

Invest for the Future



The contribution is partially funded by the EFRE (European fund for regional development) of the European Union.



www.temea.org

TTCN-3 User Conference 2009

3 – 5 June 2009 - ETSI, Sophia Antipolis, France

■ ■ ■ **Mapping AUTOSAR
Interfaces to TTCN-3**

Jürgen Großmann, Ina Schieferdecker
(Fraunhofer FOKUS)

{juergen.grossmann|ina.schieferdecker}@fokus.fraunhofer.de



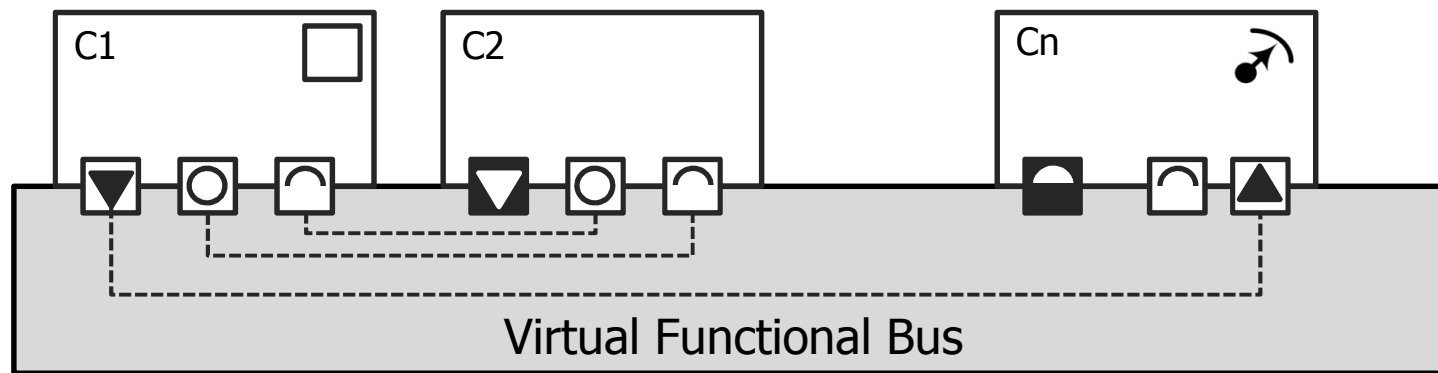
What is AUTOSAR



- AUTOSAR (AUTomotive Open System ARchitecture) is an open and standardized automotive software architecture.
 - component based design
 - standardized OS and middleware for distributed application
 - specification language with XML based exchange format
 - standardized application interfaces
- AUTOSAR supports:
 - automated configuration and deployment of software components
 - exchange and update of software and hardware over the service life of the vehicle
 - ...

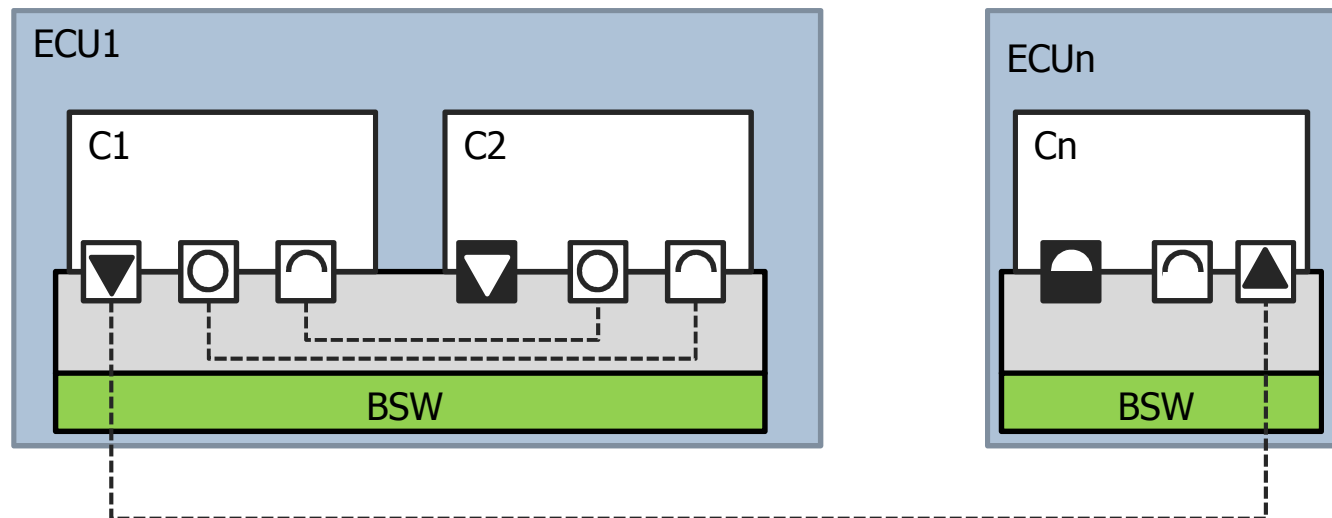
-
-
-

AUTOSAR Core Concepts

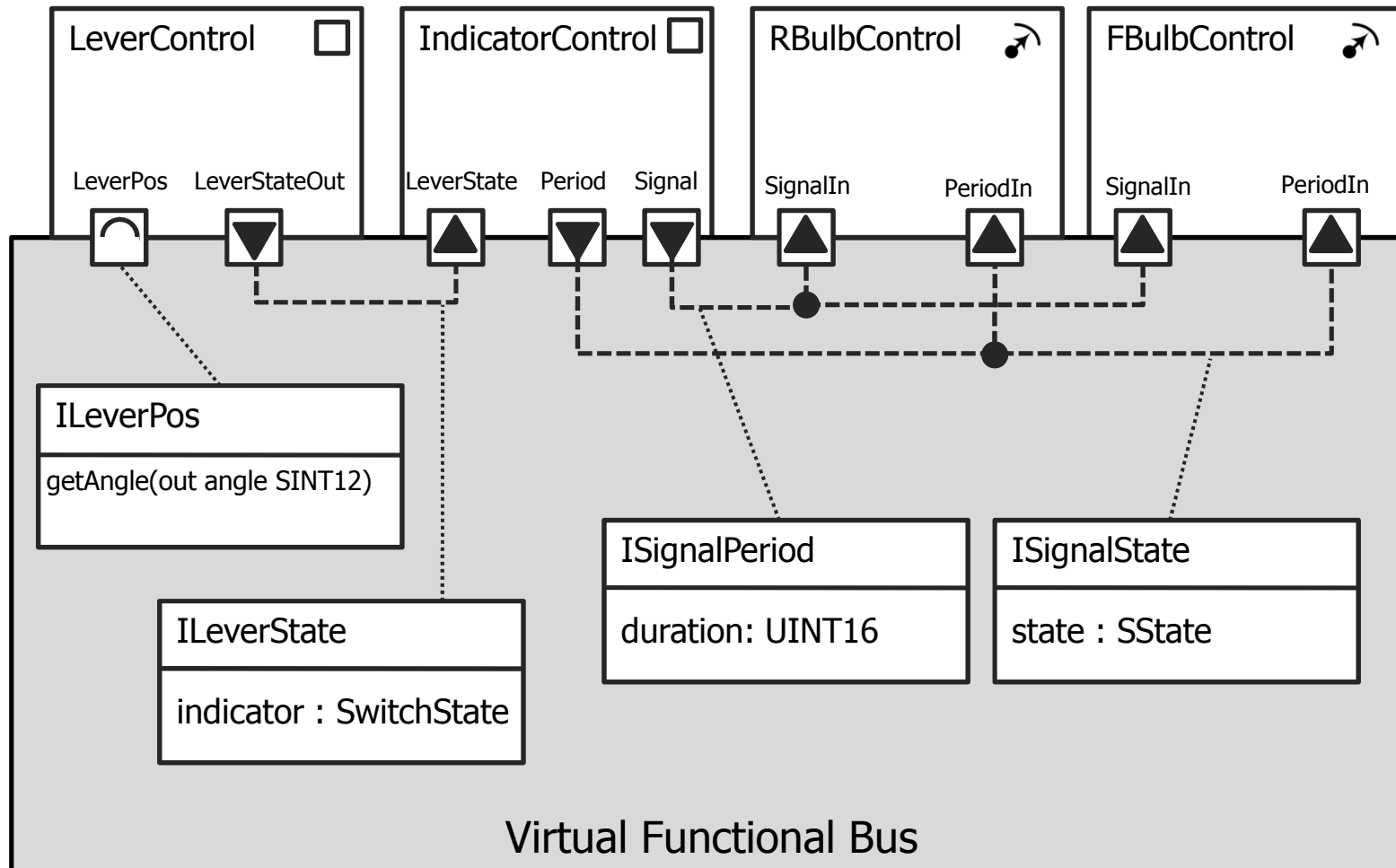


Logical View

Deployment View



Exemplary AUTOSAR Specification



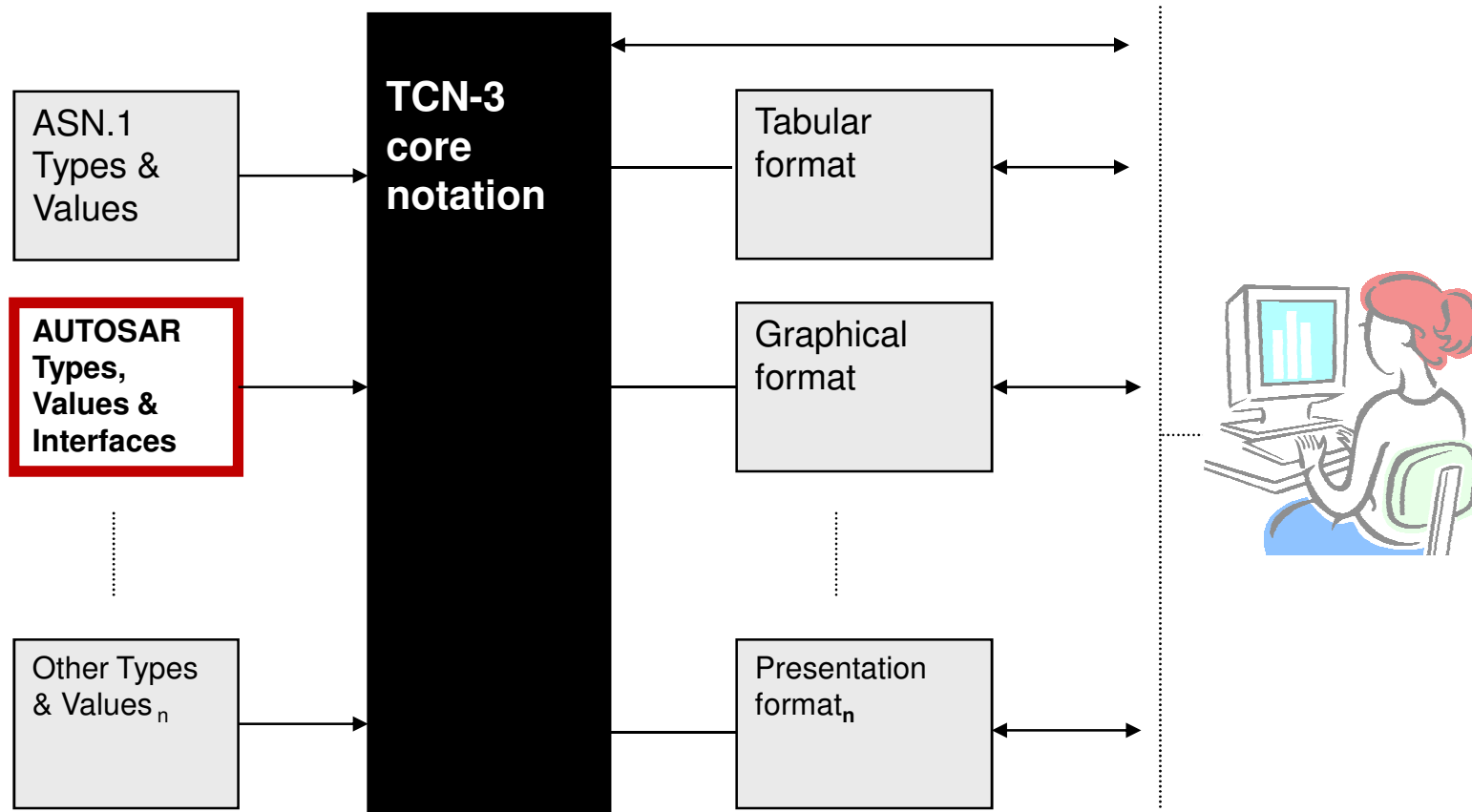
■ Specification Documents and ■ Component Types



- **Software Component Template**
- ECU Resource Template
- System Template

- **Software Components**
 - Application Software Components
 - Sensor Actuator Software Components
 - Service Components
 - Basis Software Components & Complex Driver Components
- **Compositions**
- **Calibration Parameter Components**

AUTOSAR to TTCN-3 Mapping



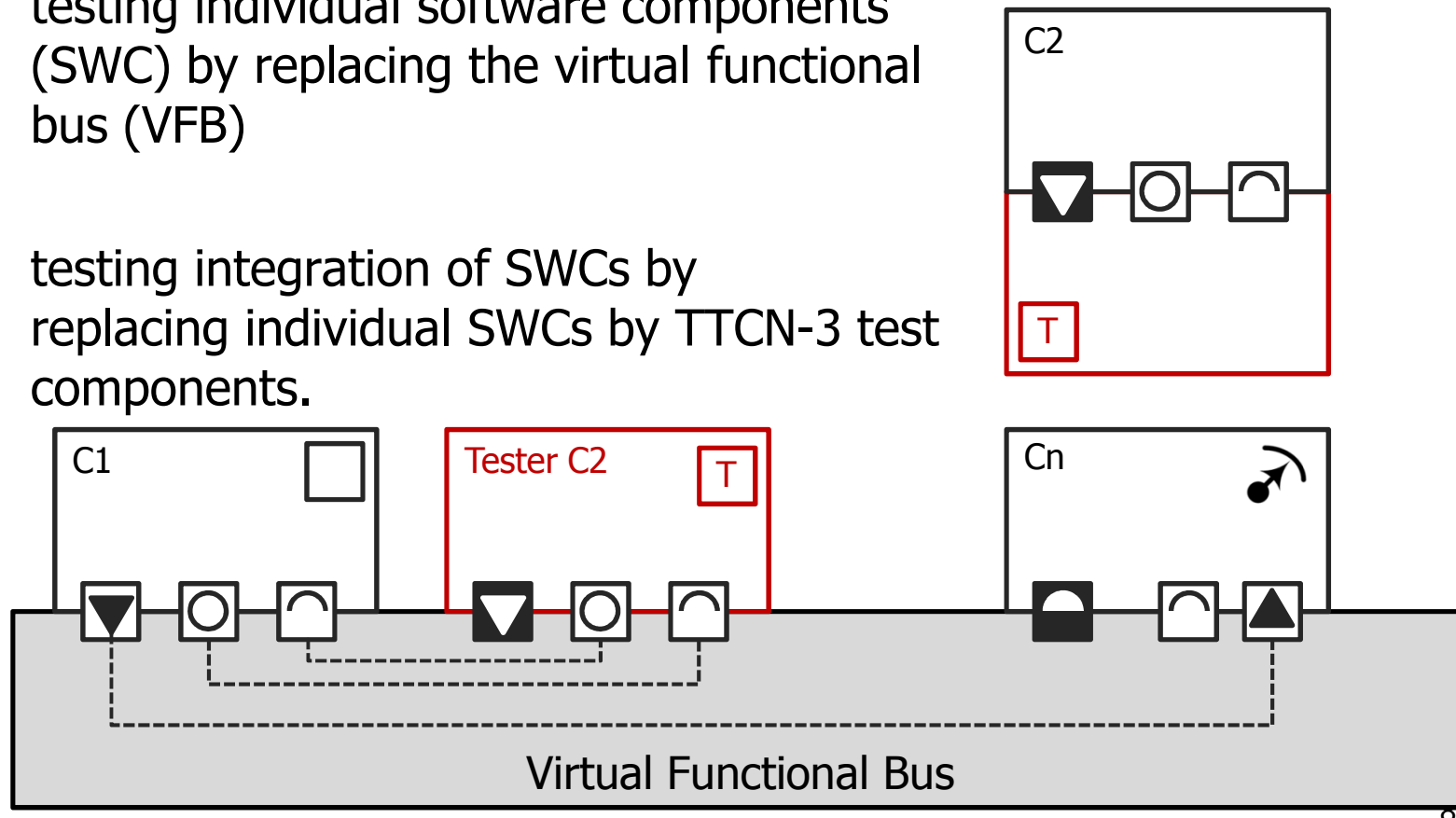
... supports the automation of test development activities for testing AUTOSAR software components.



Objectives



- Support different testing strategies for AUTOSAR SWC
 - testing individual software components (SWC) by replacing the virtual functional bus (VFB)
 - testing integration of SWCs by replacing individual SWCs by TTCN-3 test components.

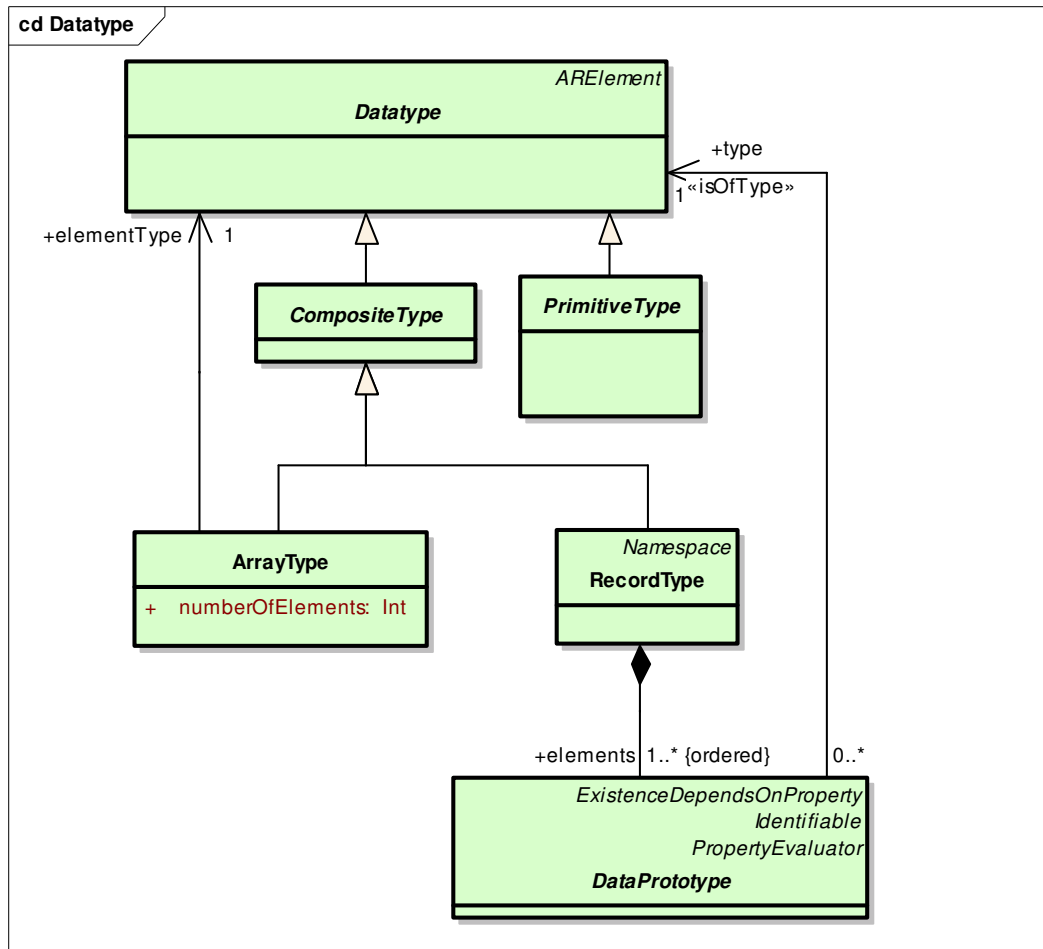


AUTOSAR Concepts to Address



- AR-Components
 - AR-Ports
 - AR-Interfaces
- AR-Communication Characteristic
 - AR-SenderReceiver Communication
 - AR-ClientServer Communication
 - AR-OS Events (periodic and triggered)
- AR-Data Types
 - AR-primitive types
 - AR-complex typed
 - AR-Modes

AUTOSAR Datatypes



- AR-CompositeType
 - ArrayType
 - RecordType
- AR-PrimitiveType
 - OpaqueType
 - CharType
 - BooleanType
 - IntegerType
 - RealType
 - StringType
- AR-Modes



Datatype Mapping (Primitive)



AUTOSAR Type	TTCN-3 Type	AUTOSAR
OpaqueType	bitstring	<pre> <INTEGER-TYPE> <SHORT-NAME>SInt12</SHORT-NAME> <DESC><L-2 L='EN'>12 bit signed integer</L-2></DESC> <LOWER-LIMIT INTERVAL-TYPE='CLOSED'> </pre>
CharType		
BooleanType		
IntegerType		
RealType		
StringType	charstring	<pre> <ENCODING>double</ENCODING> </REAL-TYPE> TTCN-3 type float myFloat (-2048.0..2047.0) with { variant "IEEE754 float double"}; </pre>

AUTOSAR Basistyp	TTCN-3 Definition
uint8	type integer uint8 (0 .. 255)
uint16	type integer uint16 (0 .. 65535)
uint32	type integer uint32 (0 .. 4294967295)
sint8	type integer sint8 (-128 .. 127)
sint16	type integer sint16 (-32768 .. 32767)
sint32	type integer sint32 (-2147483648 .. +2147483647)
float32	type float float32 (-3.4E38 .. 3.4E38)
float64	type float float64 (-1.79769E308 .. 1.79769E308)



Datatype Mapping (Structured)



AUTOSAR Concept	TTCN-3 Concept
Record	record
Array	record of

AUTOSAR
<pre><RECORD-TYPE> <SHORT-NAME>ApplicationError</SHORT-NAME> <DESC><L-2 L='EN'>Describes the ...</L-2></DESC> <ELEMENTS> <RECORD-ELEMENT> <SHORT-NAME>comp</SHORT-NAME> <TYPE-TREF DEST="INTEGER-TYPE"> /AUTOSAR/DataTypes/CompID </TYPE-TREF> </RECORD-ELEMENT> <RECORD-ELEMENT> <SHORT-NAME>error</SHORT-NAME> <TYPE-TREF DEST="INTEGER-TYPE"> /AUTOSAR/DataTypes/ErrorID</TYPE-TREF> </RECORD-ELEMENT> </ELEMENTS> </RECORD-TYPE></pre>
TTCN-3
<pre>type integer CompID (0..31); type integer ErrorID (0..255); type record ApplicationError{ comp CompID, _error ErrorID }</pre>



Mode Communication Mapping

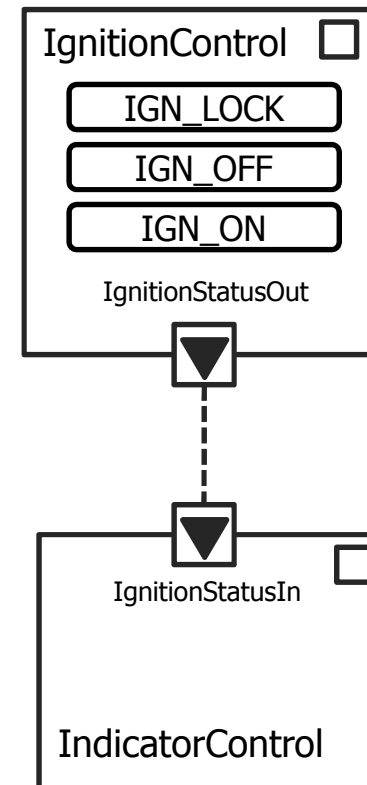


AUTOSAR

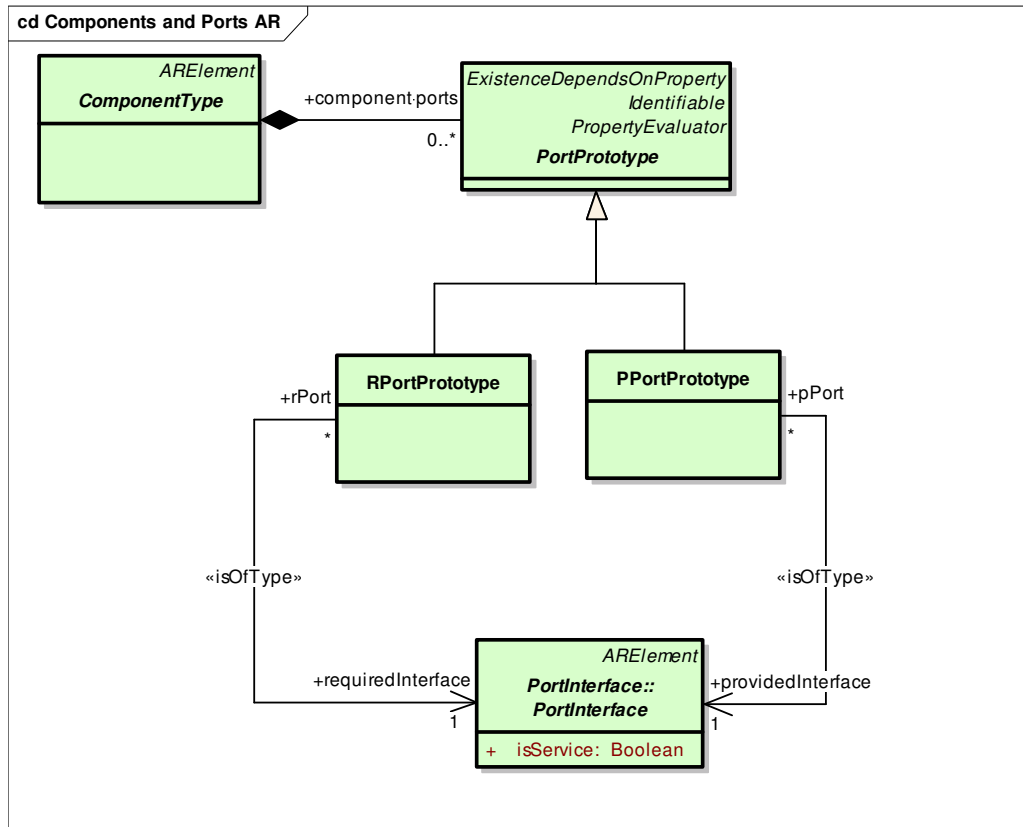
```
<MODE-DECLARATION-GROUP>
  <SHORT-NAME>IgnitionStatus</SHORT-NAME>
  <INITIAL-MODE-REF DEST="MODE-DECLARATION">
    IGN_LOCK</INITIAL-MODE-REF>
  <MODE-DECLARATIONS>
    <MODE-DECLARATION>
      <SHORT-NAME>IGN_LOCK</SHORT-NAME>
    </MODE-DECLARATION>
    <MODE-DECLARATION>
      <SHORT-NAME>IGN_OFF</SHORT-NAME>
    </MODE-DECLARATION>
    <MODE-DECLARATION>
      <SHORT-NAME>IGN_ON</SHORT-NAME>
    </MODE-DECLARATION>
  </MODE-DECLARATIONS>
</MODE-DECLARATION-GROUP>
```

TTCN-3

```
type enumerated IgnitionStatus{
  IGN_LOCK, IGN_OFF, IGN_ON
};
```



Mapping of Components & Interfaces



- ARComp. map to T3Components
- ARPortInterfaces map to T3PortType definitions
- ARPortPrototype map to T3Port definitions



Mapping SR-Interfaces

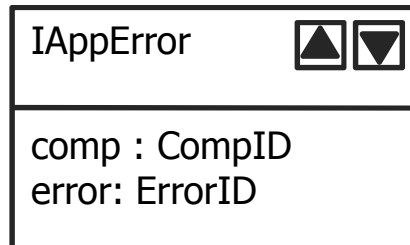
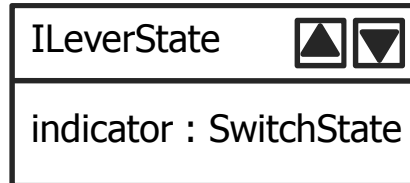
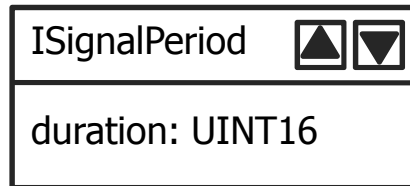


AUTOSAR

```
<SENDER-RECEIVER-INTERFACE>  
<SHORT-NAME>ISignalPeriod</SHORT-NAME>  
<DATA-ELEMENTS>  
  <DATA-ELEMENT-PROTOTYPE>  
    <SHORT-NAME>duration</SHORT-NAME>  
    <TYPE-TREF DEST='INTEGER-TYPE'>  
      /AUTOSAR/DataTypes/UINT16</TYPE-TREF>  
    <IS-QUEUED>>false</IS-QUEUED>  
  </DATA-ELEMENT-PROTOTYPE>  
</DATA-ELEMENTS>  
</SENDER-RECEIVER-INTERFACE>
```

TTCN-3

```
//alternative 3  
type port ISignalPeriod message {  
  inout UINT16  
}
```





Mapping CS-Interfaces



AUTOSAR

```
<CLIENT-SERVER-INTERFACE>
<SHORT-NAME>ILeverPos</SHORT-NAME>
<IS-SERVICE>>false</IS-SERVICE>
<OPERATIONS>
  <OPERATION-PROTOTYPE>
    <SHORT-NAME>getAngle</SHORT-NAME>
    <ARGUMENTS>
      <ARGUMENT-PROTOTYPE>
        <SHORT-NAME>angle</SHORT-NAME>
        <TYPE-TREF DEST='INTEGER-TYPE'>
          /DataTypes/SINT12</TYPE-TREF>
        <DIRECTION>OUT</DIRECTION>
      </ARGUMENT-PROTOTYPE>
    </ARGUMENTS>
  </OPERATION-PROTOTYPE>
</CLIENT-SERVER-INTERFACE>
```

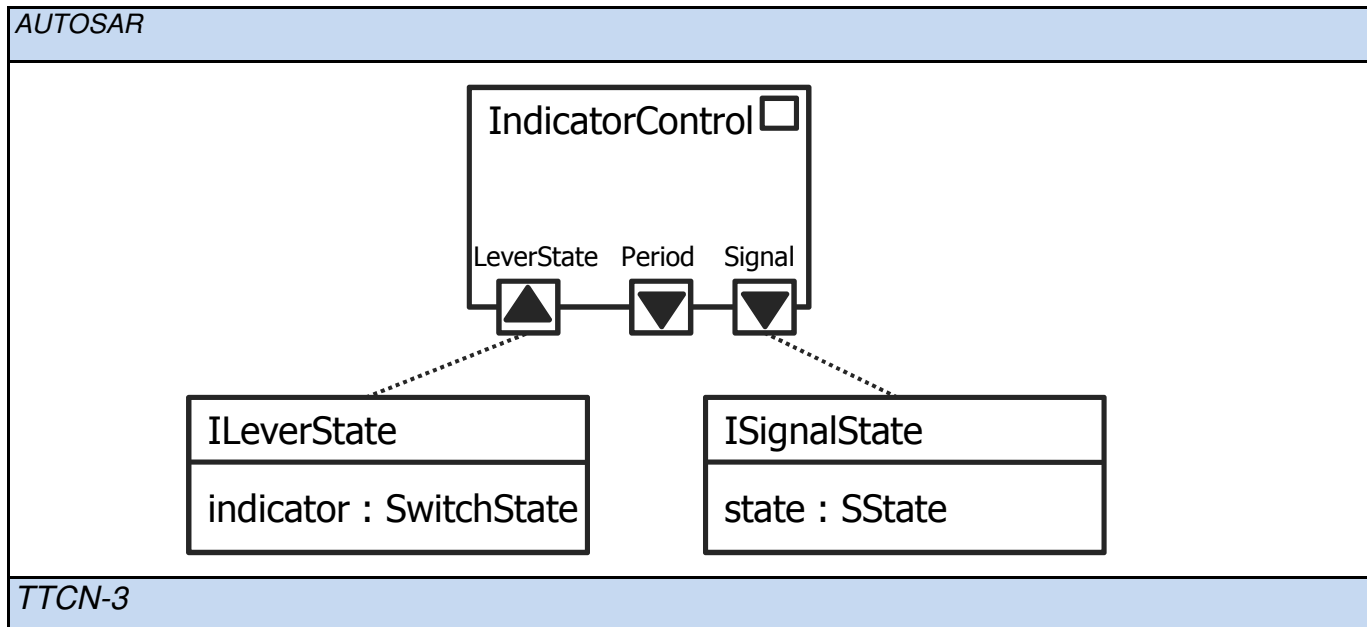
TTCN-3

```
type integer SINT12 (-2048..2047);
signature getAngle(out angle SINT12);
type port ILverPos procedure {
  inout getAngle
}
```

ILeverPos	
getAngle (out angle SINT12)	

ILeverAdv	
getAngle (out angle SINT12) getState (out state SINT4)	

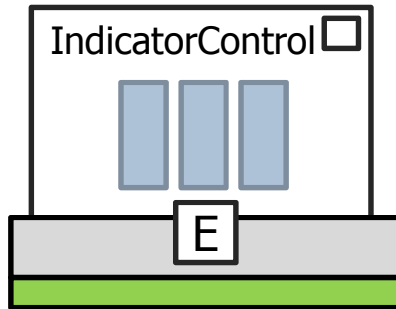
Mapping of Ports and Components



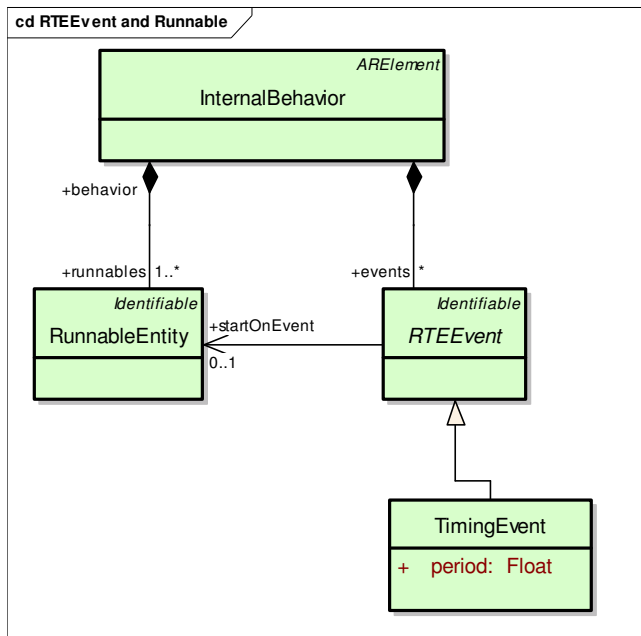
TTCN-3

```
// strategy2: TC replaces OS
type component IndicatorControl{
  port ILeverStateOut LeverState;
  port ISignalStateIn Signal;
  port ISignalPeriodIn Period;
}
```

Events & Runnables



- Runnables: smallest executable entity
- Triggered by Events
- RT event semantics realized by the test adapter



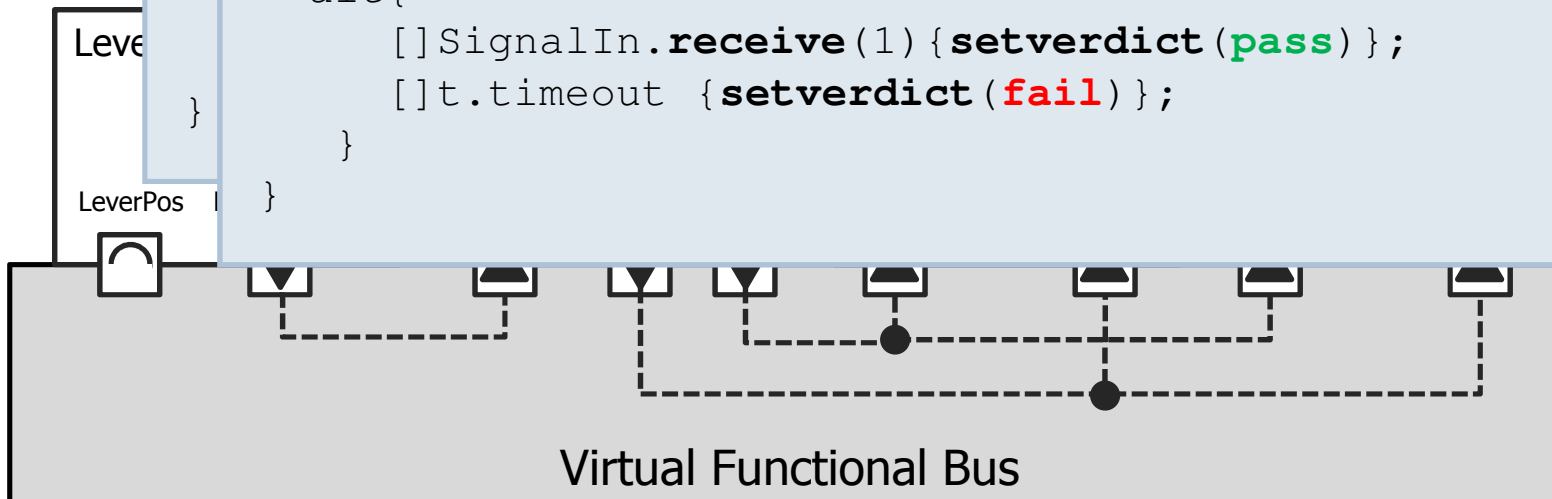
```

type record Event{
  name charstring,
  period float
}

//RT semantics given by adapter
type port EventPort message{
  out Event
}
  
```

Example

```
type integer SState (0..2);
type integer UINT16 (0..65535);
type integer SState (0..2);
}
type
testcase activation runs on FBulbControl{
timer t;
}
var LeverControlTC leverC;
leverC:= LeverControlTC.create
leverC.start(indicatorLeft())
t.start(2.0);
alt{
[]SignalIn.receive(1){setverdict(pass)};
[]t.timeout {setverdict(fail)};
}
}
```





Summary & Outlook



- Basic AUTOSAR component concepts are mapped to TTCN-3.
- TTCN-3 Concepts largely fit.
- Mapping depends on test strategy.

- Mapping to be accompanied with coding and naming conventions.
- For special kind of communication dedicated TTCN-3 concepts is desirable (e.g. periodic events).
- AUTOSAR to TTCN-3 code generator will be available this summer



Contact and Info



www.temea.org

Contact

Jürgen Großmann
Fraunhofer FOKUS

Kaiserin-Augusta-Allee 31, 10589 Berlin

Tel: +49-30-3463-7390

juergen.grossmann@fokus.fraunhofer.de