

# Optimizing Automotive Systems and Software Design and Development

## Using Model Driven Development to Optimize the AUTOSAR Development Process

May 2007

Rick Boldt

Senior Director

Automotive Industry Marketing

Rhapsody Product Marketing

Torsten Schoefbeck

Applications Engineer

# Agenda

- Introduction to AUTOSAR
- AUTOSAR Process
- Types of AUTOSAR Modeling Tools
  - AUTOSAR Systems Modeling
  - AUTOSAR Behavioral Modeling
- A UML/SysML based AUTOSAR Systems Modeling Solution
- Summary

# Automotive Electronics Dilemmas...

- Electronics and software importance is growing
  - Electronics account for 22% of a vehicle's cost[1]
    - Projected to increase to 40% by 2010[1]
  - Software accounted for 4% of total vehicle cost in 2002 [2]
    - Projected to increase to 13% by 2010[2]
  - Often is used as a product differentiator
  - Provides new found flexibility
- Shrinking development schedules
  - New vehicle development cycles are down from 36 months in the mid 1990's to 24 months or less today[3]
    - This is causing tremendous pressure on electronic development already
  - Over a 20 year period development times have gone from 60 months down to 24 months with further reductions to 18 and even 12 months targeted[4]
- Resources are fixed & lowering costs while complexity increases
  - Trend to develop more vehicles from less budget will continue into the next decade [3]

[1] Source: Databeans Estimate, [2] Source: Strategy Analytics, [3] Source: Roland Berger, [4] Source: Wards Auto World

# What is AUTOSAR

- Automotive Open System Architecture (AUTOSAR)
- Consortium of Automotive OEMs and Suppliers
- Objective: To create a basis for industry collaboration on basic functions while providing a platform which continues to encourage competition on innovative functions with the following goals:
  - Implementation and standardization of basic system functions as an industry wide "Standard Core" solution
  - Scalability to different vehicle and platform variants
  - Transferability of functions throughout network
  - Integration of functional modules from multiple suppliers
  - Consideration of availability and safety requirements
  - Redundancy activation
  - Maintainability throughout the whole "Product Life Cycle"
  - Increased use of "Commercial off the shelf hardware"
  - Software updates and upgrades over vehicle lifetime

# Key AUTOSAR Features

It will be possible to map functions and functional networks to different control nodes in the system, almost independently from the associated hardware.

- Modularity and configurability
  - Definition of a modular software architecture for automotive electronic control units
- Standardized interfaces
  - Standardization of different APIs to separate the AUTOSAR SW layers
- Runtime environment
  - Provides inter- and intra-ECU communication across all nodes of a vehicle network
  - Located between the functional SW-components and the basic SW-modules

# AUTOSAR Membership

- Core Partners
  - Driving Force
  - Overall definition of Autosar
  - Participate in working groups
  - BMW, Bosch, Continental, DaimlerChrysler, Ford, General Motors, PSA, Siemens VDO, Toyota, Volkswagen
- Premium Members
  - Participate in working groups
  - Early access to technology and approved documents
  - Telelogic is a premium member and contributes to the specifications
- Associate Members
  - Early access to technology and approved documents

# AUTOSAR Process

# AUTOSAR Process Overview

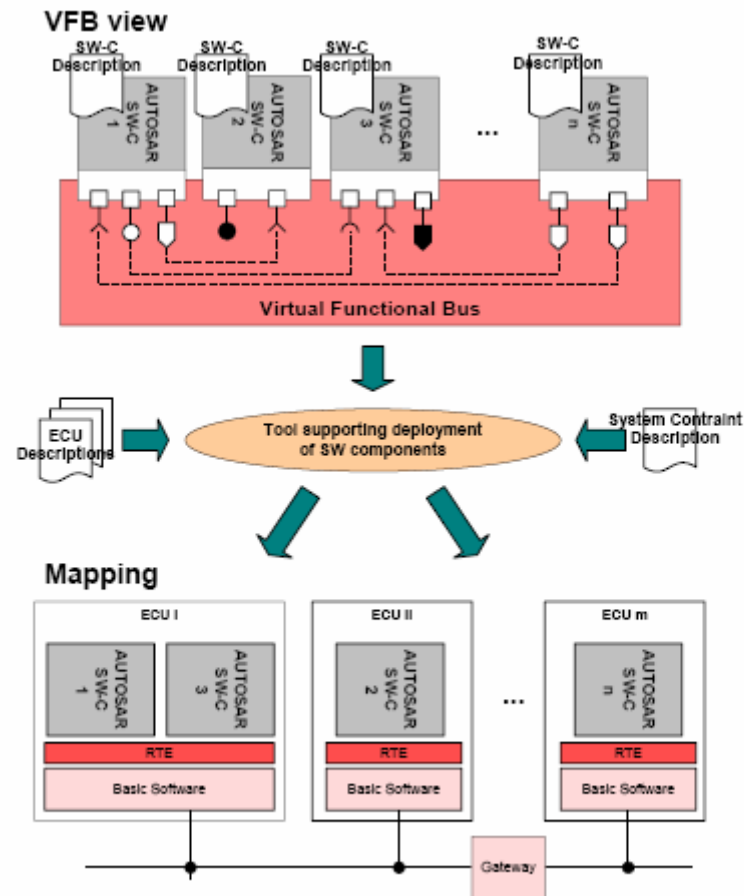


Figure 2: Basic AUTOSAR approach.

# AUTOSAR Functional System

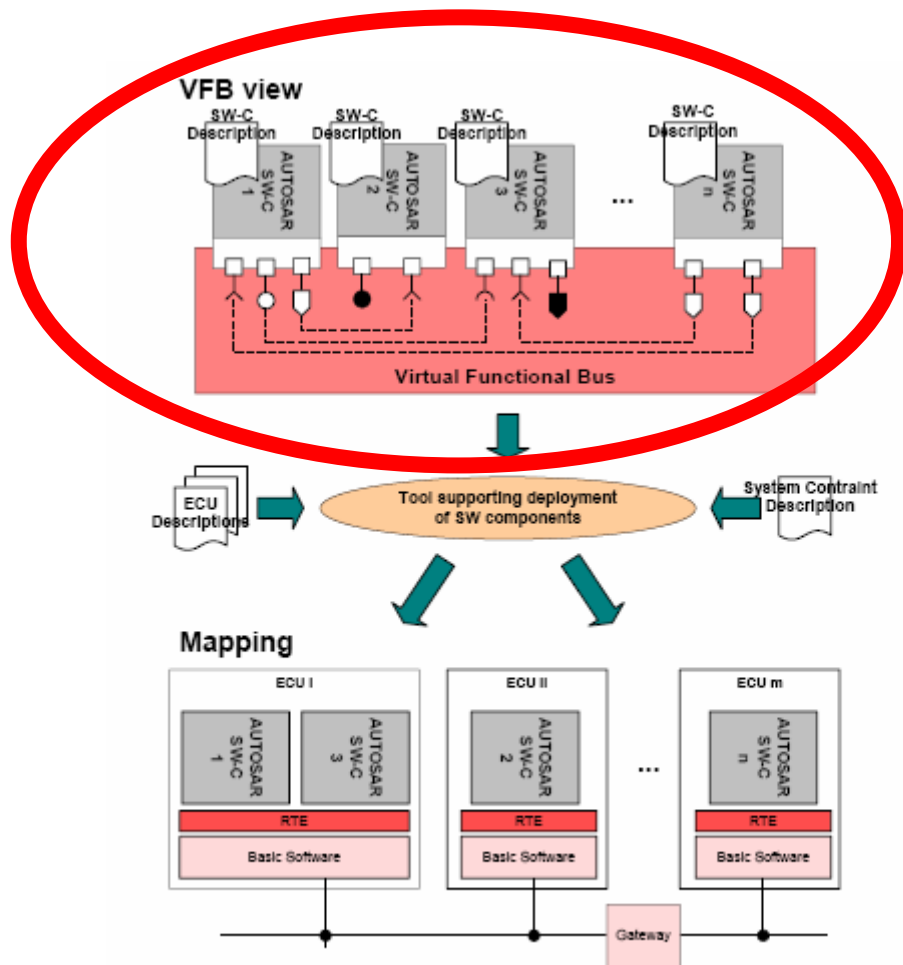
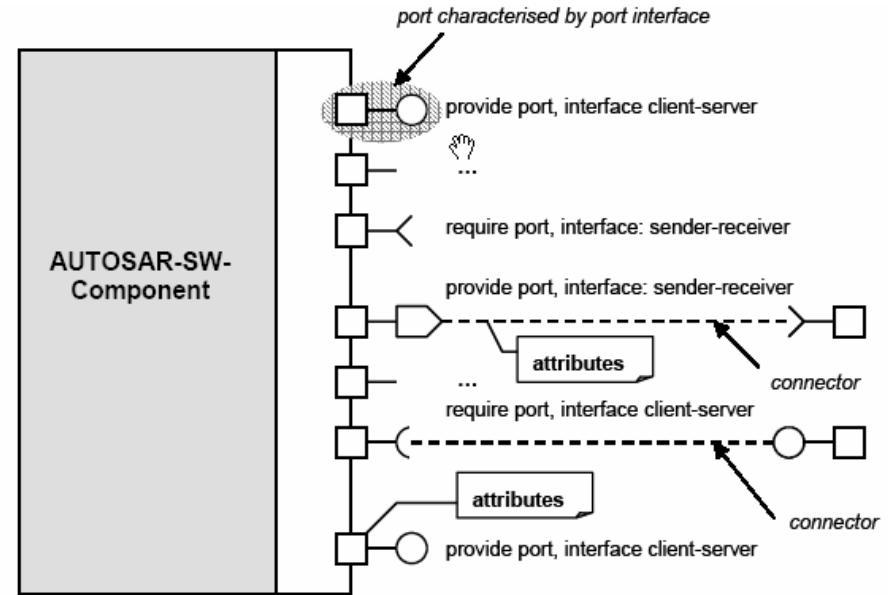


Figure 2: Basic AUTOSAR approach.



- Consists of interacting SW-Cs
- Can form SW-Compositions
- Interaction is over the VFB
- Two options
  - Sender/Receiver
  - Client/Server

# AUTOSAR Software Components

- An AUTOSAR system is built as a set of interacting *AUTOSAR Software Components (SW-C)* integrated by the AUTOSAR VFB – Virtual Functional Bus.
  - AUTOSAR defines a *layered* software architecture.
- AUTOSAR Software Component (SW-C)
  - An “**Atomic**” Software Component and cannot be distributed over several ECUs.
    - Contains everything that is needed to allow several AUTOSAR Software Components to be integrated correctly in an ECU or network of ECUs.
    - AUTOSAR does not prescribe how an AUTOSAR Software Component should be implemented.
  - A **Software Component Description** which specifies the component interface and how the infrastructure must be configured for the component.
    - Interface: the operations and data elements that the software component provides and requires.
    - The requirements that software component has on the infrastructure.
    - The resources needed by the software component (memory, CPU-time, etc.)
    - Information regarding the specific implementation of the software component
    - Shipped as an XML document.
- AUTOSAR Virtual Functional Bus (VFB)
  - Defines the **conceptual** framework and communication mechanisms which allow integration (interfaces, ports, scheduling policies, error handling...).

# AUTOSAR Vehicle Electrical Architecture

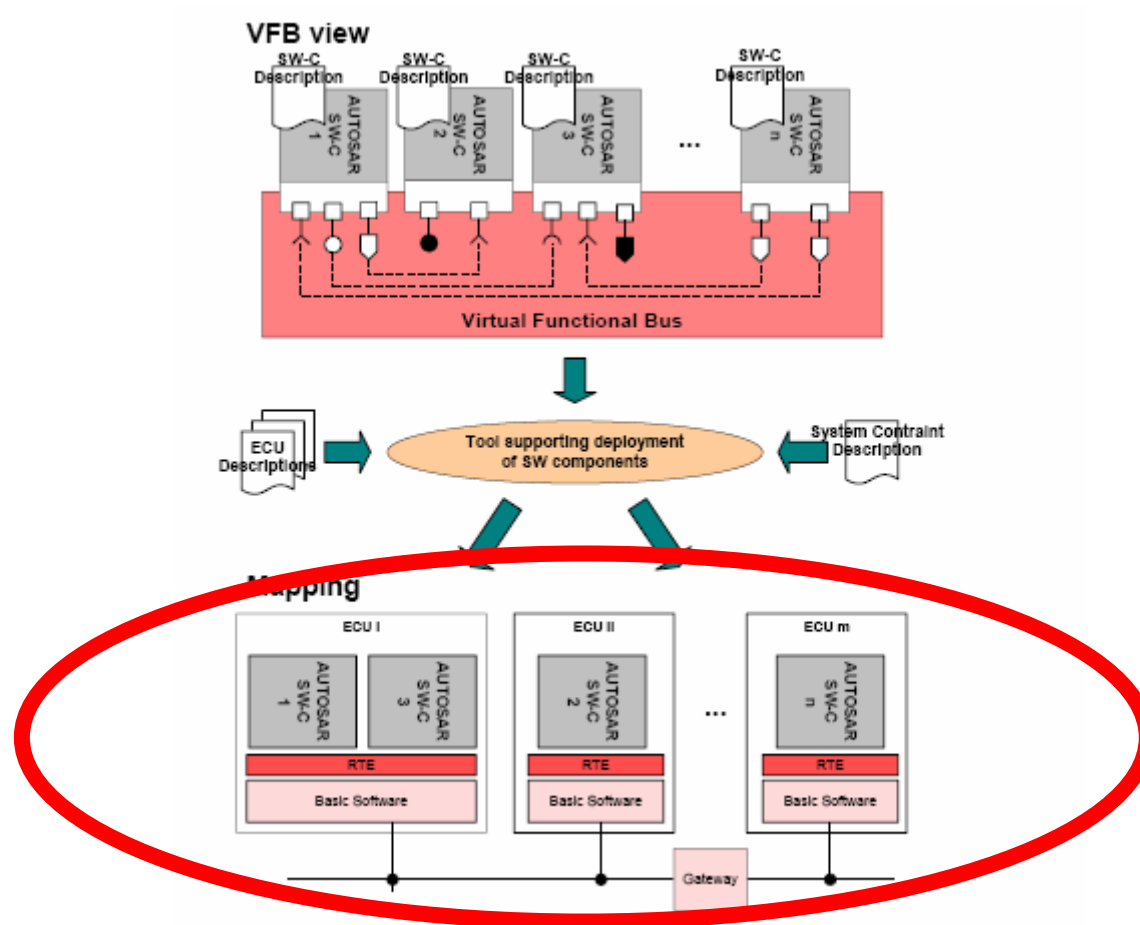
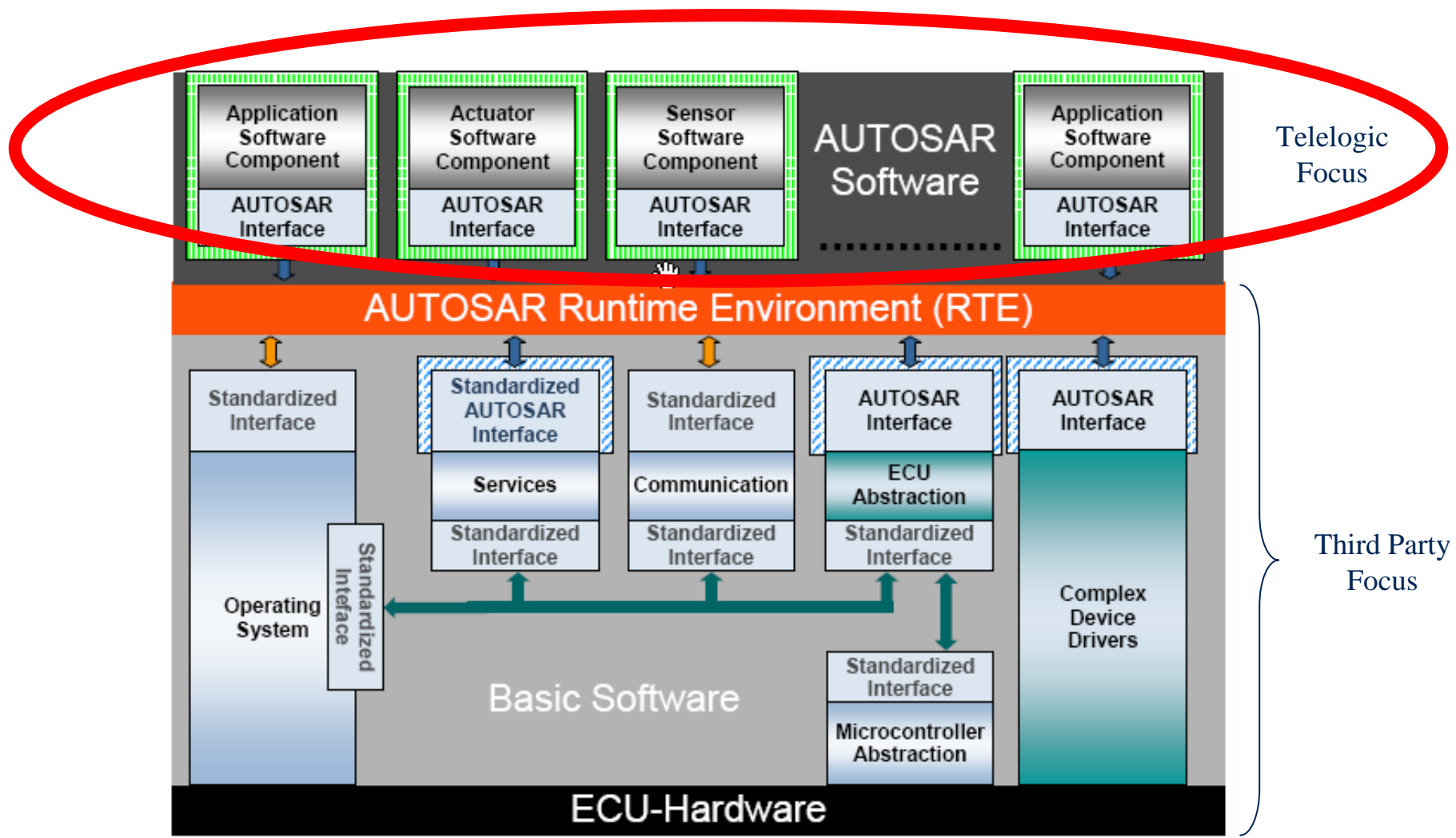
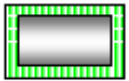



Figure 2: Basic AUTOSAR approach.

# AUTOSAR ECU Architecture



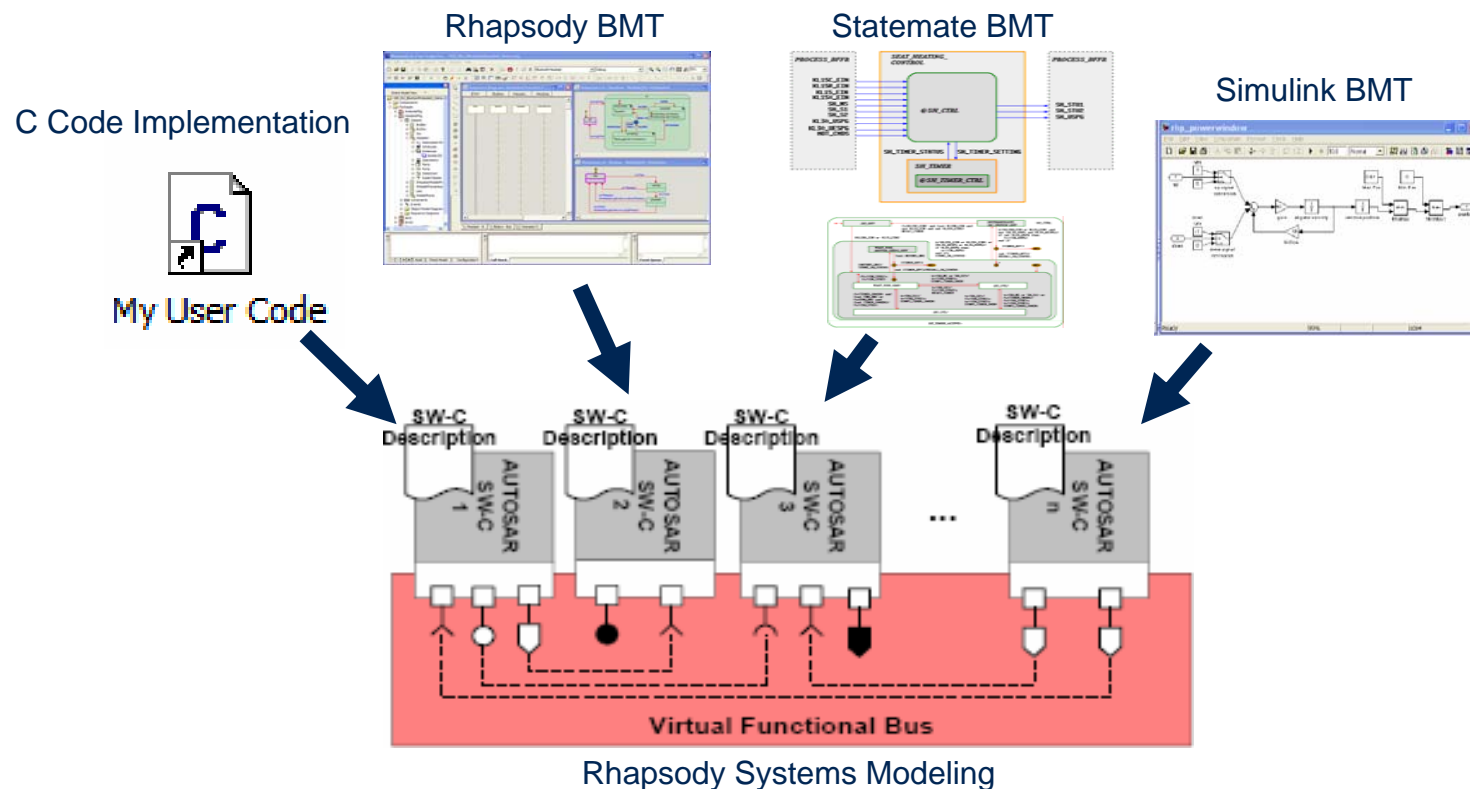
 The software component template describes these components completely

 Of these software components only the AUTOSAR Interface side can be fully described in the software component template

# Types of AUTOSAR Modeling Tools

# Two Types of AUTOSAR Modeling Tools

- AUTOSAR System Modeling
  - Defines the architecture and integrates the algorithms
- AUTOSAR Behavioral Modeling Tools (BMT)
  - Implement the algorithms that plug into the architectures



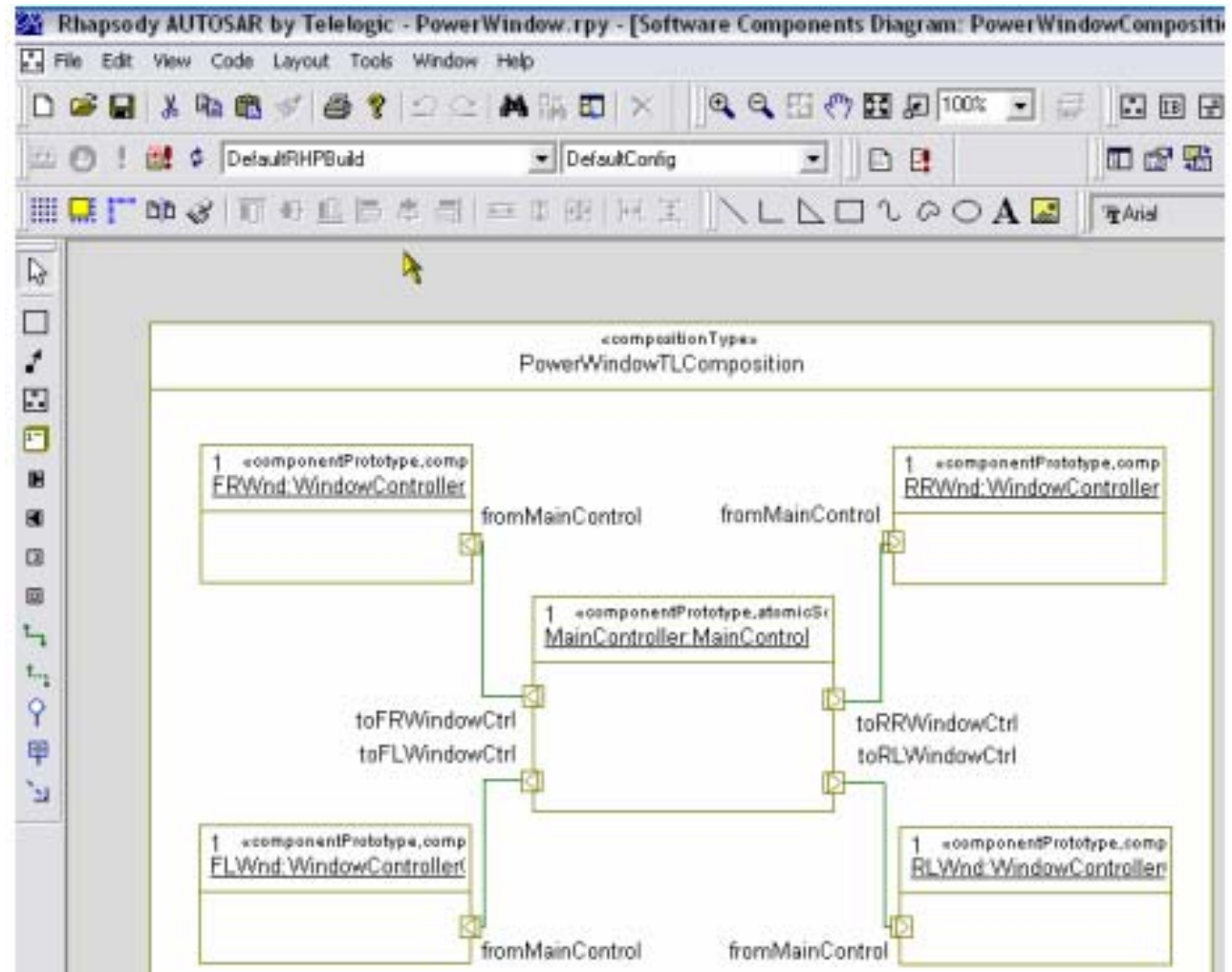
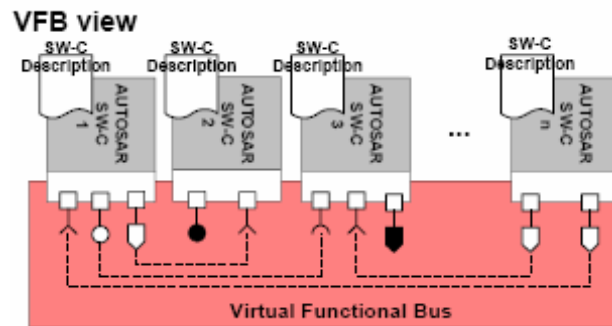
# AUTOSAR System Modeling Using UML/SysML

# UML/SysML Based AUTOSAR System Modeling Overview

- AUTOSAR modeling environment based on UML/SysML profile
  - Software Component Diagram
  - Internal Behavior Diagram
  - ECU Diagram
  - Topology Diagram
  - Systems Diagram
- AUTOSAR XML import and export enables process fit
- Implementation independent designs
  - Enabling deployment on multiple architectures
  - Supporting software component specification reuse
  - Increasing brand consistency
  - Improving time-to-market

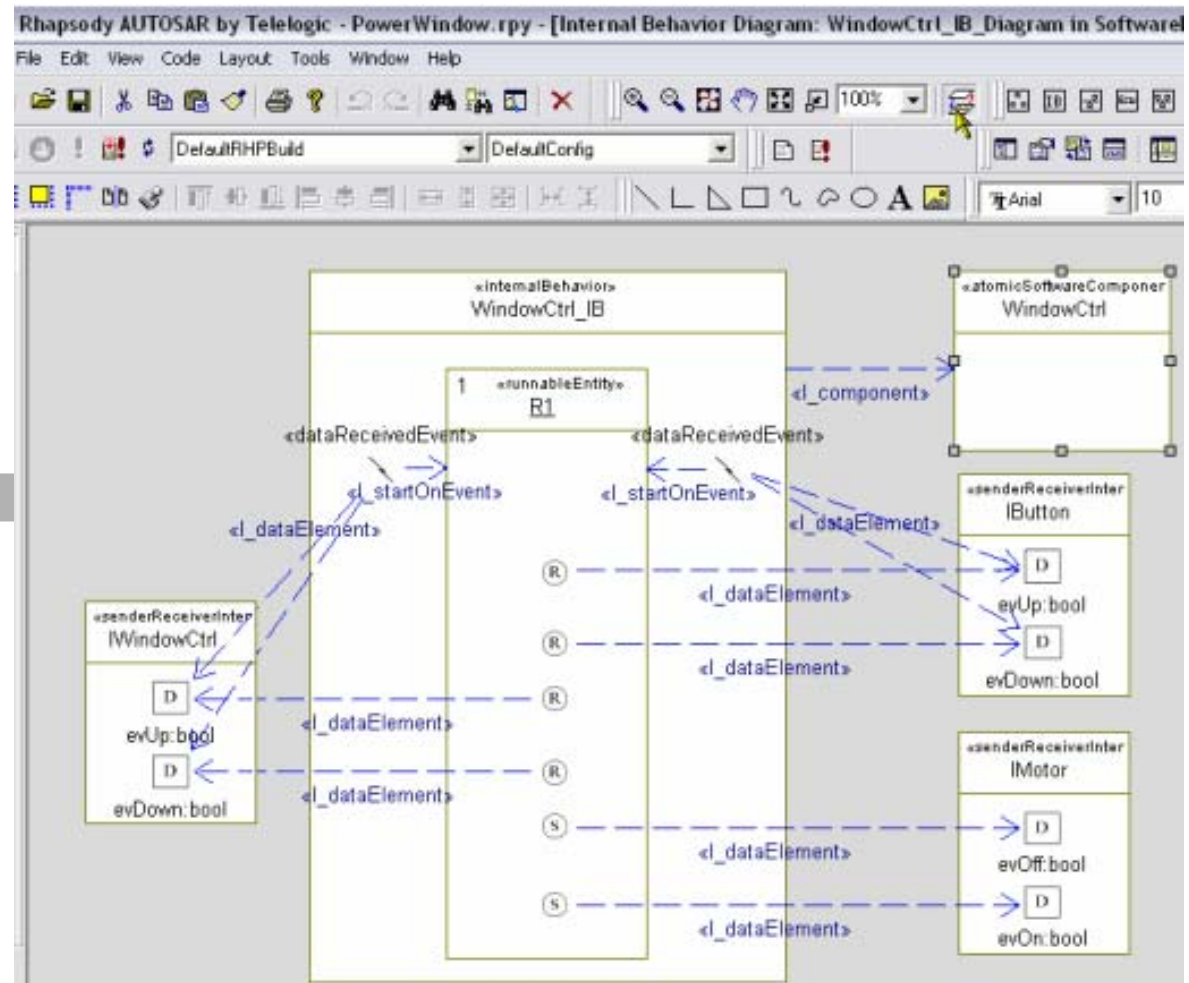
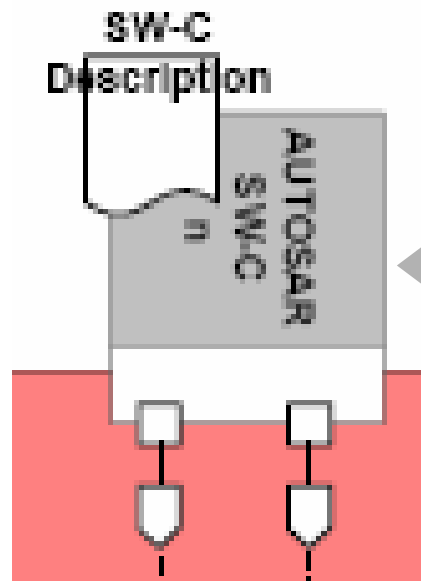
# Software Component Diagram

- Defines the software architecture of the functional system



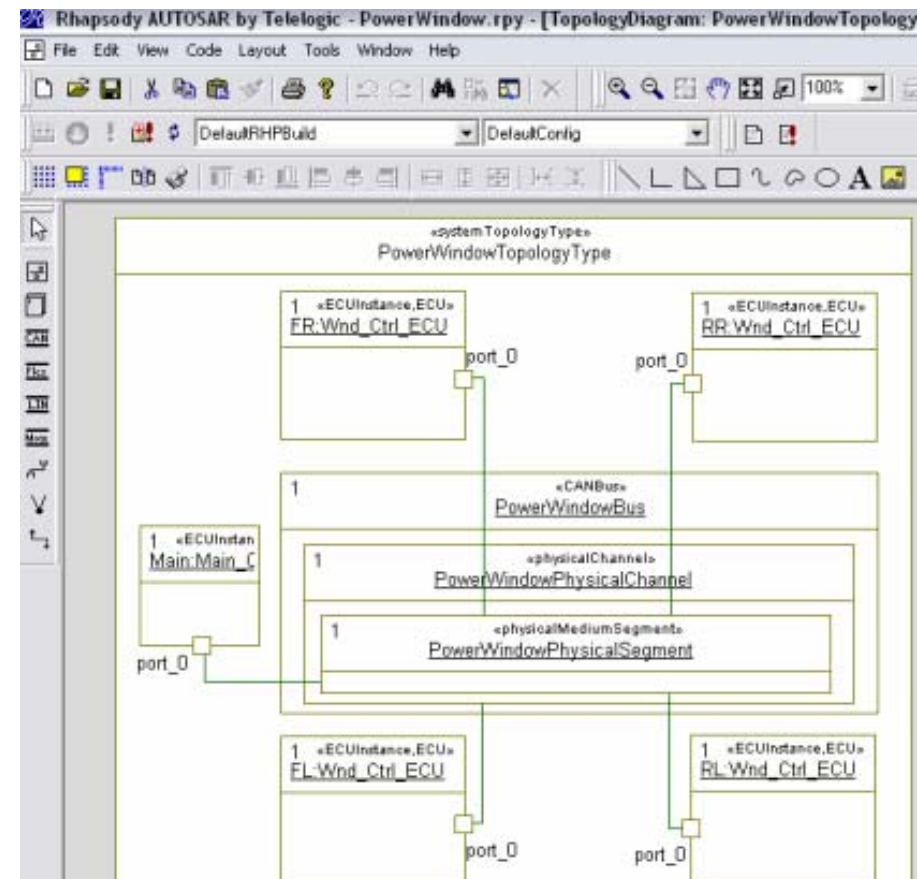
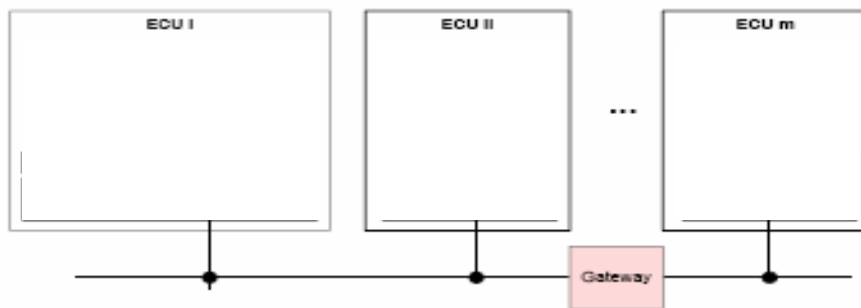
# Internal Behavior Diagram

- Specifies the internal scheduling and interface to the RTE for a software component



# Topology and ECU Diagrams

- Topology Diagram defines the physical topology or electrical architecture of the system including all the ECUs in the automobile and how they are connected.
- ECU Diagram defines the Electronic Control Unit (ECU) types and their communication ports



# Systems Diagram

- Captures the overall system including how the AUTOSAR SW-Cs map to the individual ECUs that make up the electrical architecture and the communication matrices between the ECUS.

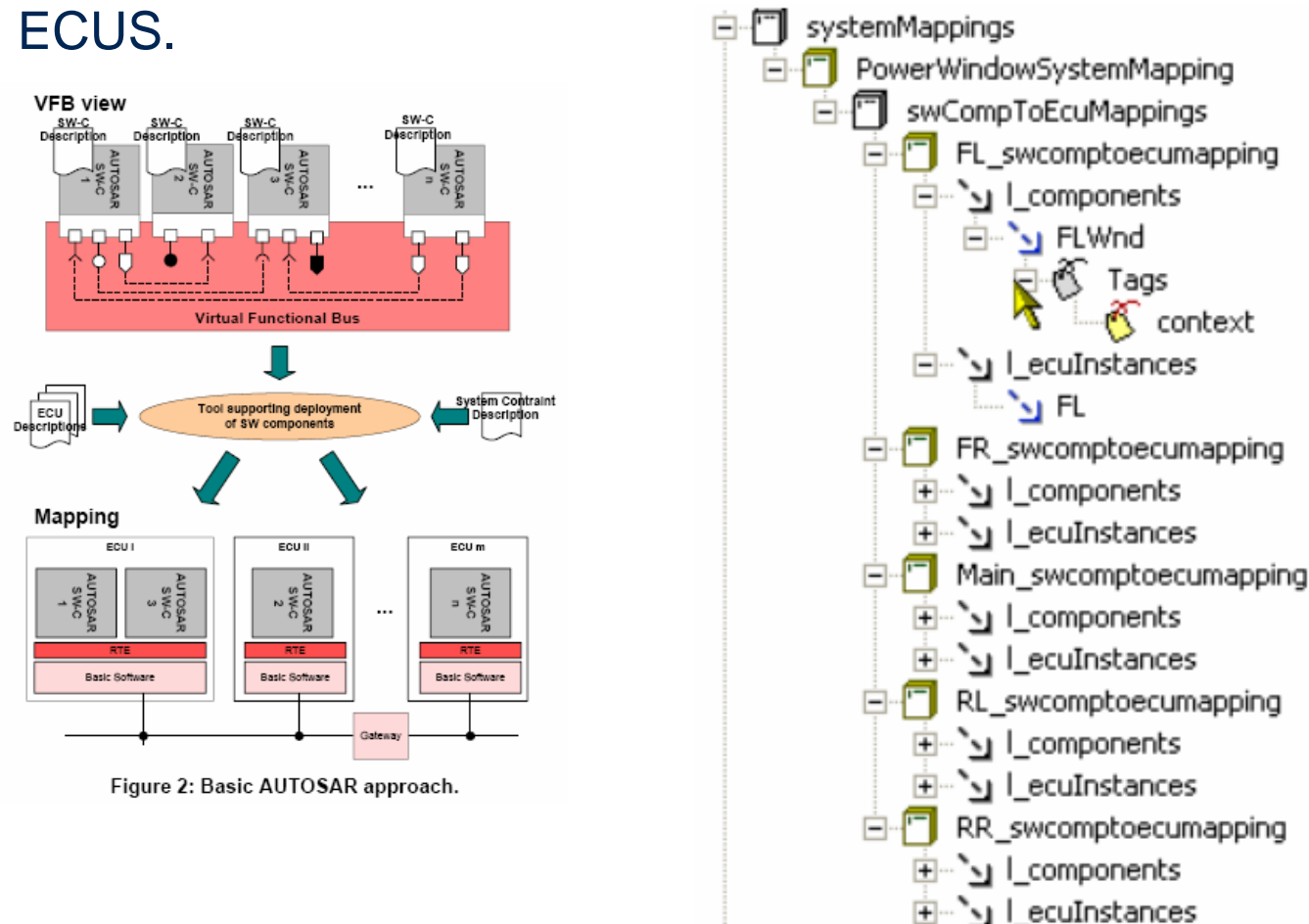


Figure 2: Basic AUTOSAR approach.

# Working with other AUTOSAR Tools

- Multiple tools are required in the AUTOSAR Methodology
- AUTOSAR XML Document is used for the exchange
- Any AUTOSAR compliant tool must support import and export of AUTOSAR XML

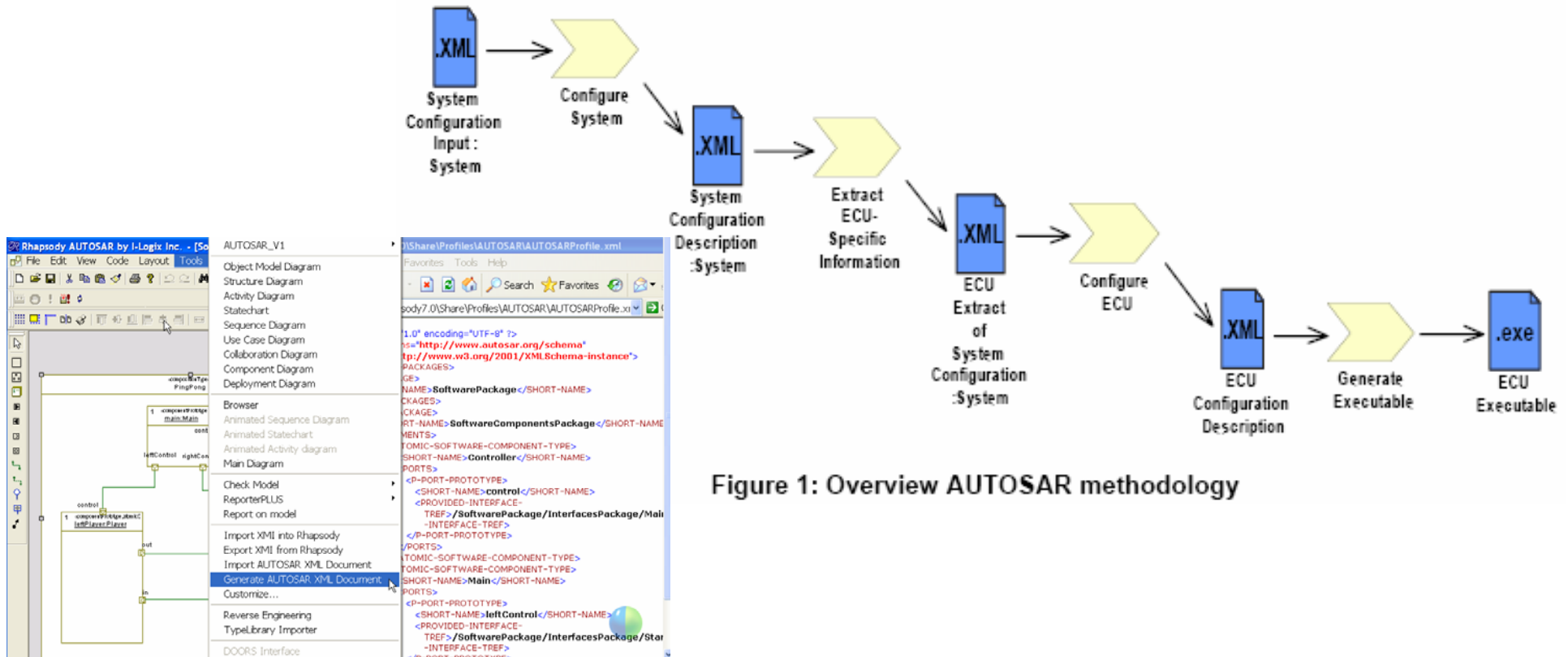


Figure 1: Overview AUTOSAR methodology

# Summary

- AUTOSAR is a response to the increasing challenges and opportunities that come with the growing importance of software and electronics in the vehicle.
- The AUTOSAR design process is based around capturing the AUTOSAR System
  - Functional system of interacting software components
  - Electrical architecture for a specific vehicle
  - Mapping of software components to ECUs
  - A profile based on UML/SysML can effectively capture this information
  - AUTOSAR XML must be used to exchange information with other tools in the process
  - Source code or Behavioral Modeling Tools (BMT) may be used to capture the algorithms