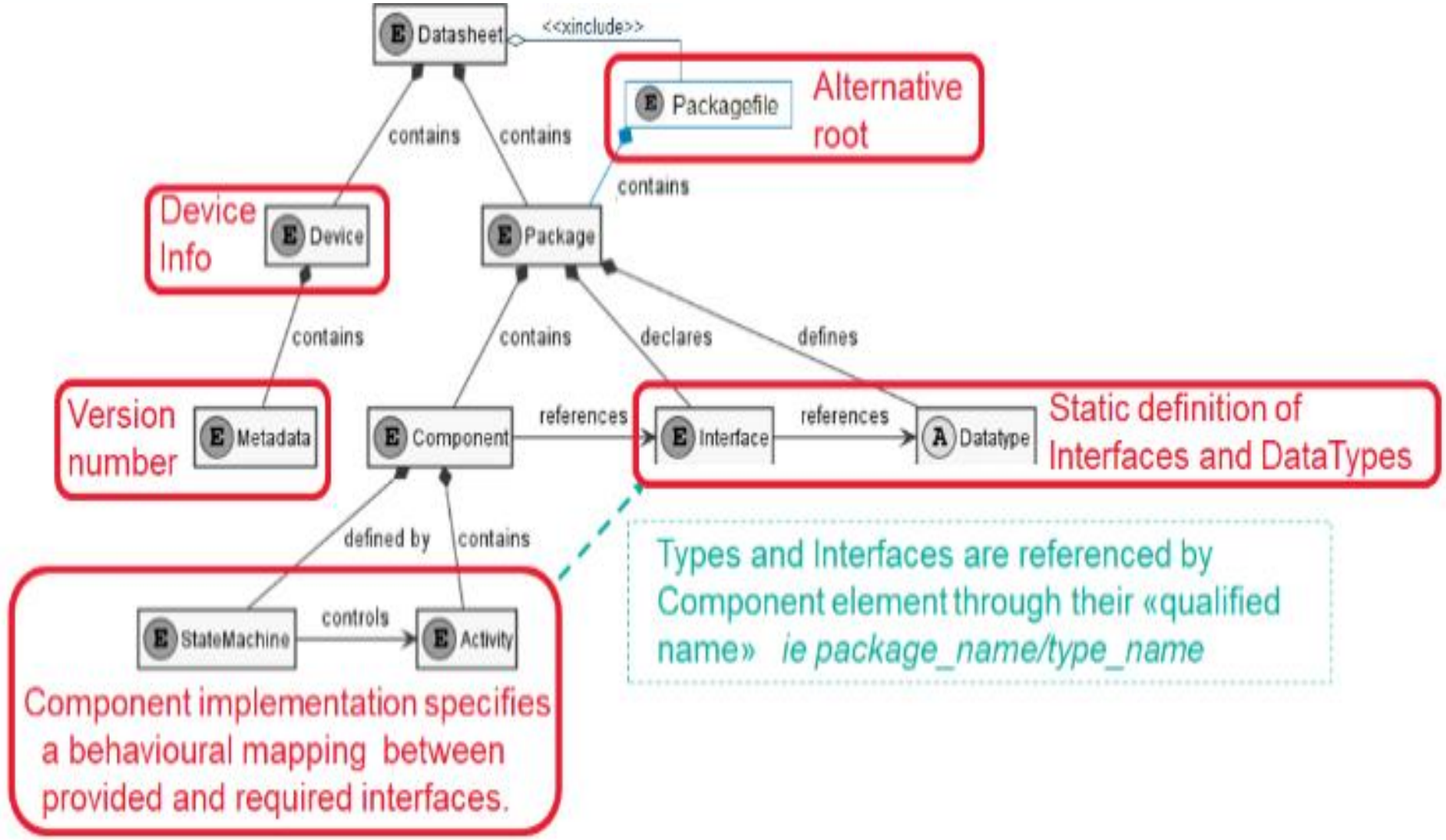
- Organize physical documents
- Hard to retrieve relevant information
- Cannot generate artefacts

Solution:

Digitalize ICD by means of SOIS Electronic Datasheets



Structure of a SOIS EDS (xml) file



EMF generated Tree Editor to Author EDS (xml files)

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)

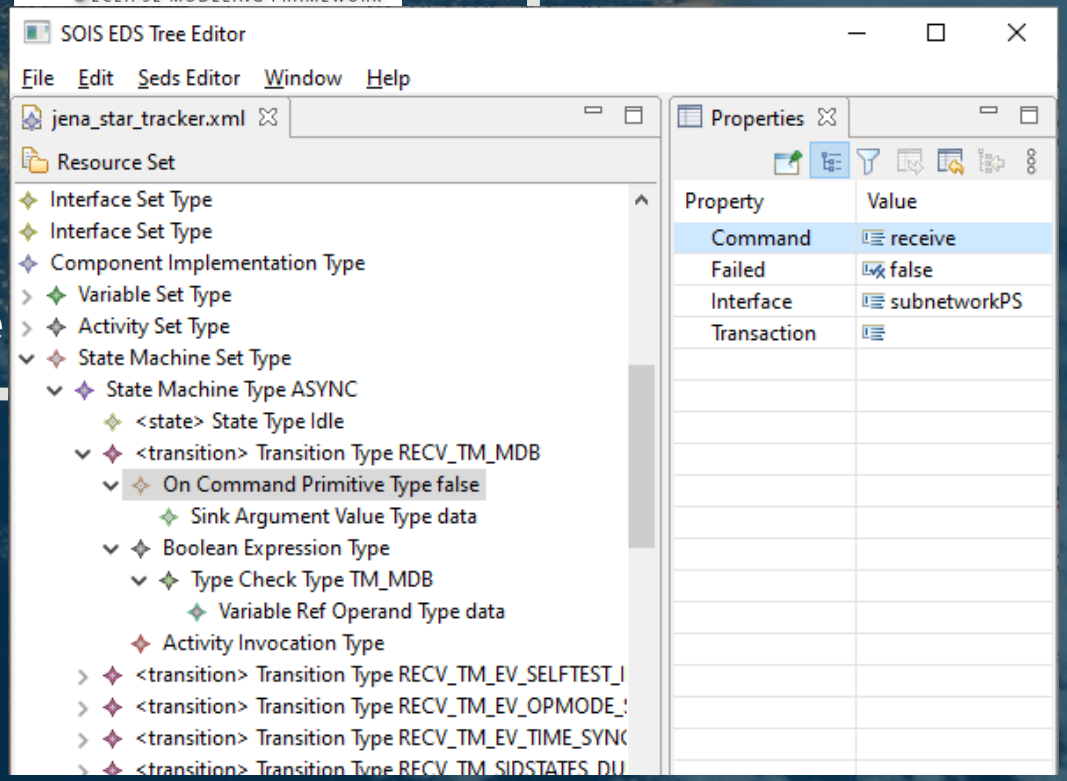
compliant to

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"/>
```

Output file



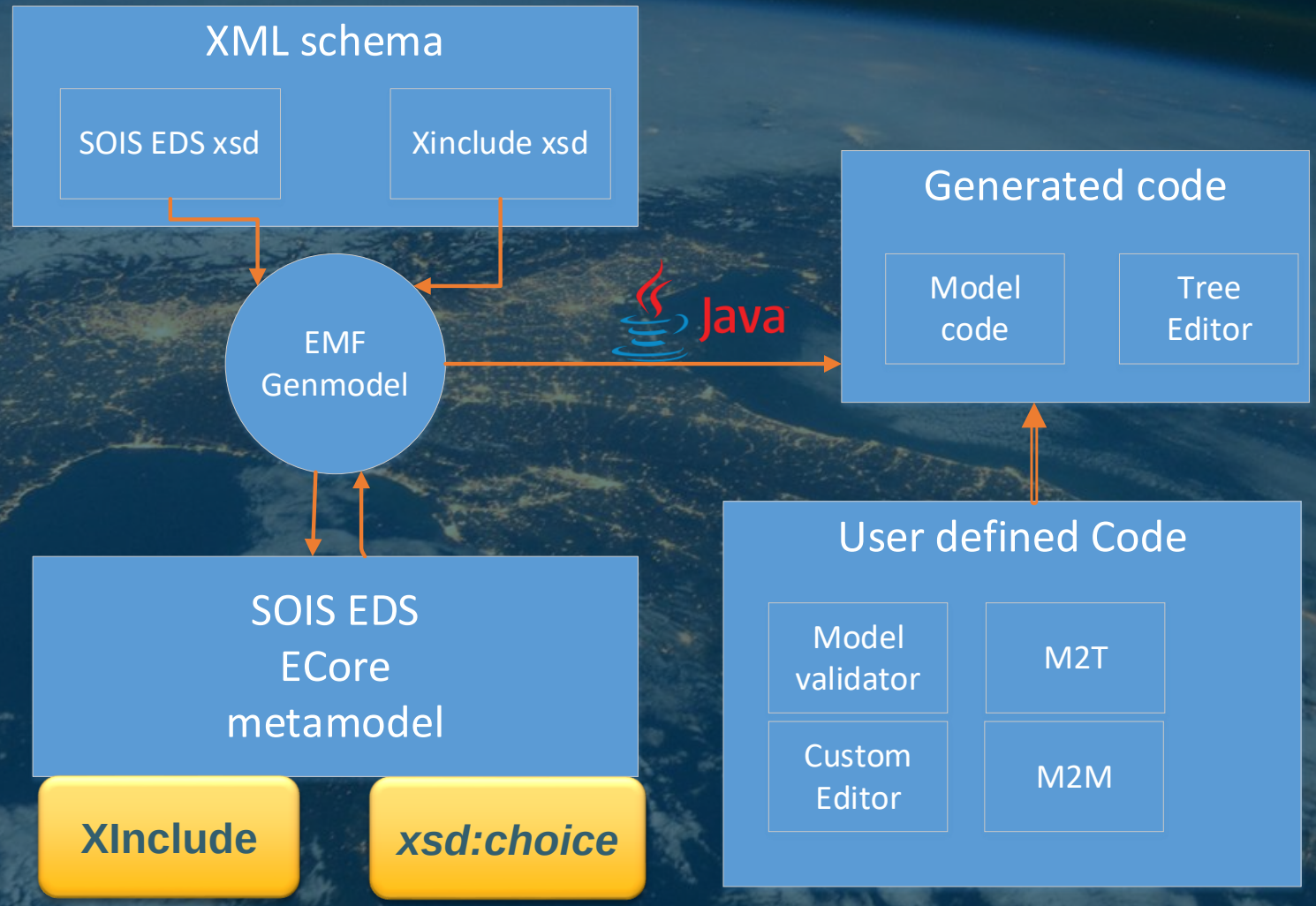
Experimenting EMF

Derive an ecore metamodel from SEDS schema

Generate Java code and Tree editor

(optional) implement missing constraints to Java code

Implement M2T/M2M transformations



Model to Text Transformations

ICD generated

maxTorque	FLOAT32[3]		[-3.4028234663852886E+38..3.402823
cmdType	ControllerActuatorCommand		[0..6]
tupdate	INT64		[-9223372036854775808..9223372036
setSunVector	ASYNC	Set Sun Vector as unit vector in satellite frame	
sunVecto	SunVecto		
epochTim	INT64		[-9223372036854775808..9223372036
setTemperatureUpdateInterval	ASYNC	Set update interval of temperature reading	
data	INT64		[-9223372036854775808..9223372036

Corresponding EDS xml extract

```
<Command mode="async" name="setSunVector">
  <LongDescription>Set Sun Vector as unit vector in satellite frame</LongDescription>
  <Argument dataUnit="false" mode="in" name="sunVector" type="SunVector"/>
  <Argument dataUnit="false" mode="in" name="epochTime" type="CCSDS/SOIS/SEDS/INT64"/>
  mand>
```

Model to Text Transformation example

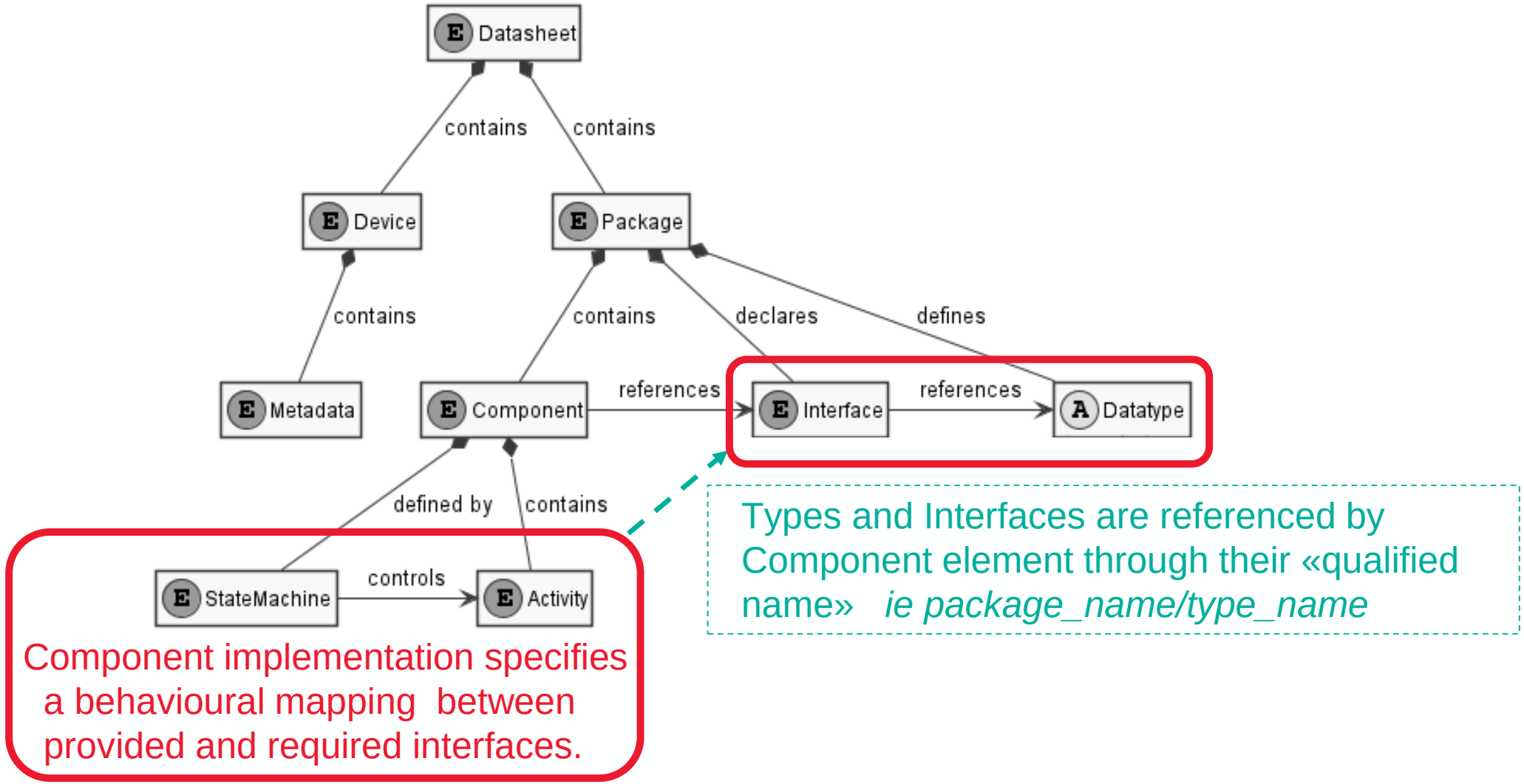
```
<ContainerDataType
name="TelemetrySet2Type">
  <EntryList>
    <Entry name="hk3" type="HKType"/>
    <Entry name="hk4" type="HKType"/>
  </EntryList>
</ContainerDataType>
```

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

Package: Demo

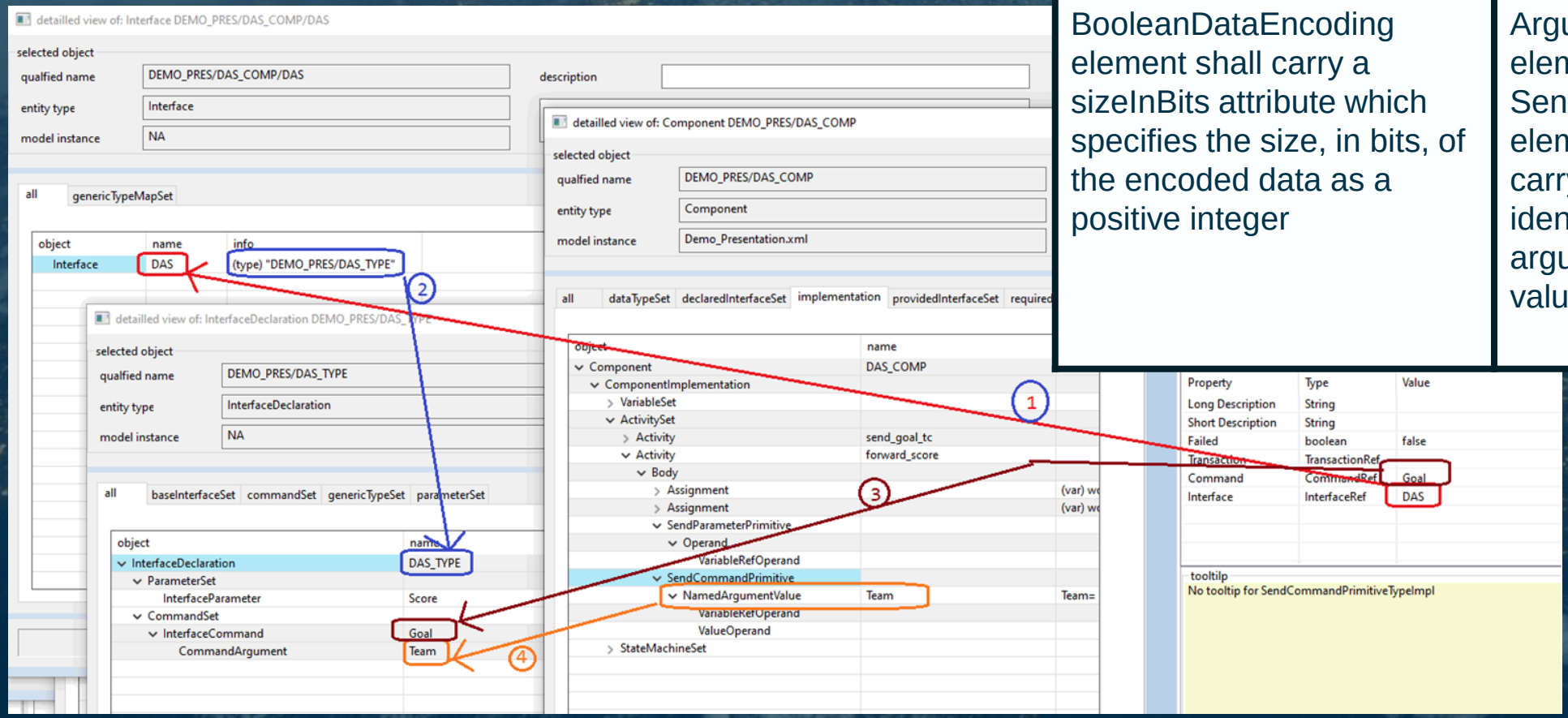
Container Name	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
StatusType	StatusFla	StatusFla				OpCount										
TelemetrySet1Type					hk1								hk2			
TelemetrySet2Type					hk3								hk4			
EventType					type								data			
ExtendedStatusOrModeType					queryCount								type			
ExtendedStatusUnionType					status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

The Need for Validation



Syntax and Semantics

Syntax	Semantics
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	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.



Conclusions and Future work

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?