Automotive & Embedded Info

Never Forget Basics Whether its Life or Anything Else ... Basics are Cores. While seeing a Tree how can we forget Seed...

RunTime Enviroment

RTE Introduction:

1. RTE is responsible to establish link between AUTOSAR software components.
2. Software components communicate via this link provided by RTE.
3. RTE acts as means by which AUTOSAR software components access basic software modules including the OS and Communication service.
4. RTE implements the VFB for each ECU.
5. RTE abstracts the software component layer from implementation of the Basic software and from the hardware.
6. RTE enables software components to be reused in different ECUs.

Interfaces in AUTOSAR:

AUTOSAR INTERFACE:

An “AUTOSAR Interface” defines the information exchanged between software components and/or BSW modules. This description is independent of a specific programming language, ECU or network technology. AUTOSAR Interfaces are used in defining the ports of software-components and/or BSW modules. Through these ports software-components and/or BSW modules can communicate with each other (send or receive information or invoke services). AUTOSAR makes it possible to implement this communication between Software-Components and/or BSW modules either locally or via a network.

STANDARDIZED AUTOSAR INTERFACE:

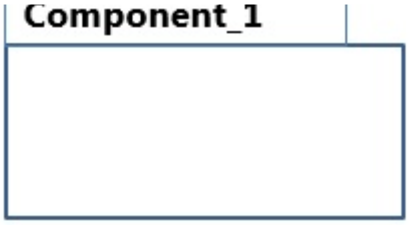
A "Standardized AUTOSAR Interface" is an "AUTOSAR Interface" whose syntax and semantics are standardized in AUTOSAR. The "Standardized AUTOSAR Interfaces" are typically used to define AUTOSAR Services, which are standardized services provided by the AUTOSAR Basic Software to the application Software-Components.

STANDARDIZED INTERFACE:

A "Standardized Interface" is an API which is standardized within AUTOSAR without using the "AUTOSAR Interface" technique. These "Standardized Interfaces" are typically defined for a specific programming language (like "C"). Because of this, "standardized interfaces" are typically used between software-modules which are always on the same ECU. When software modules communicate through a "standardized interface", it is NOT possible any more to route the communication between the software-modules through a network.

RTE Entities:

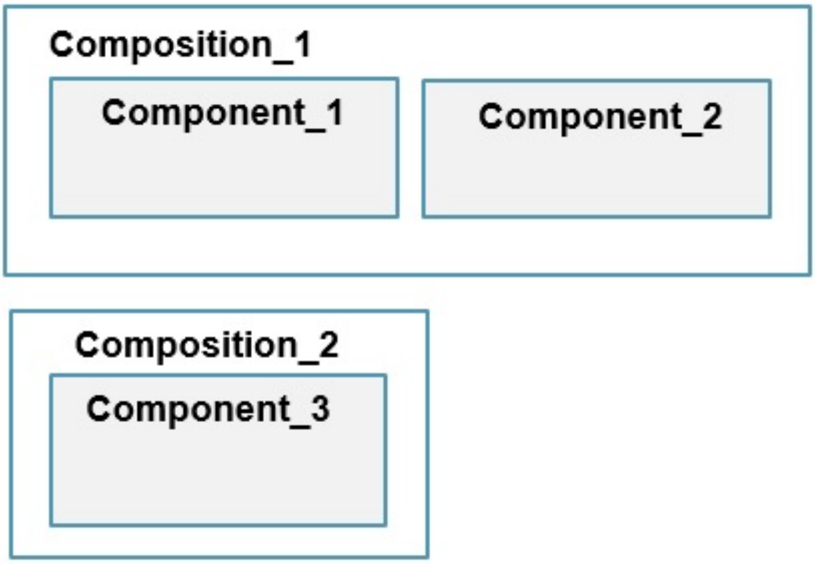
SOFTWARE COMPONENT:



Different types of software components:

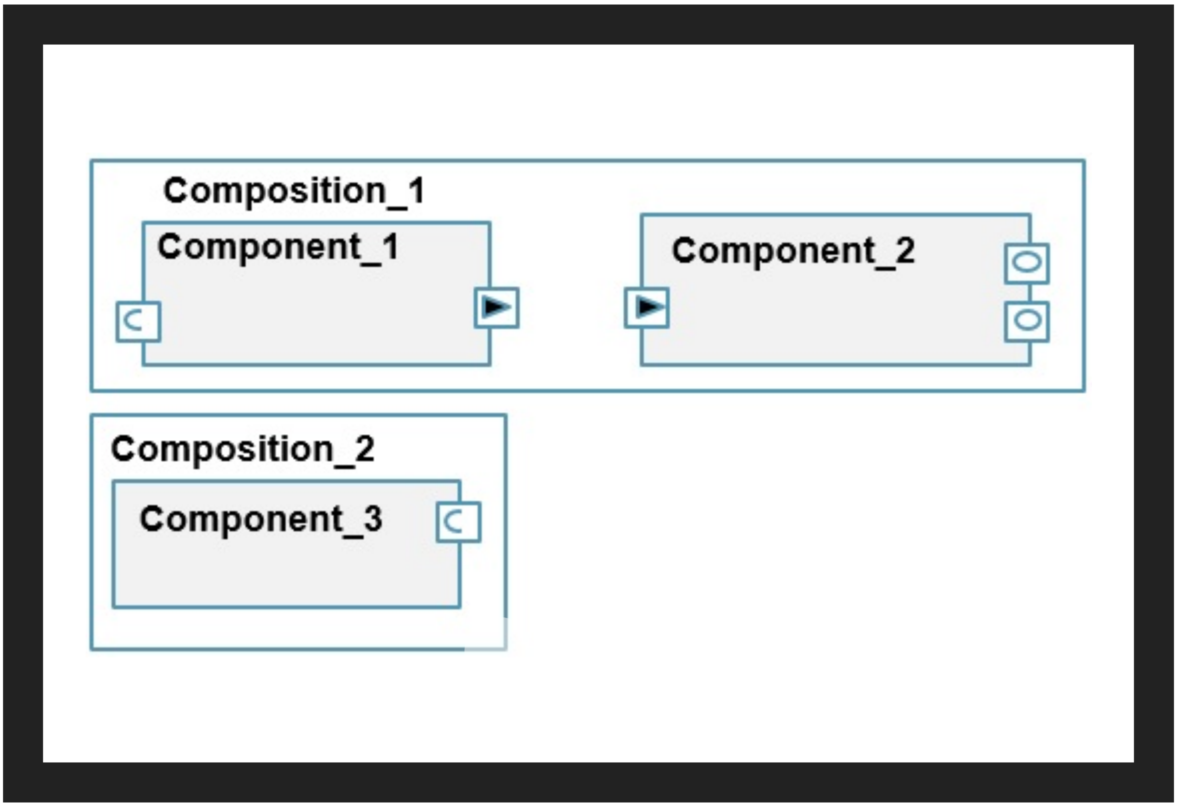
1. Application Software Component
2. Sensor Actuator Component
3. Calibration Parameter Component
4. Service Component
5. ECU Abstraction Component
6. Complex Device Driver Component

COMPOSITION:



Used as structuring elements to build up a hierarchical system.

PORT:



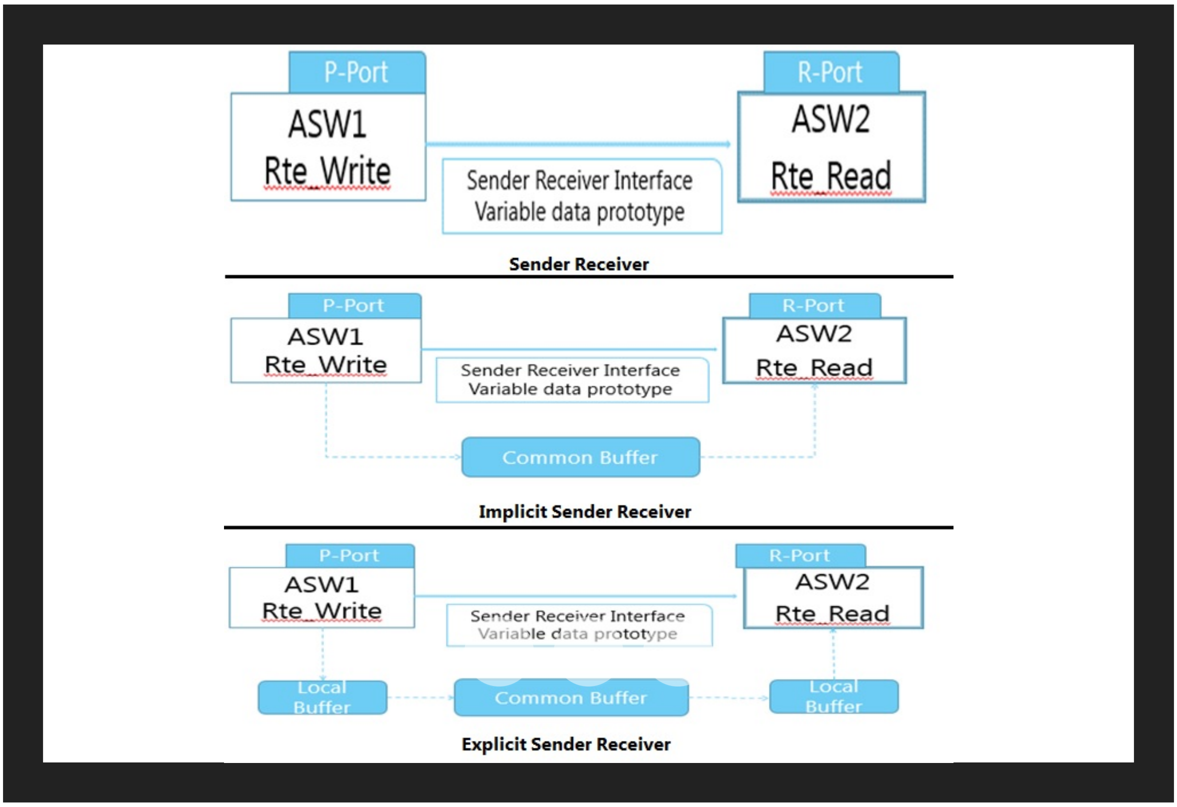
Type of ports used in RTE:

1. Provider Port
2. Receiver Port

PORT INTERFACE:

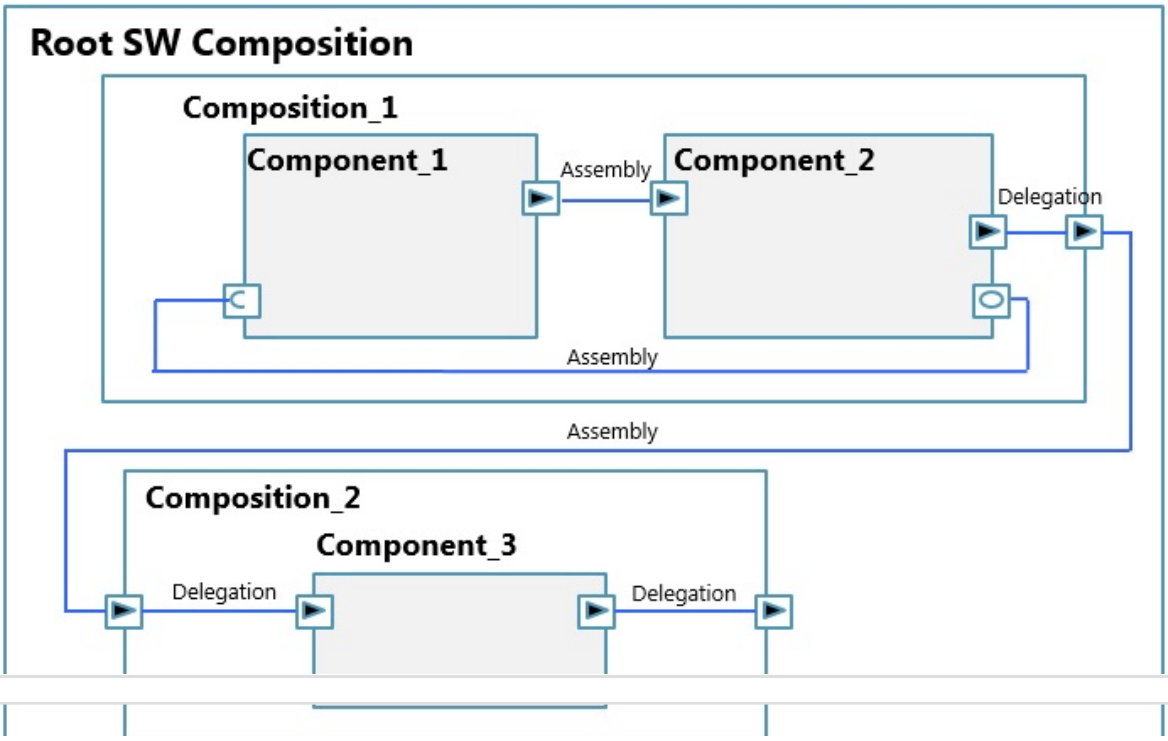
Different types of interfaces:

1. Sender Receiver Interface
2. NV Data Interface
3. Mode Switch Interface
4. Client Server Interface
5. Parameter Interface
6. Trigger Interface



ELEMENTS:

CONNECTOR:



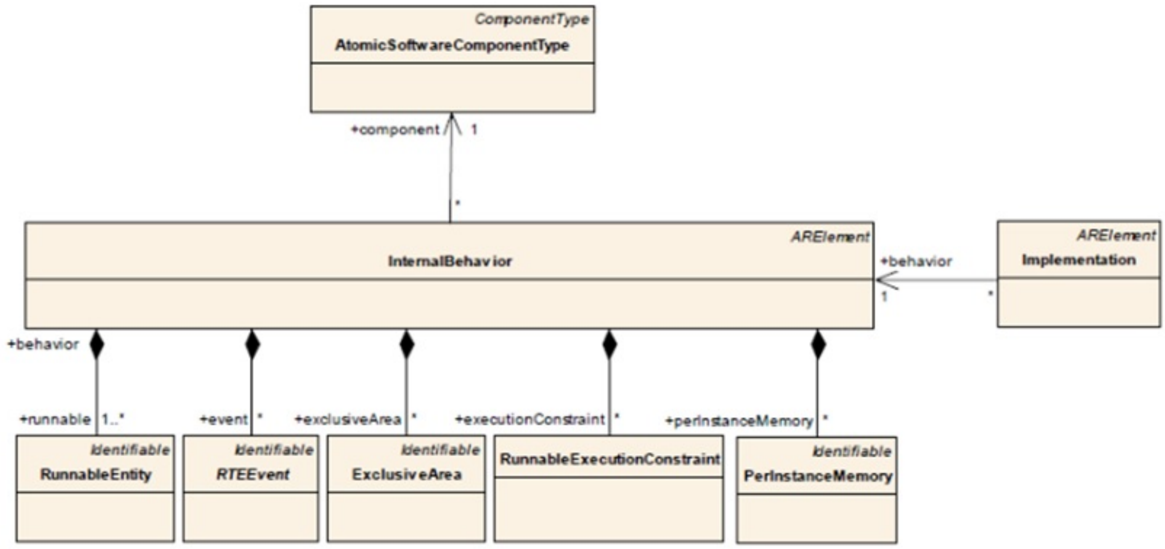
Type of connectors:

1. Assembly
2. Delegation

INTERNAL BEHAVIOUR:

Internal structure of the Software Component and consists of:

1. Runnable Entity
2. RTE Events
3. Exclusive Areas
4. Runnable Execution Constraints
5. Per Instance Memory



RUNNABLE ENTITY:

Runnable Entity are schedulable unit of a Software Component. Runnable Entity are ‘C’ functions within the Software Component. Runnable Entity are triggered by RTE Event.

Share this:

Twitter Facebook

Like

Be the first to like this.