AUTOSAR Configuration Overview

KPIT

Table of Contents

1. System Input
2. Software Component Input (from MATLAB)
3. Additional configuration to MATLAB output
4. Mapping System & Software Elements

1. System Input

a. System Template

- As System Constraint Description (e.g. CAN DBC, FIBEX, LDF, etc.) it serves as input to the AUTOSAR system generator

- As System Configuration Description it defines the output of the AUTOSAR System Configuration Generator and serves as input to the AUTOSAR ECU Configuration Generator for the different ECUs defined in the description

- As ECU Extract of the System Configuration Description it describes the ECU specific view on the System Description. It is individually generated for each of the System's ECU as the output of the AUTOSAR ECU Configuration Generator

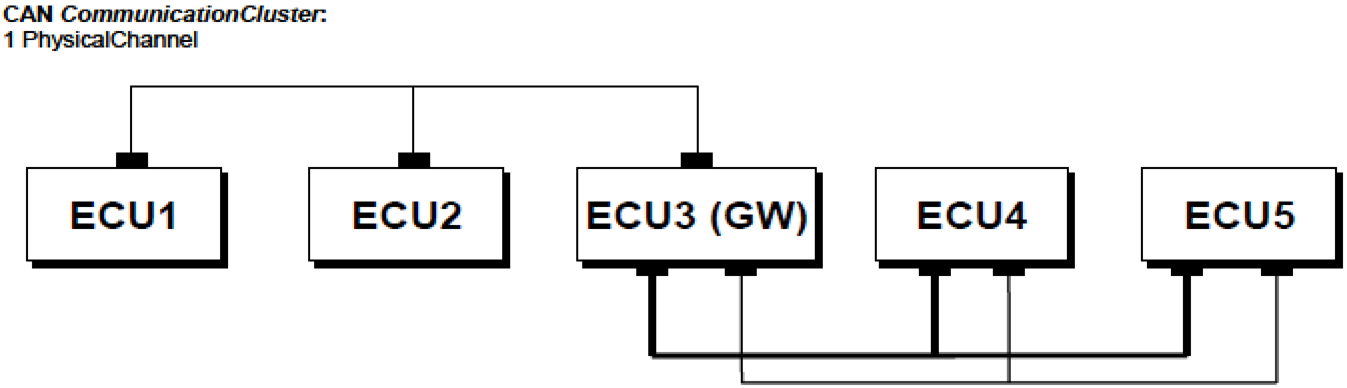
- The System Template defines five major elements: Topology, Software, Communication, Mapping and Gateway

1. System Input

b. Topology

- A topology is formed by a number of **ECUInstances** that are interconnected to each other in order to form ensembles of ECUs and **CommunicationClusters**
- **ECUInstance** describes the presence of a microcontroller in the vehicle
- An **ECUInstance** needs to have one or more **CommunicationController**, the actual hardware device by means of which devices send and receive frames from the communication medium
- **ECUInstance** has one or more **CommunicationConnectors** which describe the bus interfaces of the ECUs and to specify the sending/receiving behavior
- An **ECUInstance** can serve as a gateway if it is connected to two or more different clusters by two or more of its **CommunicationControllers**
- A **CommunicationController** is a dedicated hardware device by means of which hosts are sending frames to and receiving frames from the communication medium
- **CommunicationConnector** connects the **ECUInstance** it is associated with to the **PhysicalChannel** using the **CommunicationController** it references

Note:
Topology can be ***bus, star, ring, tree***



Example for a Communication Cluster within a physical network topology

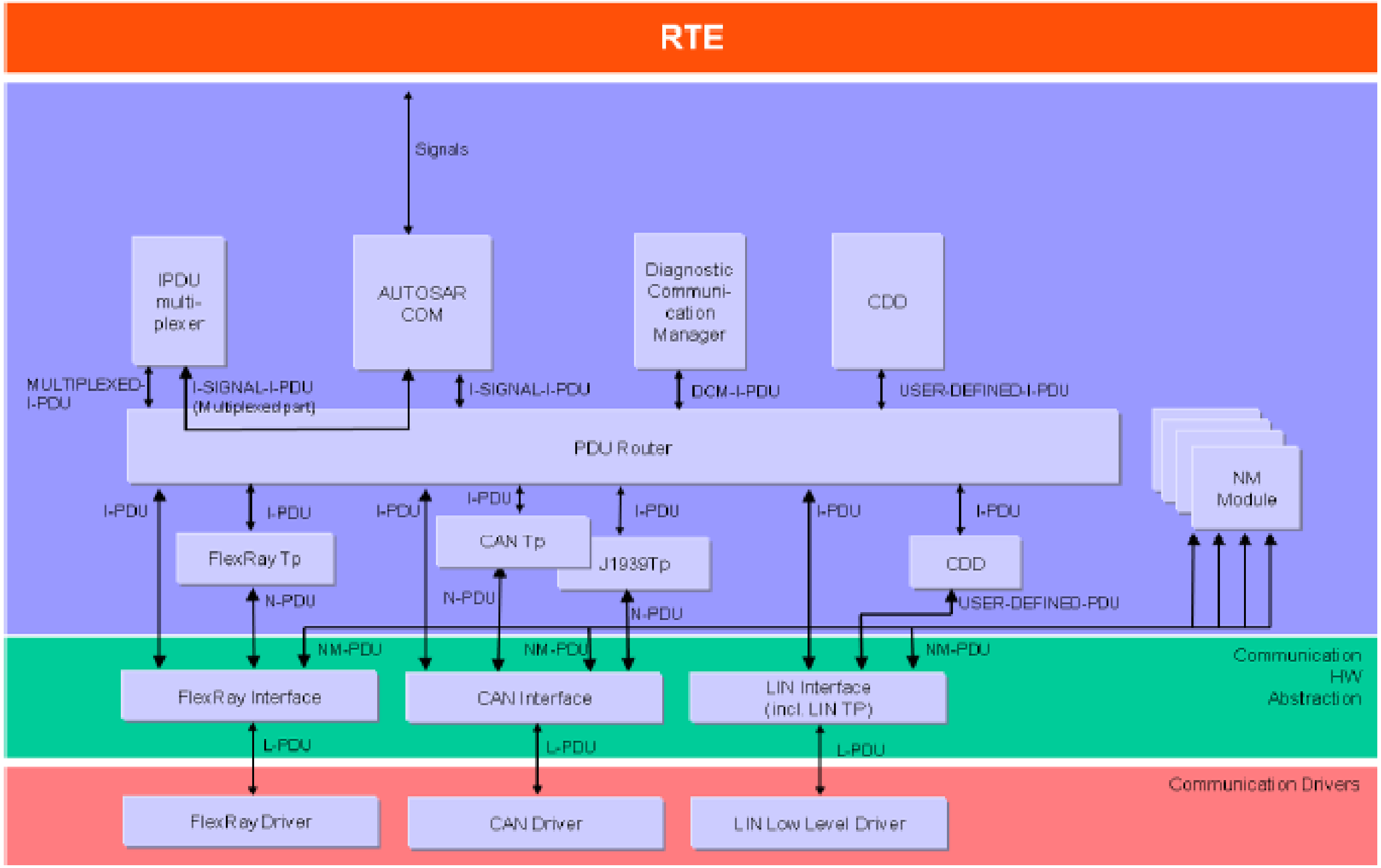
1. System Input

c. Communication

- The main elements to describe communication in the System Template are the **Signals** (System Signals and ISignals), **PDUs** (I-Pdus, N-Pdus and NmPdus) and **Frames**
- A **Frame** is a piece of information that is exchanged over the communication channels
- A **PDU (Protocol Data Unit)** is the information delivered through a network layer
  - **I-PDU** - Interaction Layer Protocol Data Unit (assembled and disassembled in COM)
  - **N-PDU** - Network Layer Protocol Data Unit (assembled and disassembled in a Transport Protocol module)
  - **L-PDU** - Data Link Layer Protocol Data Unit (assembled and disassembled in AUTOSAR Hardware Abstraction layer)
- The elements ***FrameTriggering***, ***PduTriggering*** and ***SignalTriggering*** are describing the usage of ***Frames***, ***IPdus*** and ***Signals*** on a physical channel
- In the System Template the COM controlled timing is described with the aggregation between the ***SignalIPdu*** and the ***IPduTiming***
- The LIN and FlexRay Scheduling Tables are described in the ***FrameTriggering***
- Timing requirements for FlexRay, TTCAN and LIN Pdus can be specified with the Timing Extension model
- In case a NmPdu contains user data and is handled by the BusNm via the PduR and Com the NmPdu gets PduTriggering and IPduPort

1. System Input

d. Communication elements in different software layers



1. System Input

e. Gateways

- A **gateway** is a function within an ECU that performs as a Frame, I-Pdu or signal mapping function between two or more communication clusters
- It contains the following mapping functions:
  - ❑ Frame Mapping
  - ❑ I-Pdu Mapping
  - ❑ Signal Mapping
- The ***FrameMapping*** arranges those frames that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them. Each pair consists in a Source and a Target referencing to a FrameTriggering
- The ***IPduMapping*** arranges those I-Pdus that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them. Each pair consist of a source and a target referencing to a PduTriggering
- The ***ISignalMapping*** arranges those signals and signal groups that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them. Each pair consists of a source and a target referencing to an ISignal-Triggering. Each ISignalTriggering points to either an ISignal or an ISignalGroup which are part of an ISignalIPdu. The ISignal refers to the to be routed SystemSignal, the ISignalGroup refers to the to be routed SystemSignalGroup.

2. Software Component Input (from MATLAB)

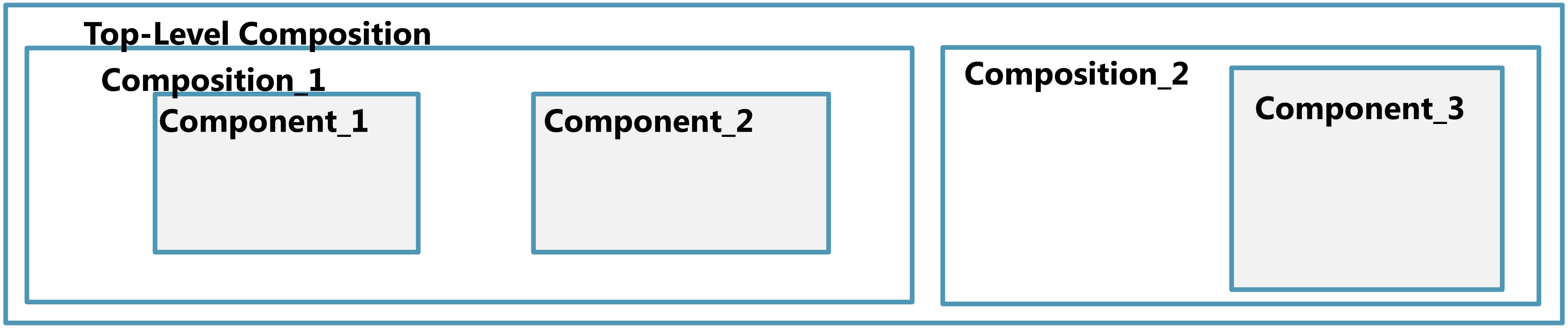
- ✓ Software Component Description file is an arxml file generated from MATLAB after modeling the application as per AUTOSAR requirements

- ✓ It consists of the following information:
  - ➢ Data Types
    - o SwBaseType
    - o ImplementationDataType
    - o ApplicationDataType
  - ➢ Interfaces
    - o SenderReceiverInterface
    - o ClientServerInterface
    - o ModeSwitchInterface
  - ➢ Application Software Components
    - o PPortPrototype
    - o RPortPrototype
    - o SwInternalBehavior

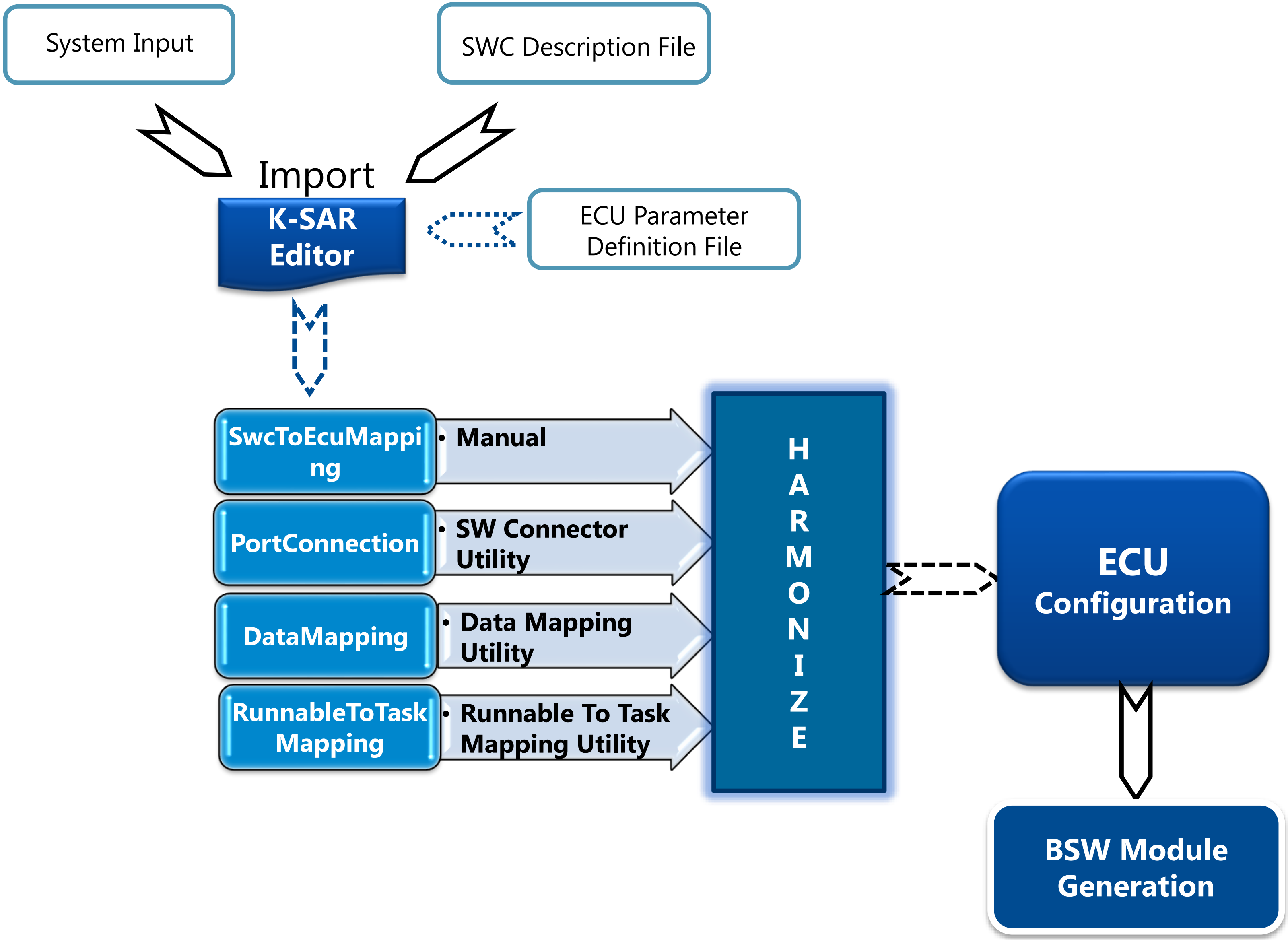
3. Additional Configuration to MATLAB output

Top-level Software Composition

- One of the most important inputs for the System Generator is the knowledge about the Application Software Components, their communications capabilities and the connections between them
- Each SystemSignal that is going to be exchanged between mapped Software Components onto different ECUs is a consequence of a connection between such application Software Components
- In AUTOSAR, Software Components can either be atomic (**AtomicSwComponentType**) or may consist of a composition of other Software Components **CompositionSwComponentType**
- System should have an outermost **CompositionSwComponentType** which forms a kind of ***Top-Level Composition***
- The ***Top-Level Composition*** connects to the SWC by their **ComponentTypes**, **PortPrototypes**, **PortInterfaces**, **DataElementPrototypes**, **InternalBehavior** etc
- A System considers such a **top-level CompositionSwComponentType** as its application software system input by owning exactly one **RootSwCompositionPrototype** class, which points to the **CompositionSwComponentType**



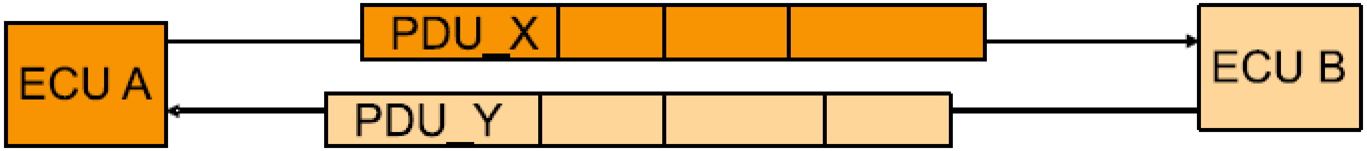
4. Mapping System & Software Elements - Workflow



4. Mapping System & Software Elements - Details

Mapping

- A central part of the system generation process is the **mapping of software components (SwComponentPrototypes) to ECUs**, and the subsequent **mapping of the communication between these software components to bus frames**
- The following mappings are defined:
  - ❑ The ***SwCompToEcuMapping*** meta-class maps one or several SwComponent-Prototypes to ECUs
  - ❑ The ***SwCompToImplMapping*** meta-class is used to assign one Implementation to one or more SwComponentPrototypes
  - ❑ The ***MappingConstraint*** meta-class is used to define constraints that constrain the mapping of software components
  - ❑ The ***DataMapping*** meta-class is used to map VariableDataPrototypes and ClientServerOperations in software component ports (i.e. the data exchanges between software components) to signals. Examples are:
    - ✓ SenderReceiverToSignalMapping
    - ✓ SenderReceiverToSignalGroupMapping
    - ✓ ClientServerToSignalGroupMapping
  - ❑ The ***ECUResourceMapping*** meta-class is used to map the hardware related topology elements onto their counterpart definitions in the ECU Resource Template
  - ❑ Finally, meta-class ***EcuResourceEstimation*** specifies the resource estimation for RTE and BSW



Example topology with two ECUs and two PDUs exchanged between them

AUTOSAR Configuration Overview

Reference Documents

Sl. No.	Document name	AUTOSAR Release	Document Version
1	Specification of ECU Configuration http://www.autosar.org/download/R4.0/AUTOSAR_TPS_ECUConfiguration.pdf	4.0.3	3.2.0
2	System Template http://www.autosar.org/download/R4.0/AUTOSAR_TPS_SystemTemplate.pdf	4.0.3	4.2.0
3	Software Component Template http://www.autosar.org/download/R4.0/AUTOSAR_TPS_SoftwareComponentTemplate.pdf	4.0.3	4.2.0

Thank you

www.kpit.com