



爱开发 重创新 更智慧

Innovate2011

IBM Rational 软件创新论坛

 Software. Everywhere.



IBM解决方案如何支持汽车电子系统软件架构标准AUTOSAR

周末

IBM Rational高级技术顾问



内容安排

- AUTOSAR 的由来
- AUTOSAR技术架构
- AUTOSAR开发流程
- Rhapsody对AUTOSAR的支持
- Rhapsody和Simulink的关系
- 总结

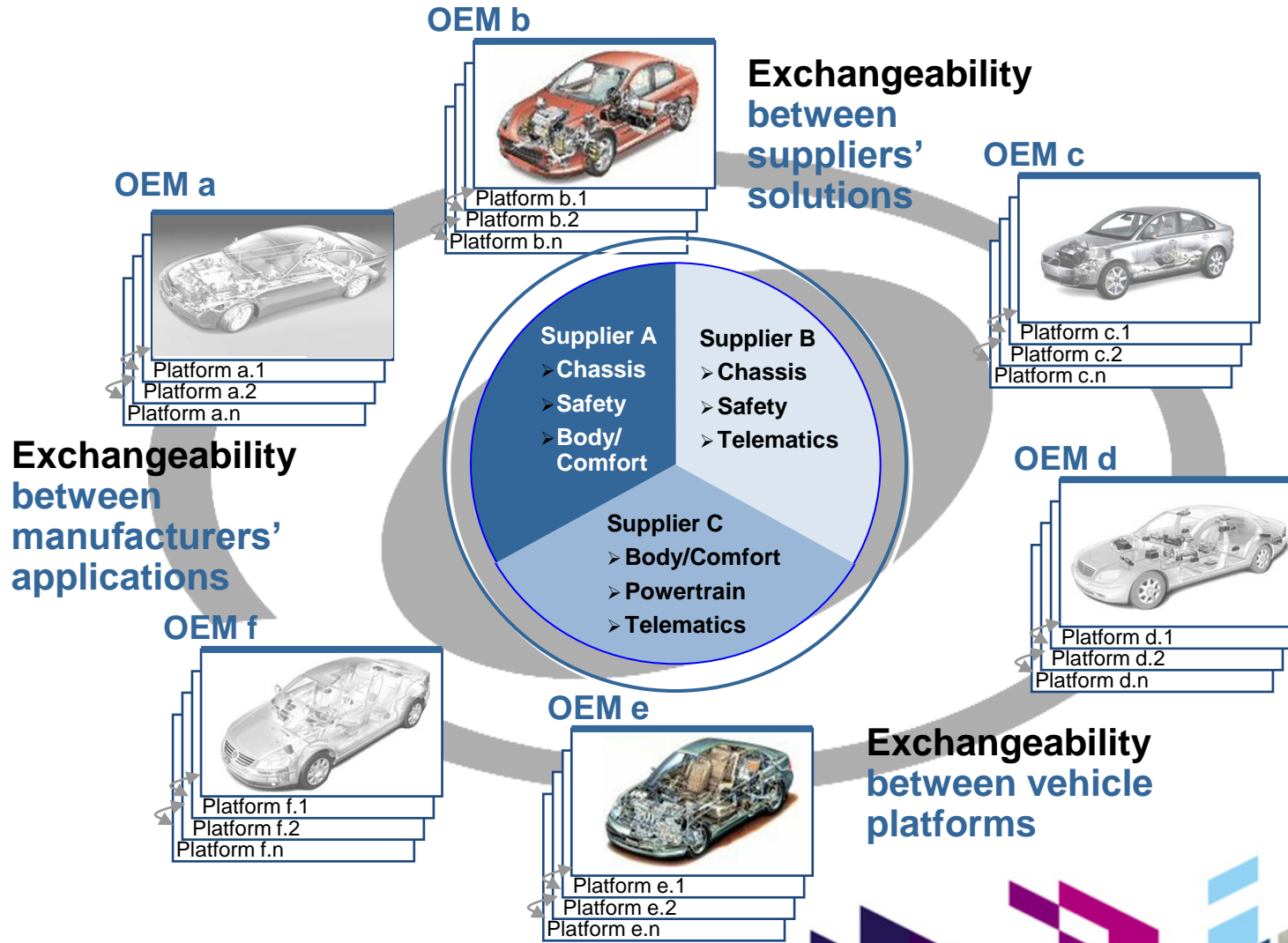
AUTOSAR由来

- 随着汽车电子的复杂性不断增强，领先的汽车OEM和Tier1 零部件供应商共同认识到汽车电子架构的3大问题
 - 法规遵循，包括环保和安全性强制标准
 - 乘客需要更多的娱乐和舒适方面的需求
 - 驾驶员需要更多的驾驶辅助设施以应对复杂、高密度的驾驶环境
- 2003年，由欧洲的汽车OEM和来自ECU、软件以及电子供应商推出了AUTOSAR（AUTomotive Open System ARchitecture）标准
 - 目的1：统一汽车电子软件的运行平台和接口，从而达到高度重用性和互换性
 - 目的2：鼓励在统一平台和接口的基础上，在开发更多消费者喜爱的方面展开竞争
 - 在标准上合作，在实现上竞争
- 日本在2007年中，成立了类似的组织JASPAR（Japan Automotive Software Platform and ARchitecture）



AUTOSAR 愿景1

AUTOSAR 通过在OEM和供应商之间增加软件模块的重用性和可互换性来降低复杂度

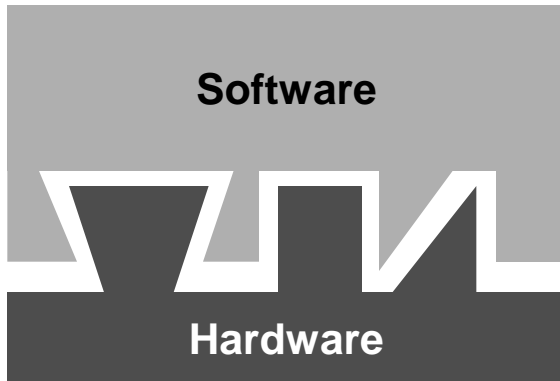


Motivation and Principles
Visions and Objectives

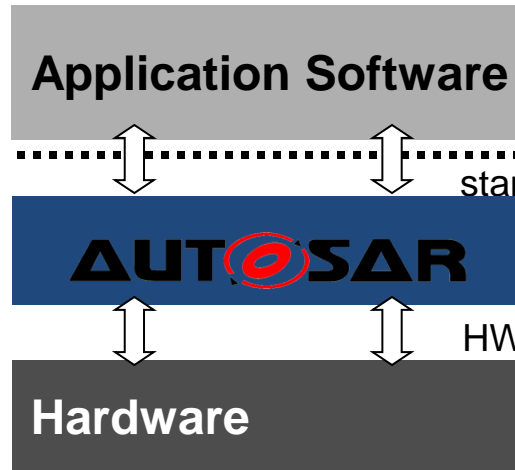
AUTOSAR 愿景2

AUTOSAR 统一ECU的软件接口标准

Yesterday



AUTOSAR



Customer needs

- Adaptive Cruise Control
- Lane Departure Warning
- Advanced Front Lighting System
- ..

Using standards

- Communication Stack
- OSEK
- Diagnostics
- CAN, FlexRay

- 软硬件高度独立
- 分层次开发的思想（类似OSI7层协议），降低开发成本和难度
- 软件重用提高质量和开发效率

AUTOSAR – Core Partners and Members Status September 2010

9 Core Partners

11 Development Members

47 Premium Members

63 Associate Members
6 Attendees

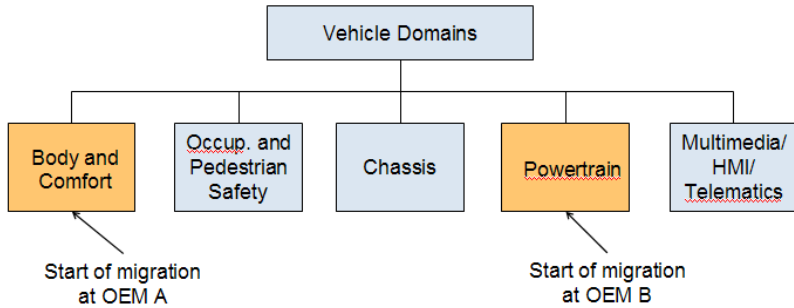


Up-to-date status see: <http://www.AUTOSAR.org>

Motivation and Principles
Development Cooperation

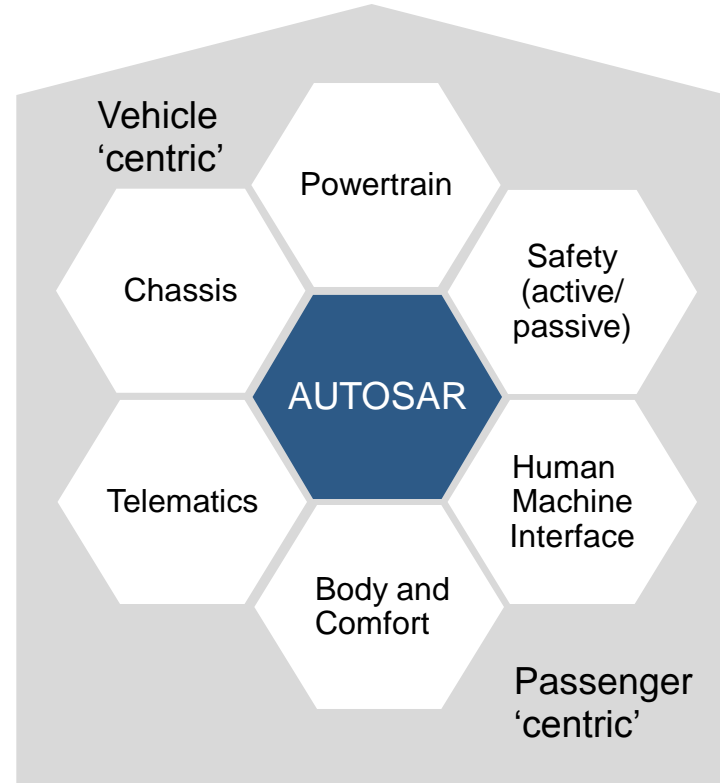
AUTOSAR 适用的领域

Motivation and Principles
Visions and Objectives



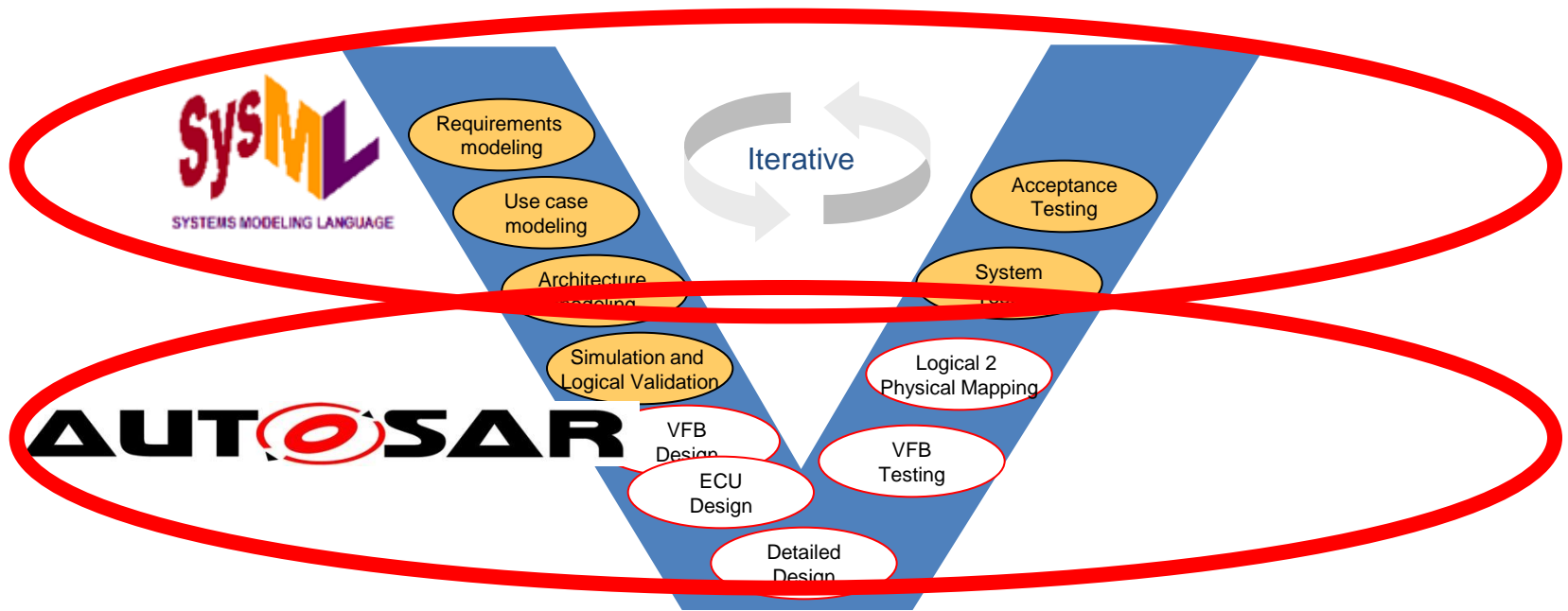
- 向AUTOSAR的演进是一个渐进的过程
- AUTOSAR目前的状态类似OSI7层协议，目的是变成TCP/IP + Java虚拟机类似的工业标准

功能领域



AUTOSAR & SysML在汽车系统开发中的位置

- SysML 主要用在系统功能分析和模块划分
- AUTOSAR 主要用在汽车电子软件的设计和实现



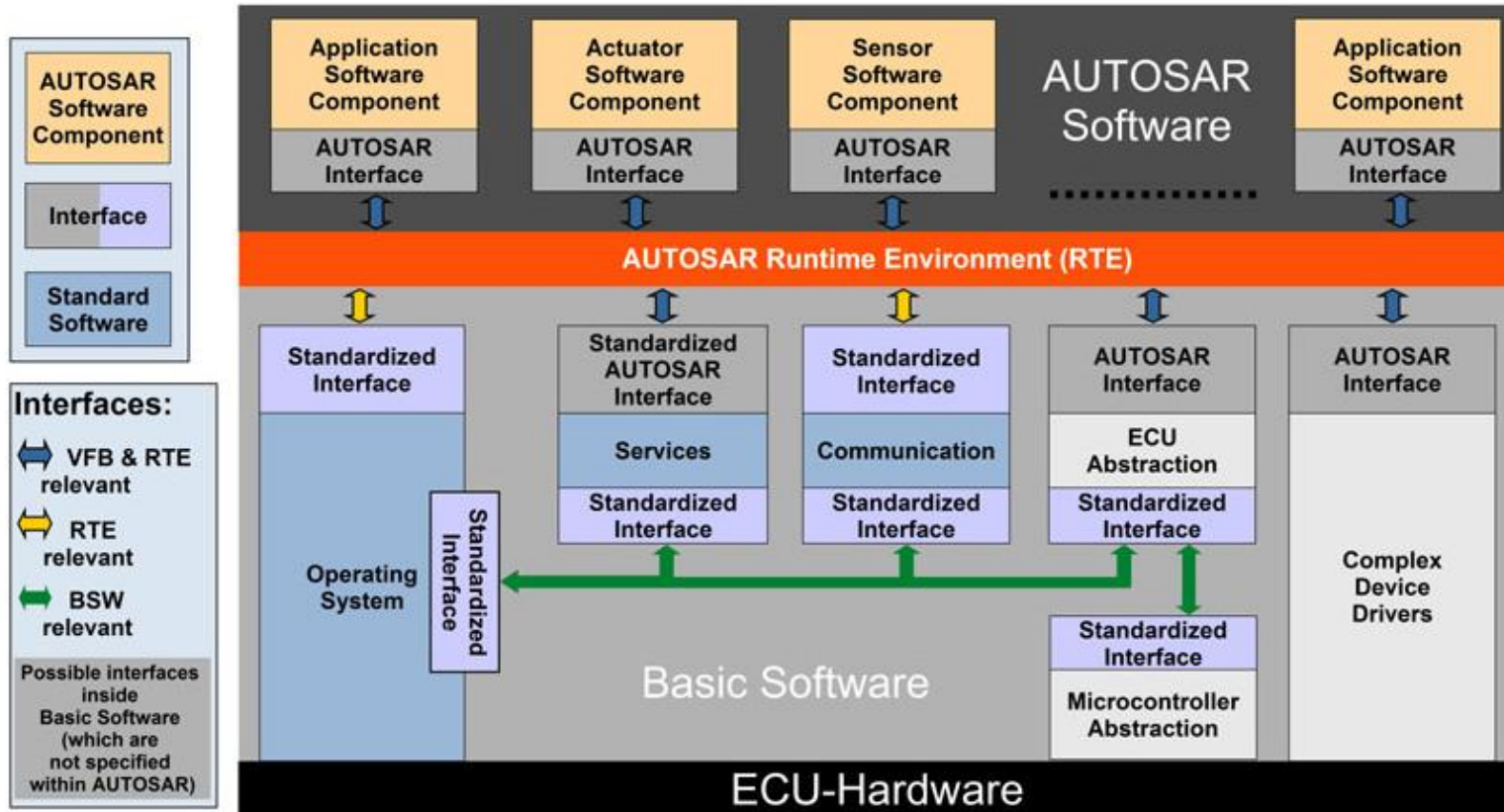
内容安排

- AUTOSAR 的由来
- AUTOSAR技术架构
- AUTOSAR开发流程
- Rhapsody对AUTOSAR的支持
- Rhapsody和Simulink的关系
- 总结

AUTOSAR架构

Interfaces

Components and interfaces view (simplified)



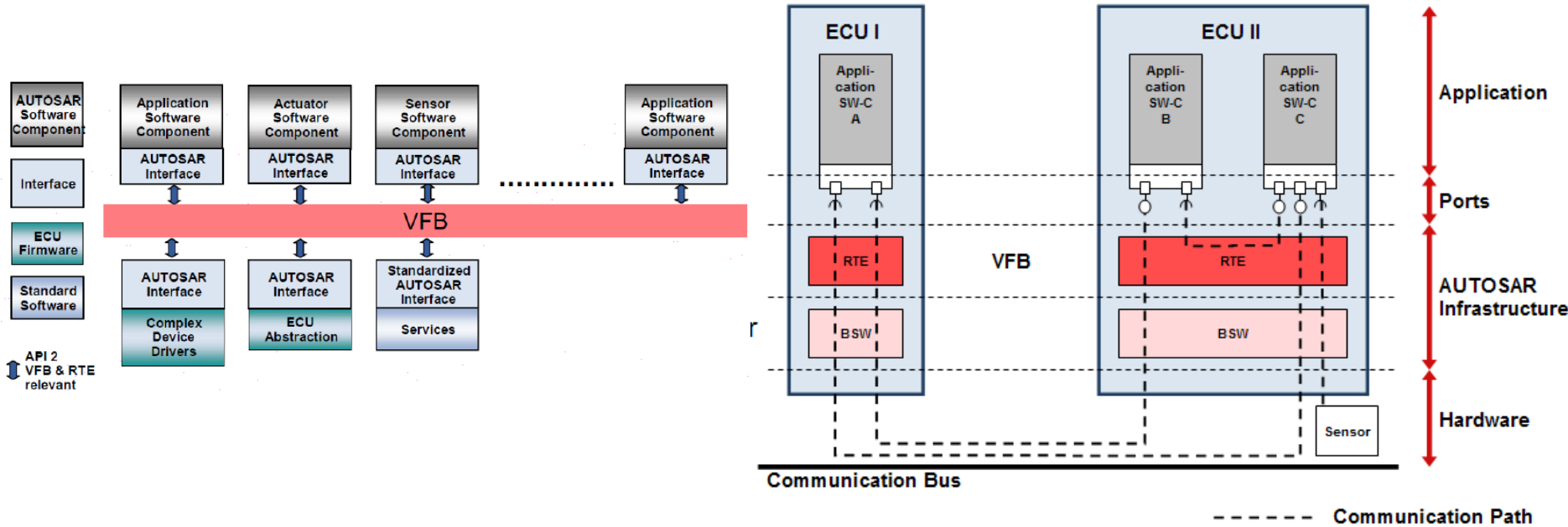
AUTOSAR 架构 – software component (SW-C)

- SW-C是一个软件功能模块，每个SW-C仅可被分配至一个ECU，一个ECU可拥有多个SW-C
- SW-C之间的通信，无论是intra-ECU或者inter-ECU，软件编程人员无需关心，类似于CORBA
- SW-C和RTE之间通过AUTOSAR interface进行通信，interface也是AUTOSAR的重要工作成果
- SW-C的描述方式为Description(software component template)+ implementation
- Sensor/Actuator是2种特别的SW-C，作用分别是物理信号转换为软件信号，以及驱动电机



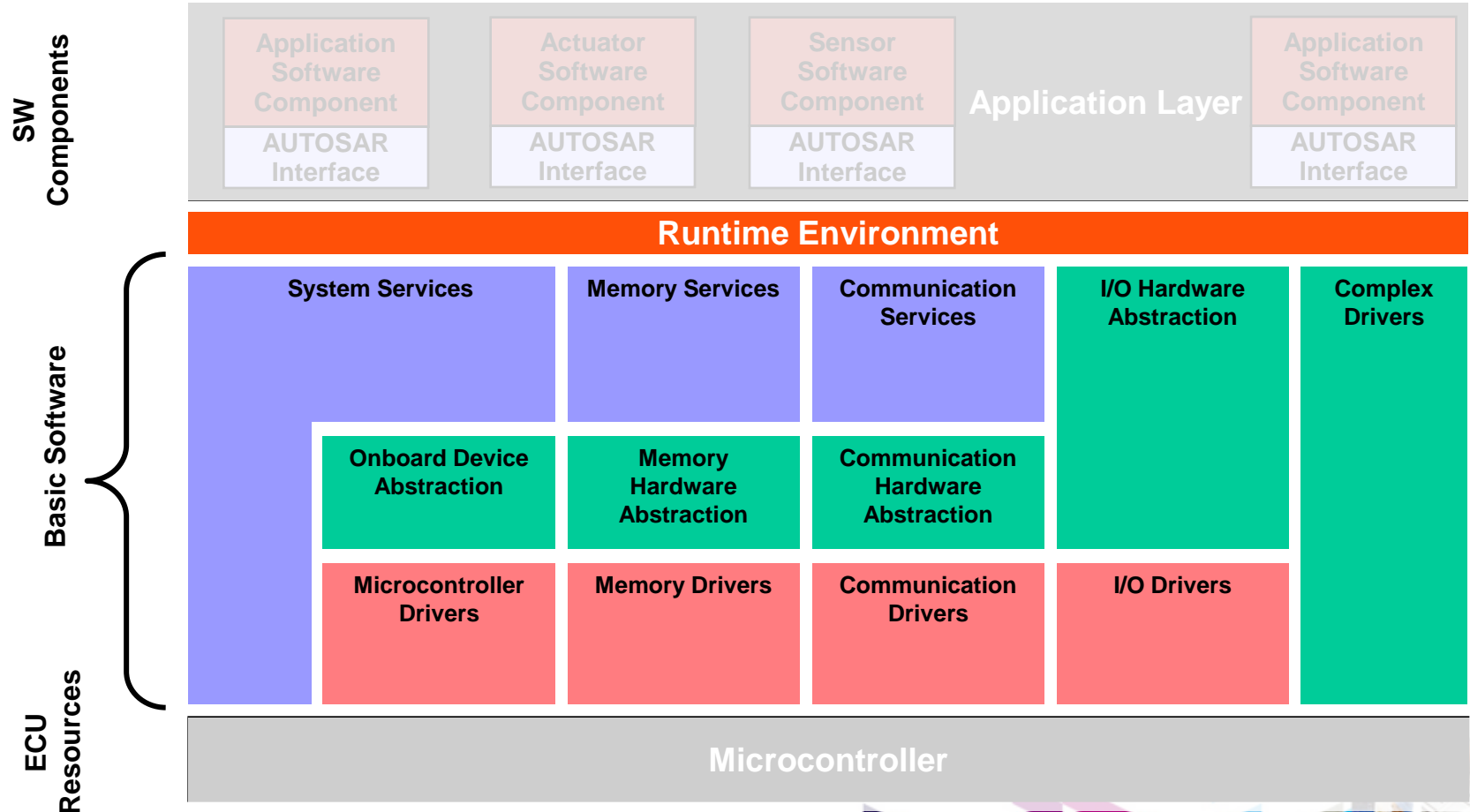
VFB and RTE

- RTE为SW-C（包含application和Sensor/Actuator）提供了通信服务，实现了VFB（virtual function bus）概念

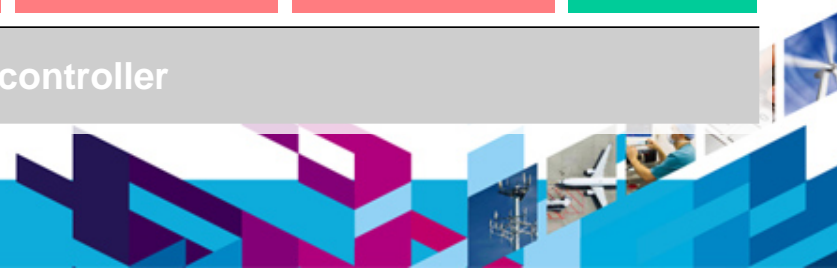


Software Architecture – Basic Software

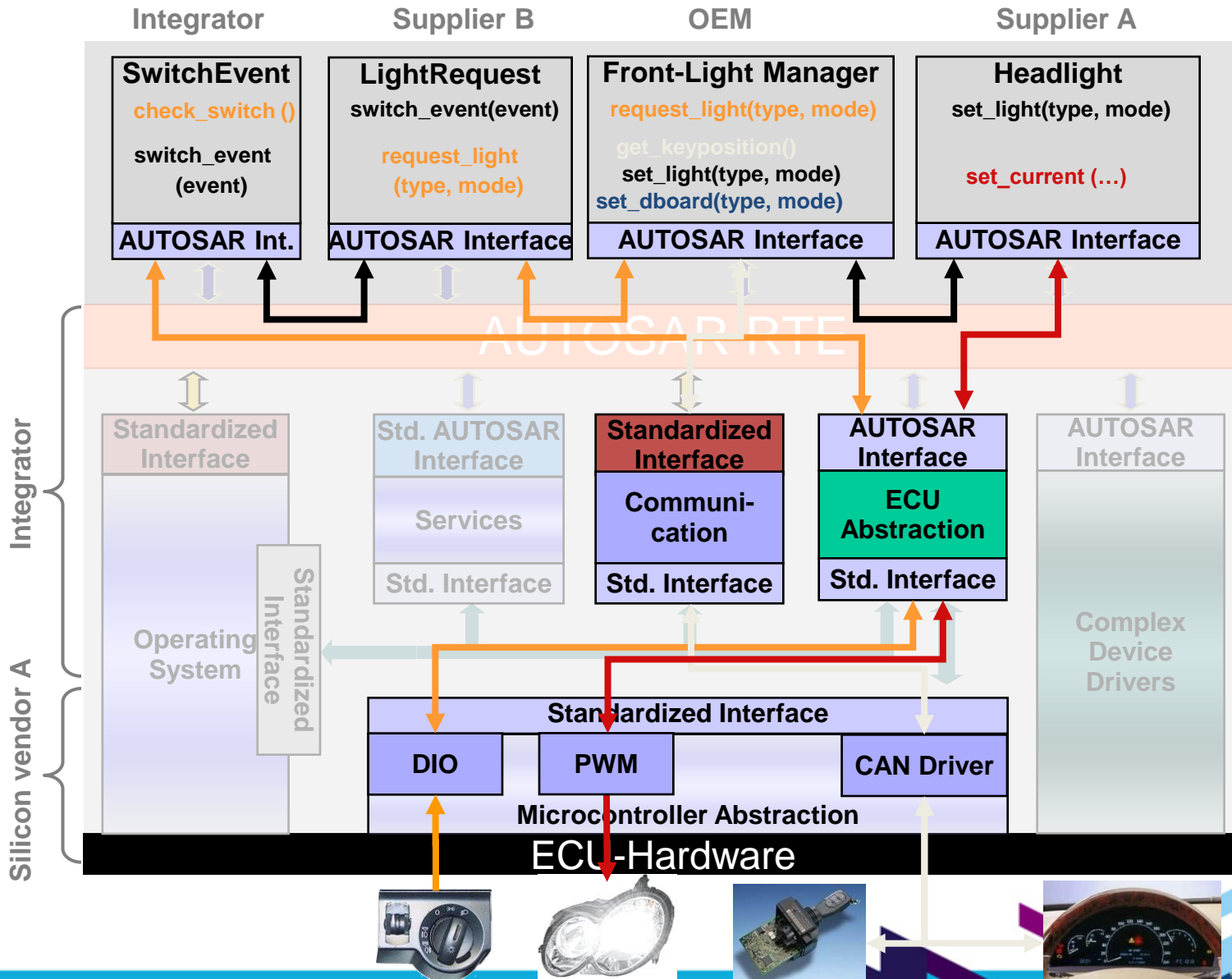
Basic Software 主要包含服务层、驱动层和抽象层



Technological Conception
AUTOSAR ECU Software Architecture



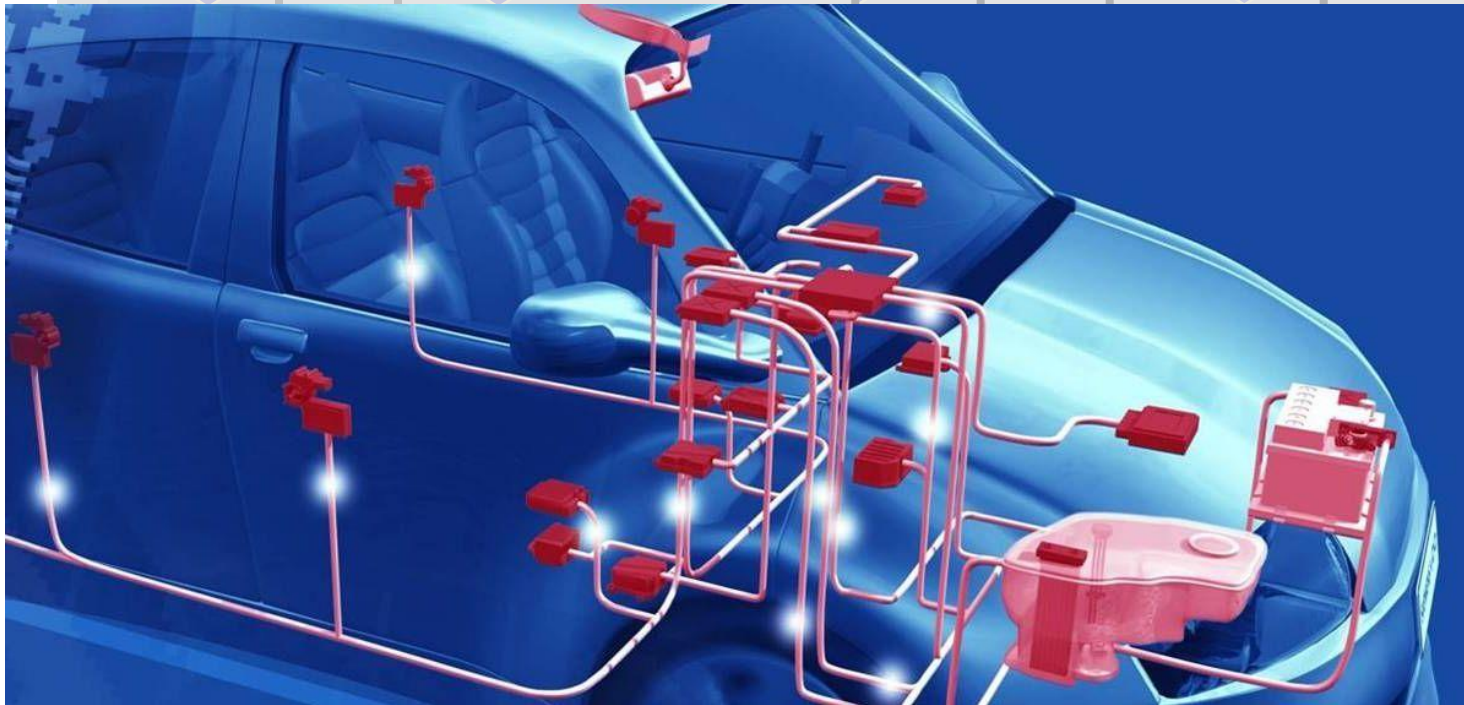
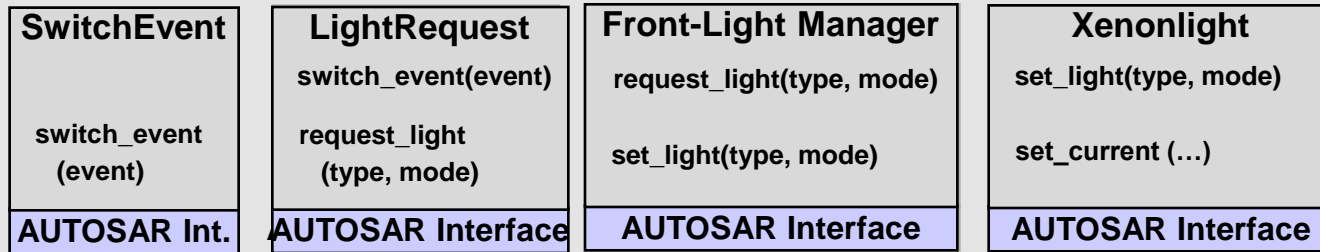
Use Case 'Front Light Management' mapped to AUTOSAR architecture



Technological Conception
AUTOSAR Application Interfaces

Integrator
Silicon vendor A

Distribution on ECUs



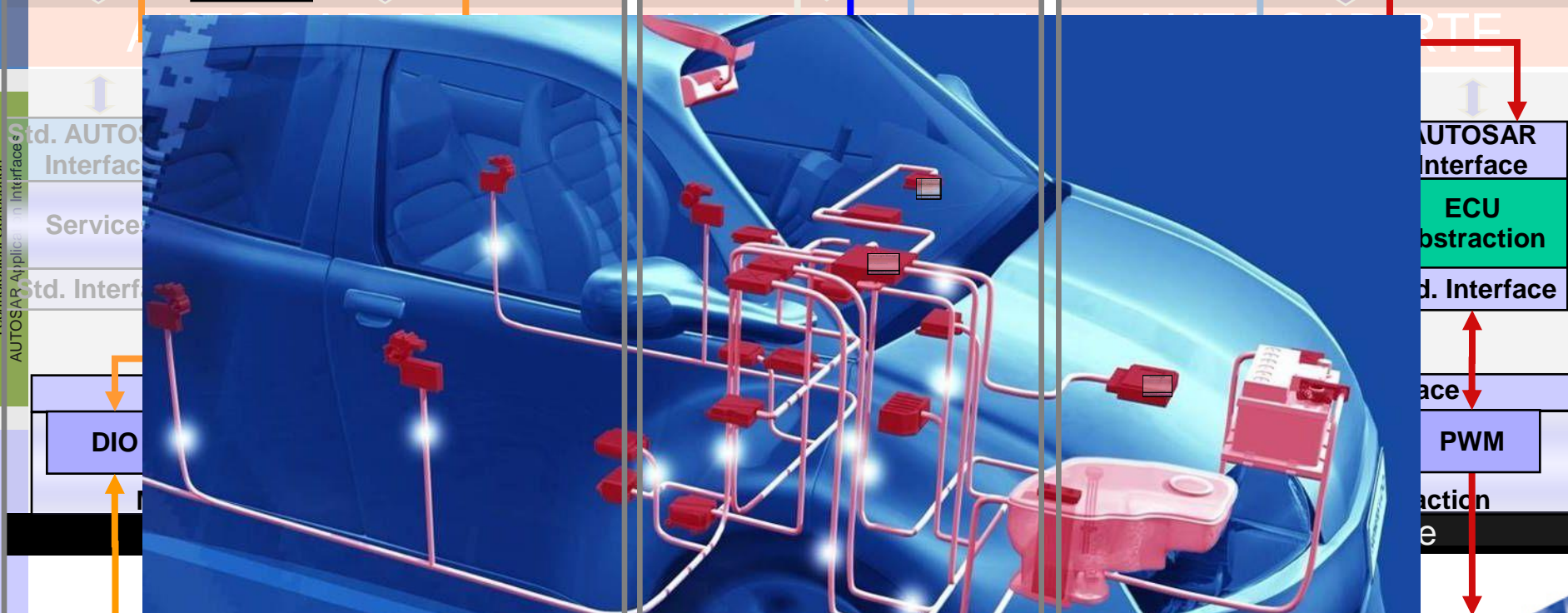
Distribution on ECUs

SwitchEvent
`check_switch ()`
`switch_event (event)`
AUTOSAR Int.

LightRequest
`switch_event(event)`
`request_light (type, mode)`
AUTOSAR Interface

Front-Light Manager
`request_light(type, mode)`
`get_keyposition()`
`set_light(type, mode)`
AUTOSAR Interface

Xenonlight
`set_light(type, mode)`
`set_current (...)`
AUTOSAR Interface



Technological Conception
 AUTOSAR Application Interface
 Std. AUTOSAR Interface
 Service
 Std. Interface

AUTOSAR Interface
 ECU Abstraction
 Std. Interface
 Interface
 PWM
 Action
 e

DIO

CAN Bus

内容安排

- AUTOSAR 的由来
- AUTOSAR技术架构
- AUTOSAR开发流程
- Rhapsody对AUTOSAR的支持
- Rhapsody和Simulink的关系
- 总结

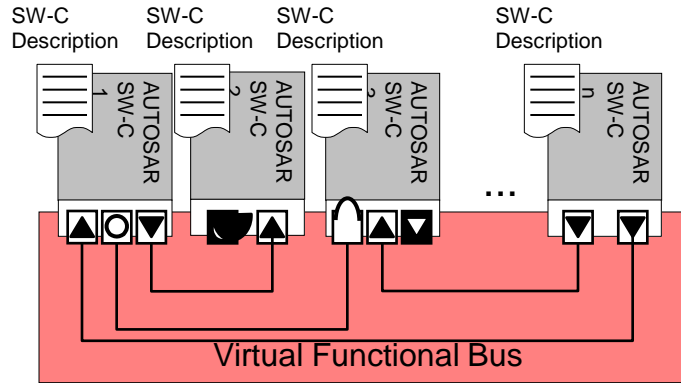


基本 AUTOSAR 开发方法

Virtual Integration (VFB view)

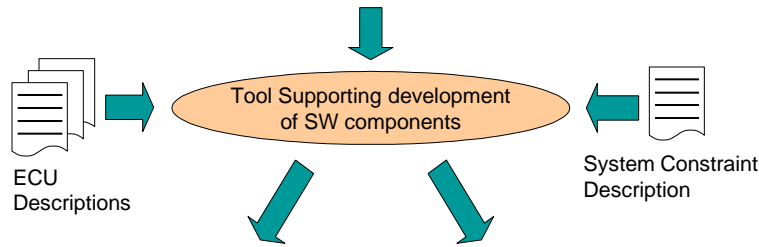
Independent of hardware

Virtual Functional Bus



Introduction of HW Attributes

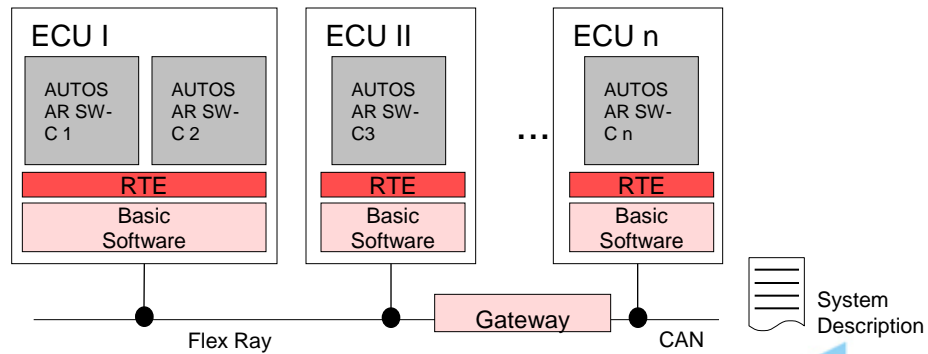
Holistic view of the entire system, both software and hardware



ECU Configuration Mapping

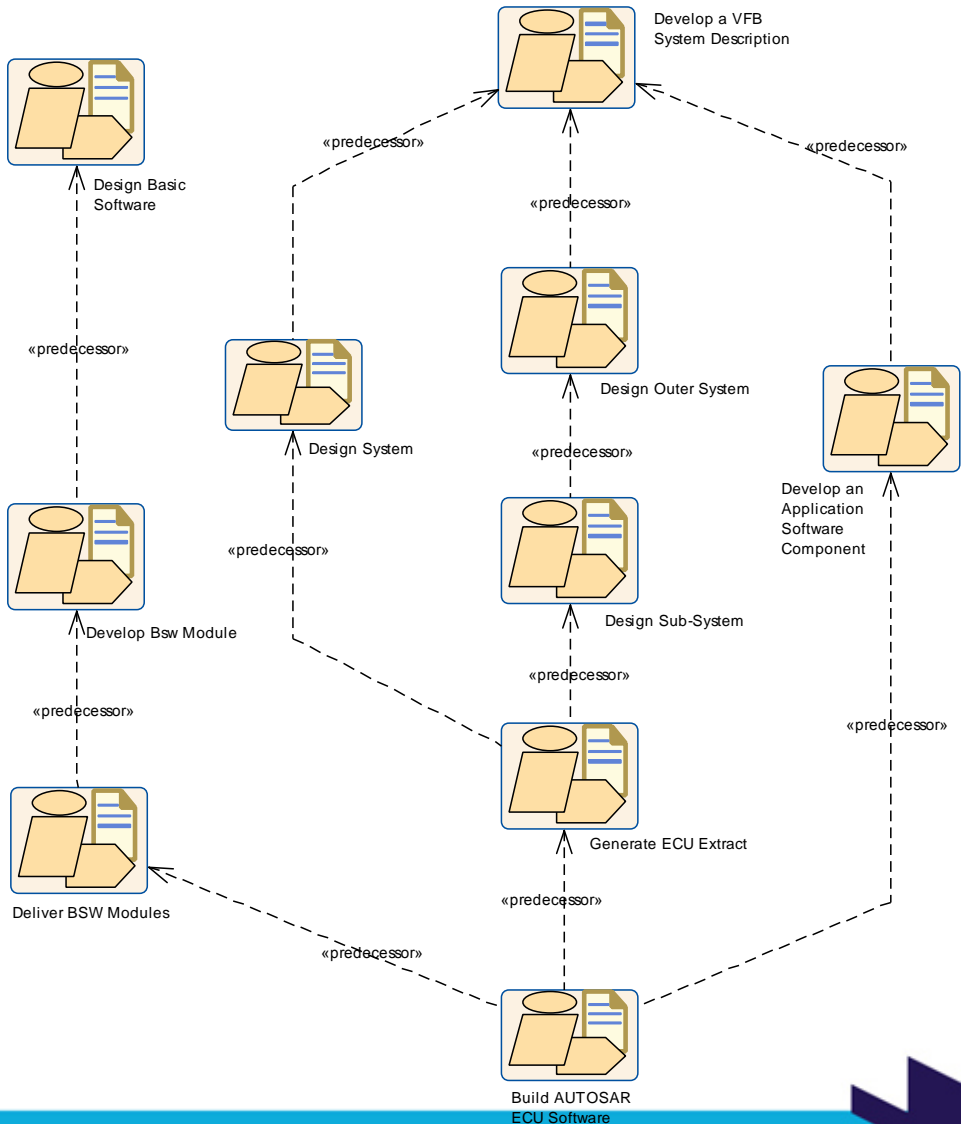
Run-Time Environment

Separation of system into its ECU (plus common infrastructure)

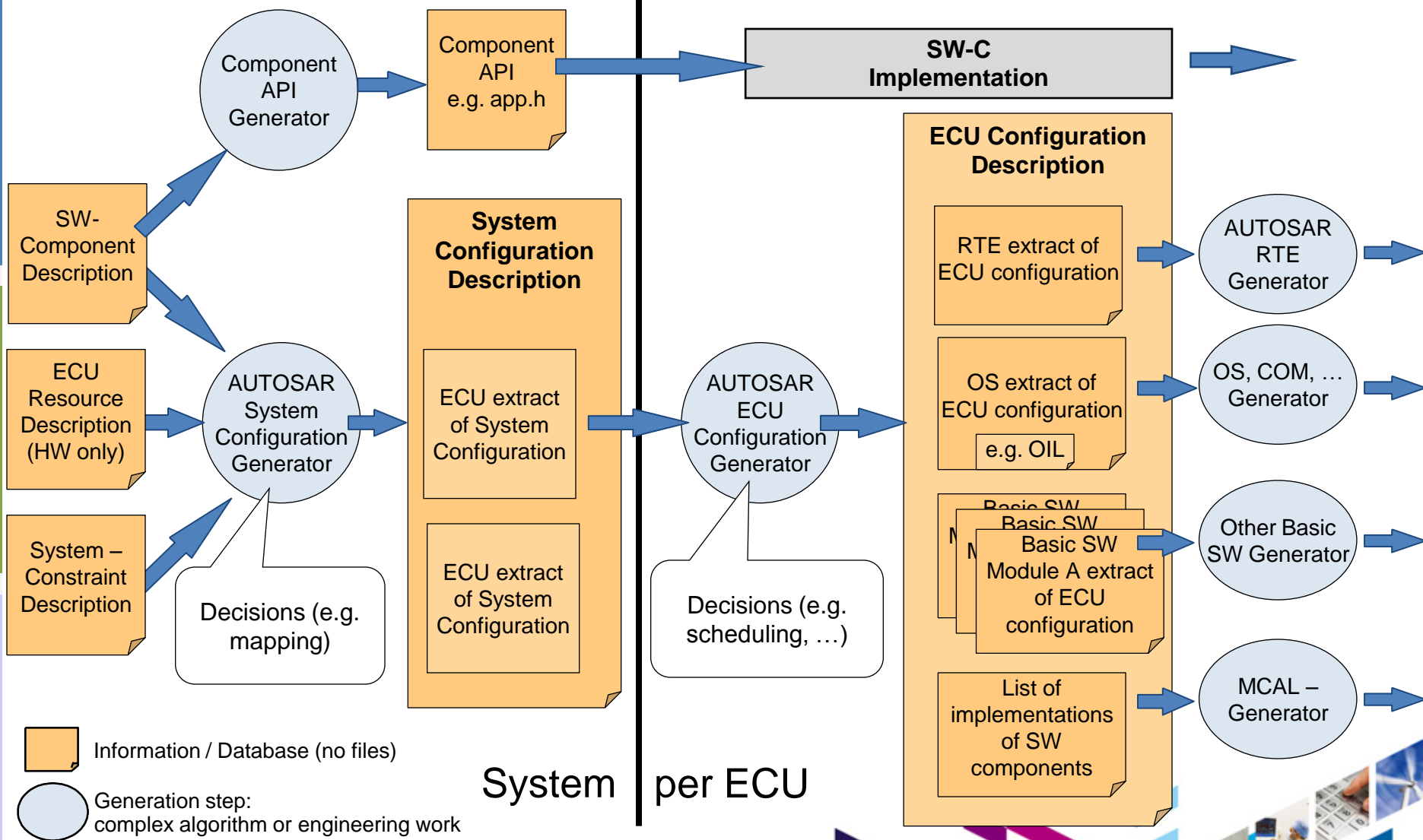


Technical Conception
Technical Scope and Approach

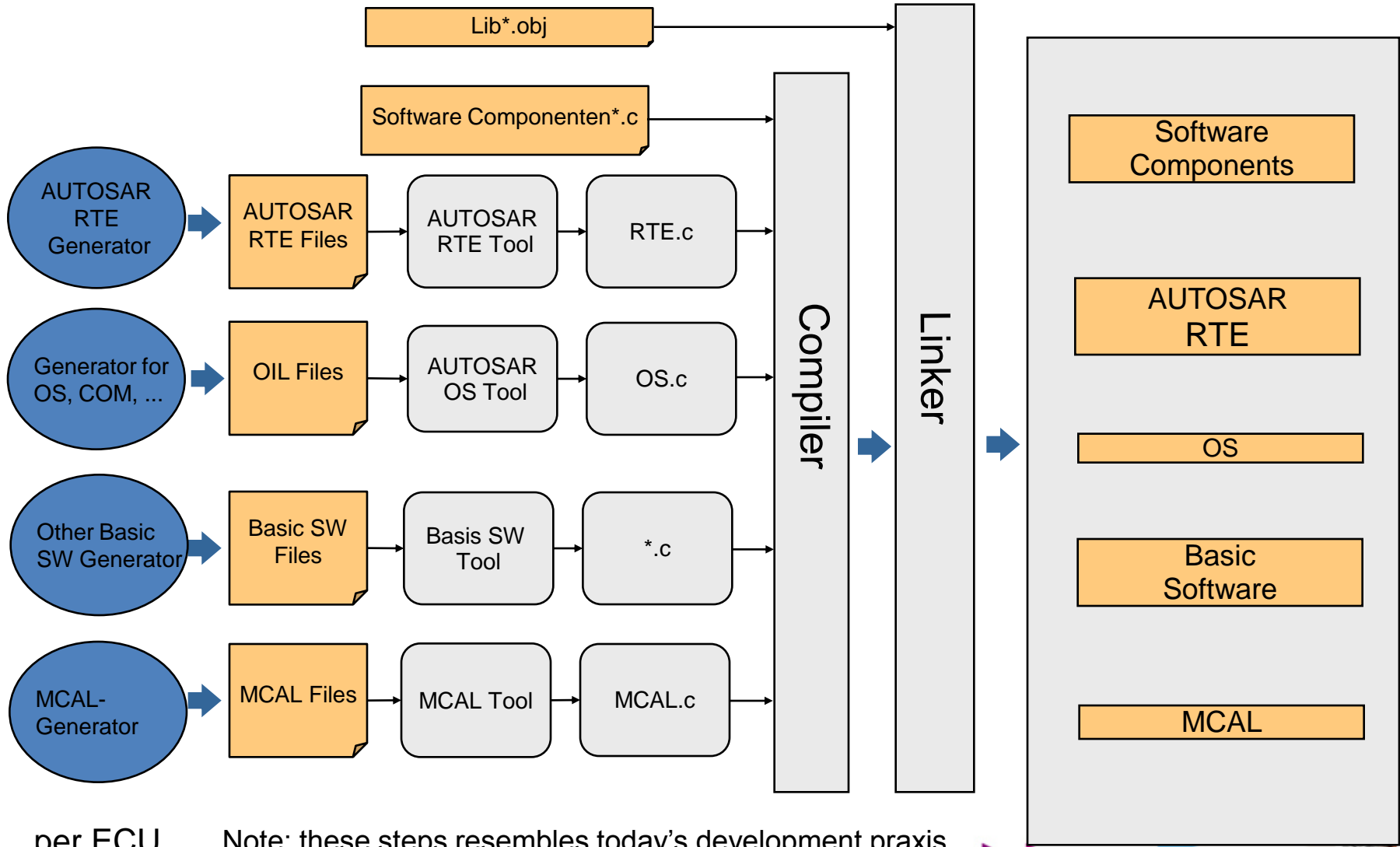
AUTOSAR methodology (published on www.AUTOSAR.org)



AUTOSAR Methodology – Alternative Visualization (1 of 2)



AUTOSAR Methodology – Alternative Visualization (2 of 2)

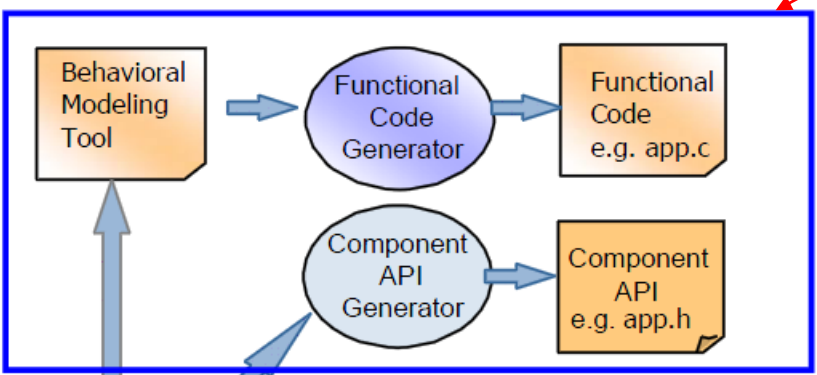


per ECU

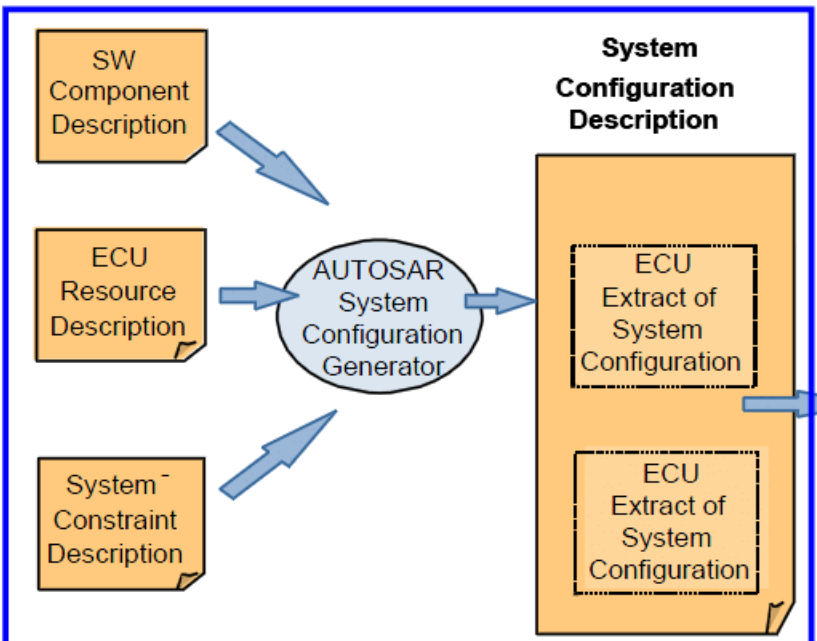
Note: these steps resembles today's development praxis

Tooling for AUTOSAR Tool Clustering

Rational
Rhapsody fits in

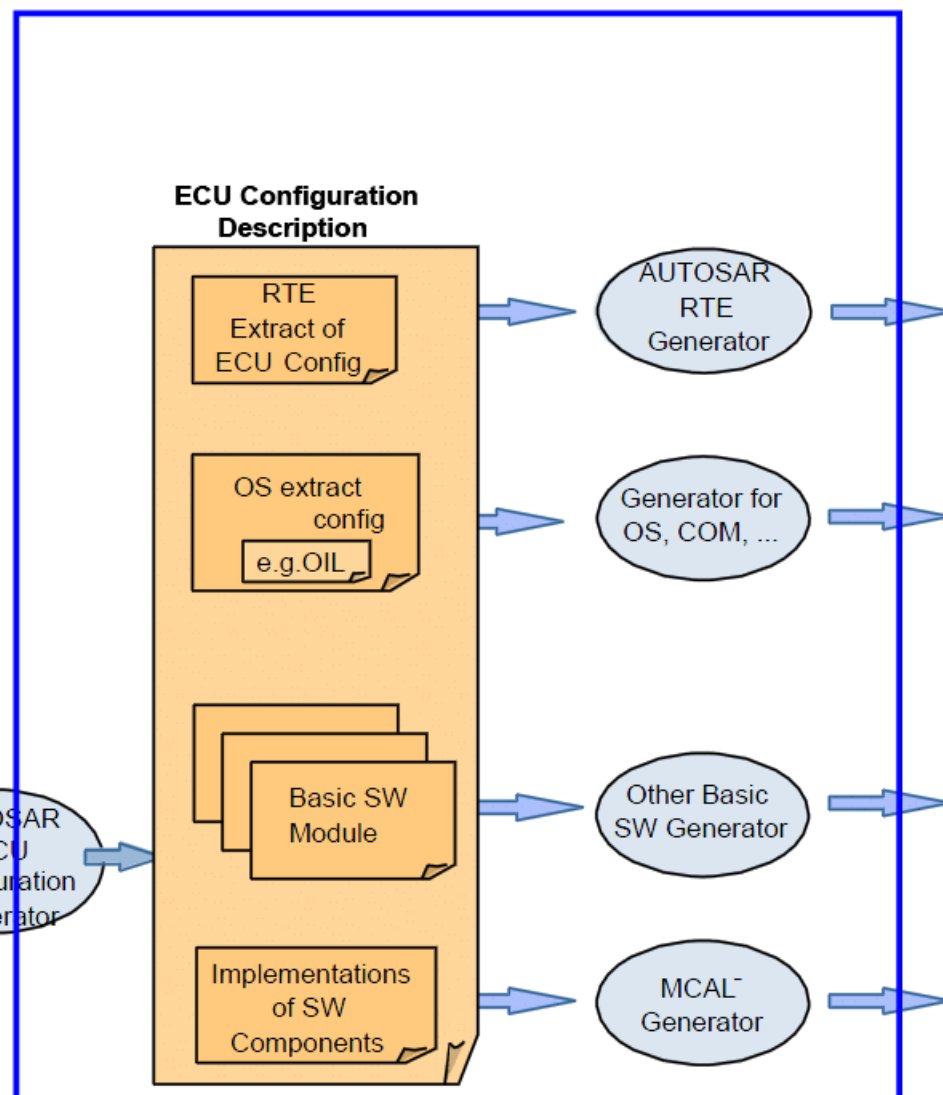


Behavior Modeling



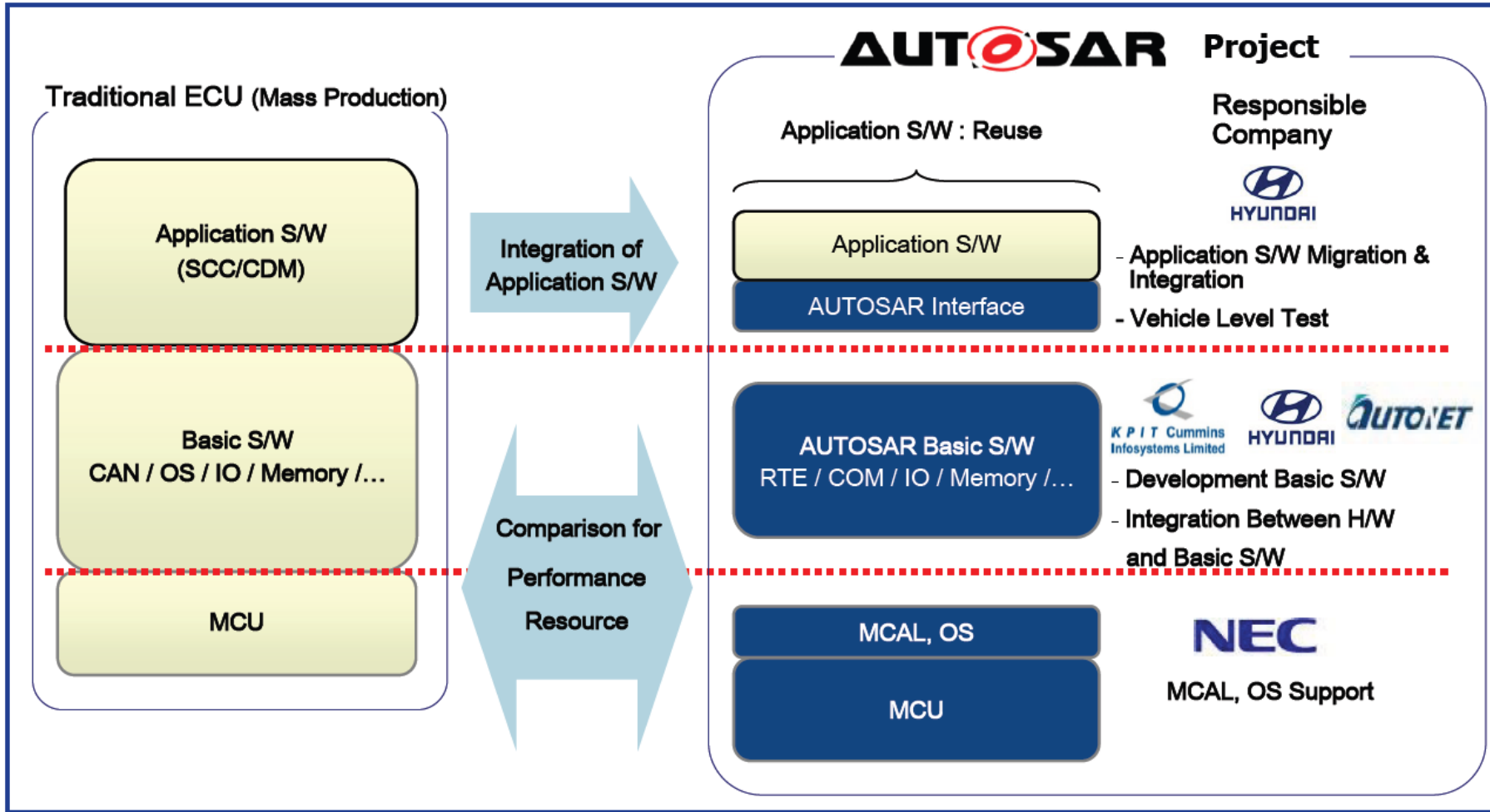
Authoring

ECU mapping



ECU configuration & BSW generation

产业链举例

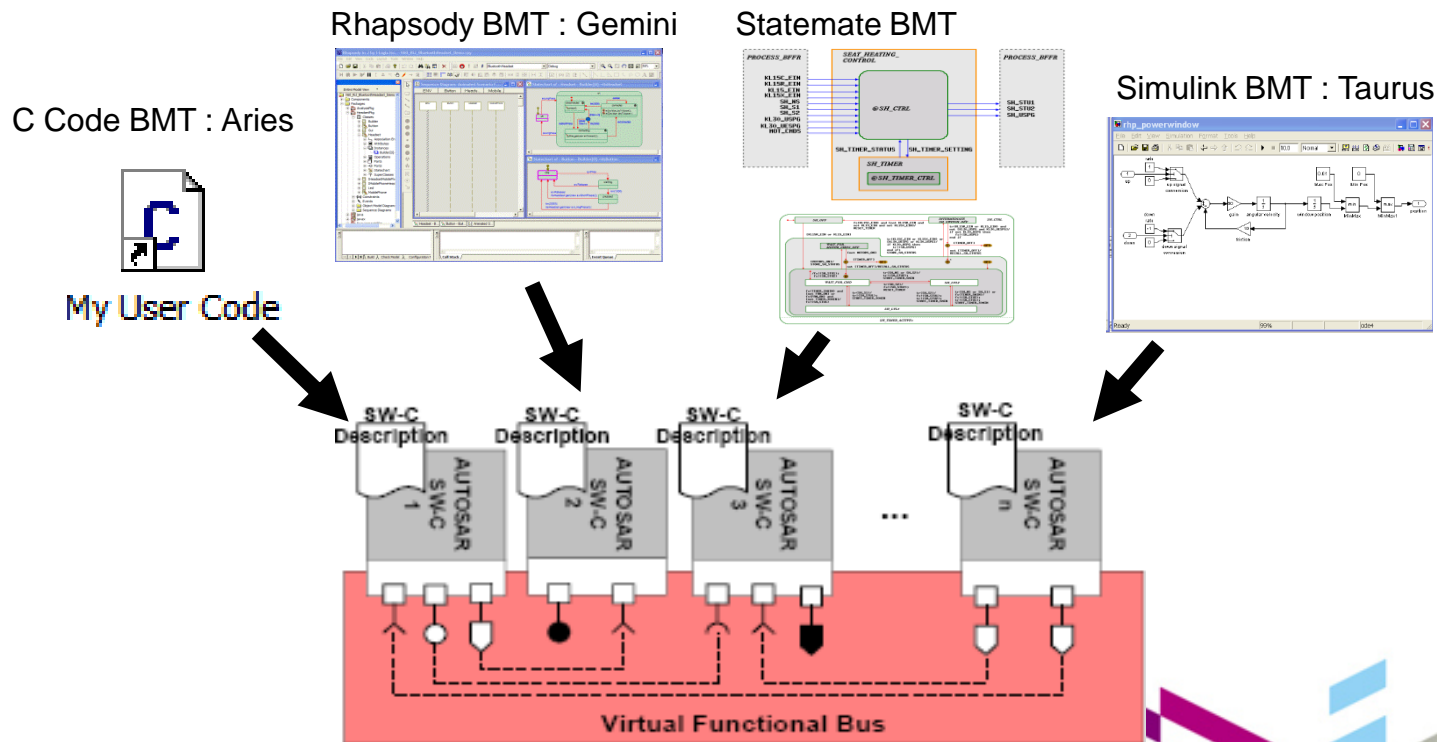


内容安排

- AUTOSAR 的由来
- AUTOSAR技术架构
- AUTOSAR开发流程
- Rhapsody对AUTOSAR的支持
- Rhapsody和Simulink的关系
- 总结

Rhapsody support for AUTOSAR

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



Rhapsody Systems Modeling : Award-Winning Solution!

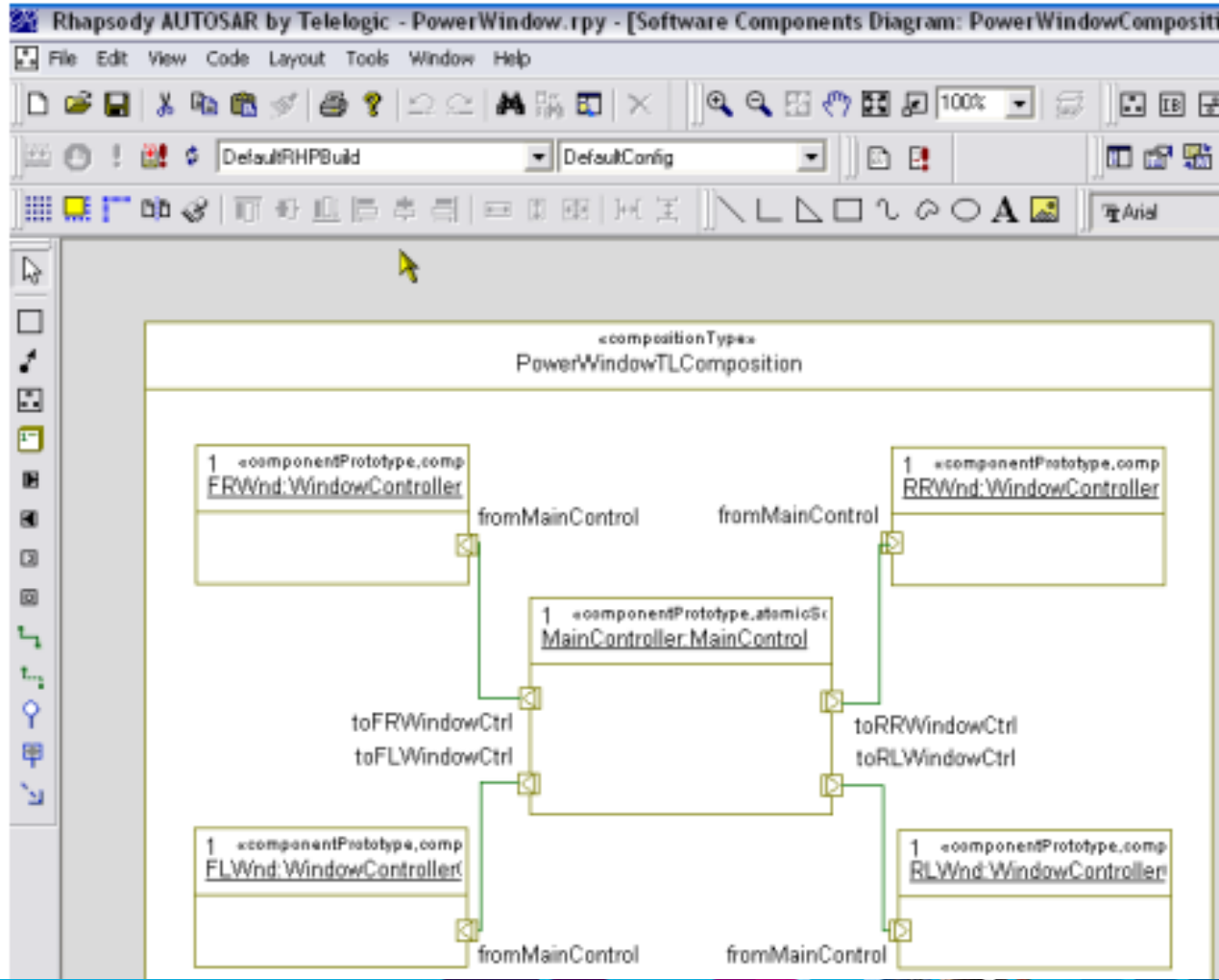
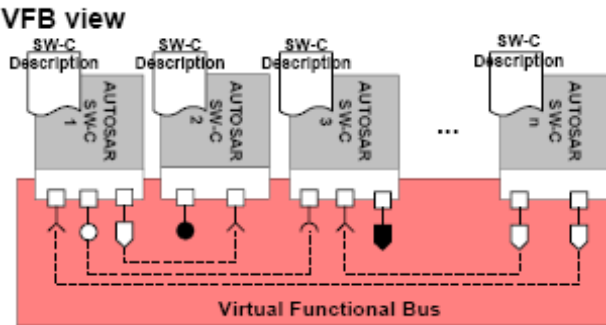
Rhapsody support for AUTOSAR

- 主要作为 modeling tool, 支持 versions 2.1.4, 3.1.0, 4.0.1
- 通过 AUTOSAR profile, 描述下列 diagrams
 - Software Component Diagram
 - Internal Behavior Diagram
 - ECU Diagram
 - Topology Diagram
 - Implementation Diagram
 - Systems Diagram
- XML Export
- Code-gen (AR3x_BMT profile), 产生的 code 调用 RTE API
- RTE API contract generator



Software Component Diagram

- Defines the software architecture of the functional system



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

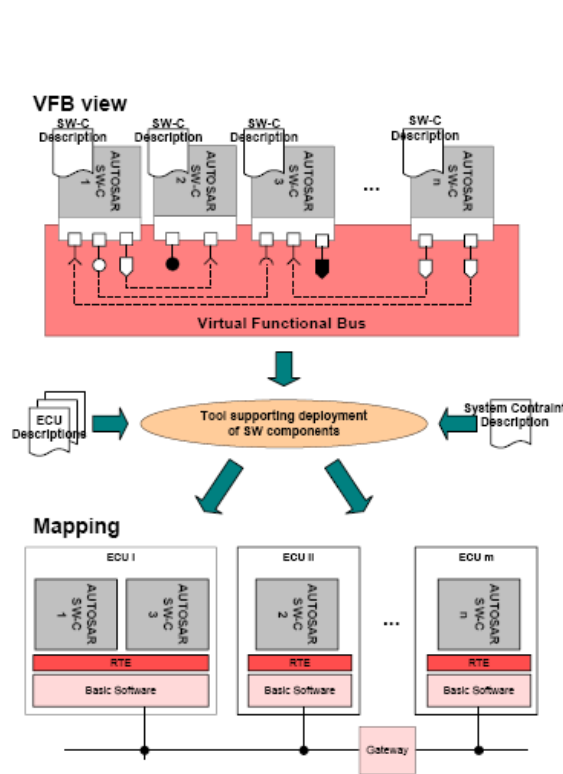
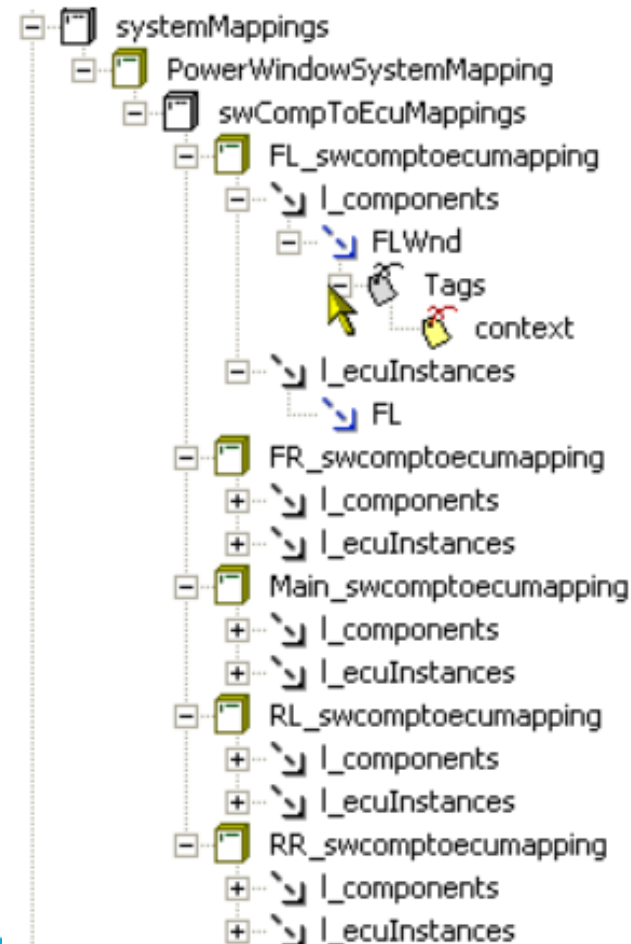
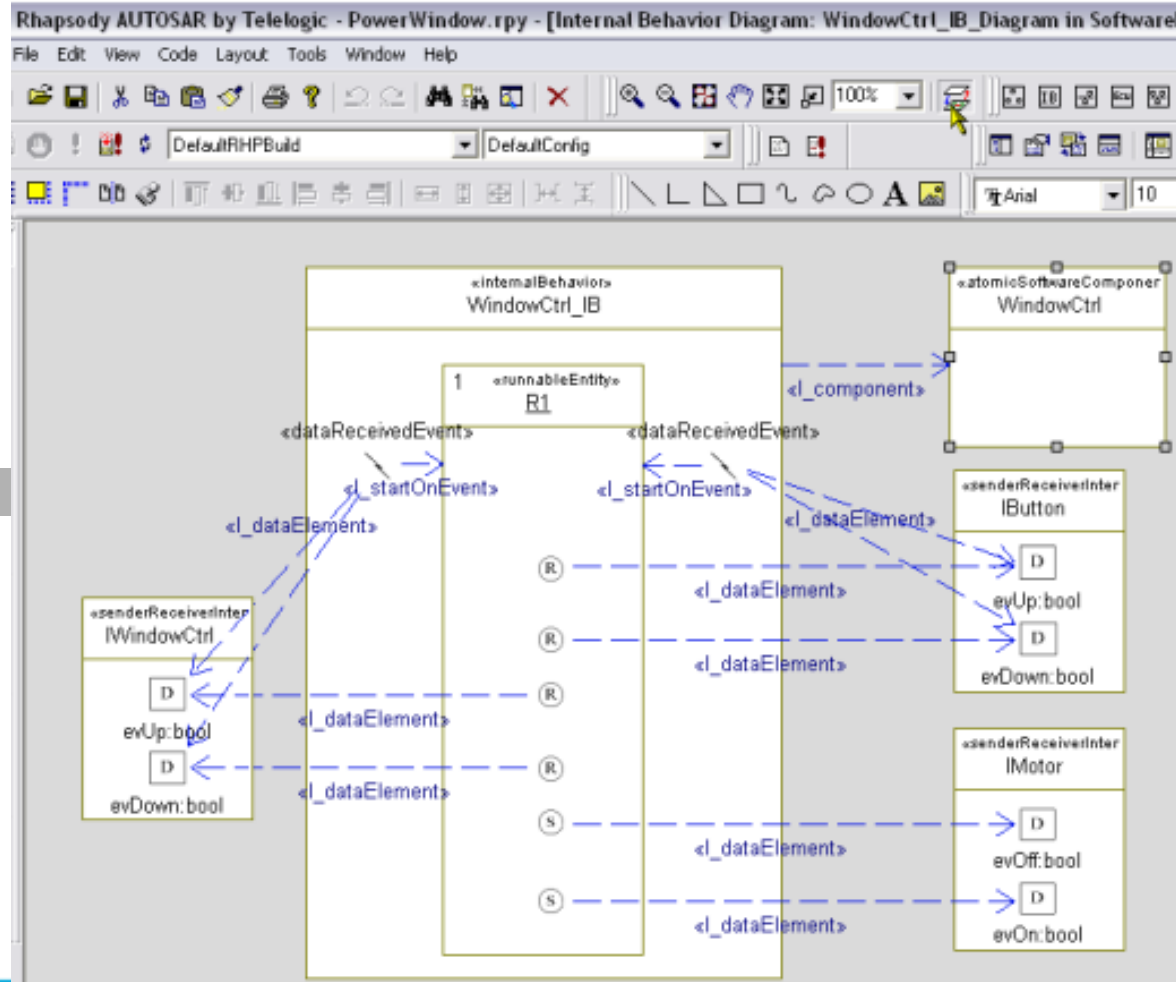
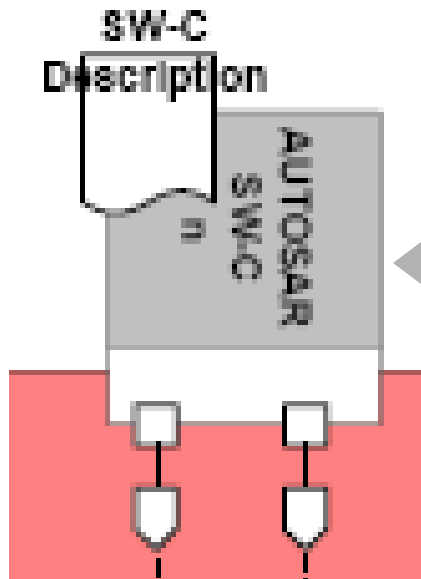


Figure 2: Basic AUTOSAR approach.

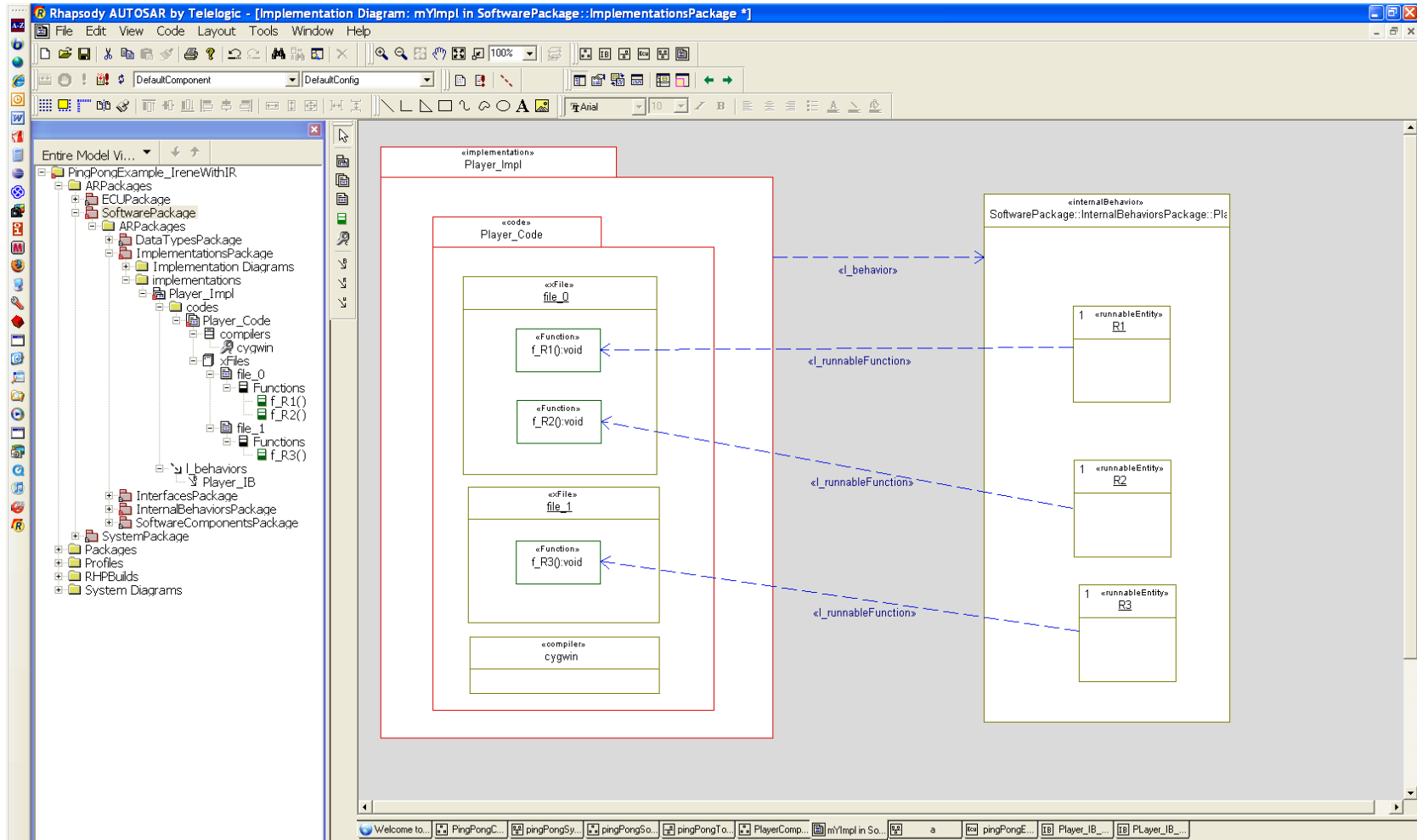


Internal Behavior Diagram

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



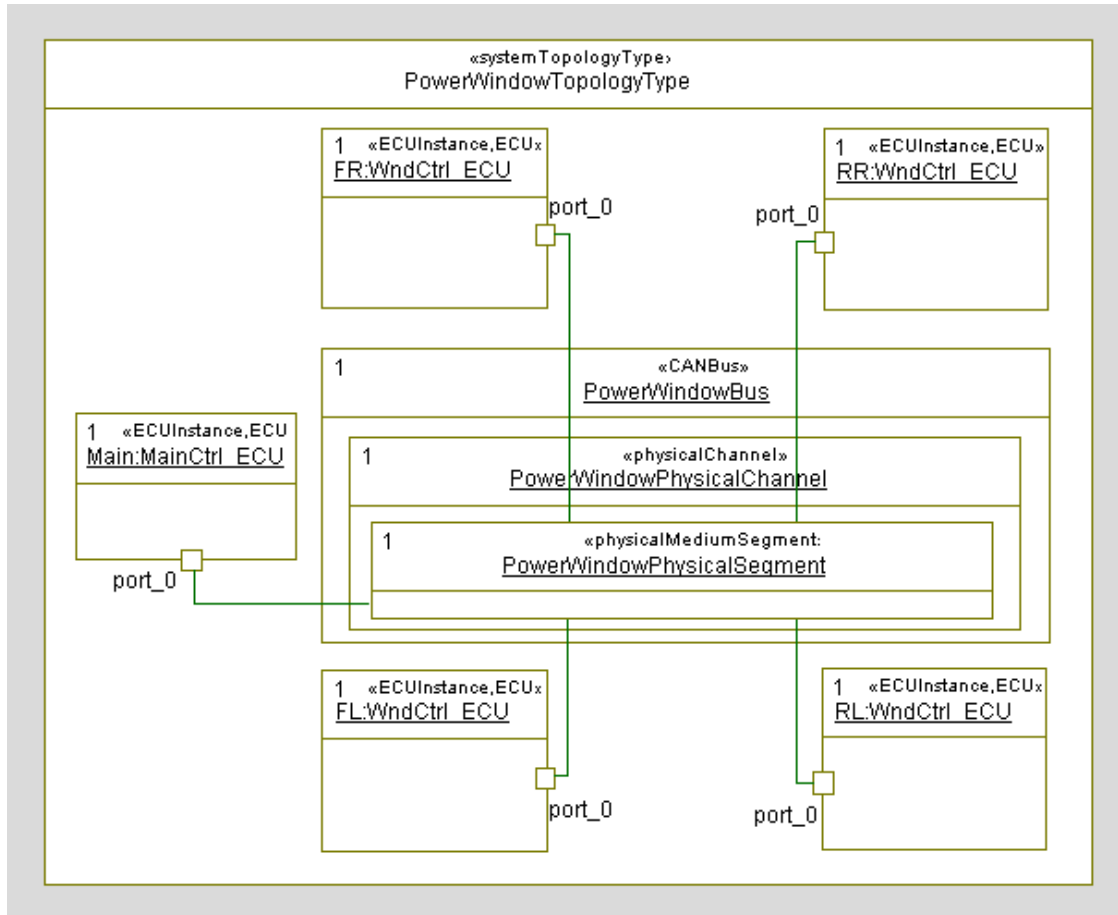
Implementation Diagram



For Help, press F1

GE MODE | Mon, 23, Apr 2007 12:55 PM

Topology Diagram



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

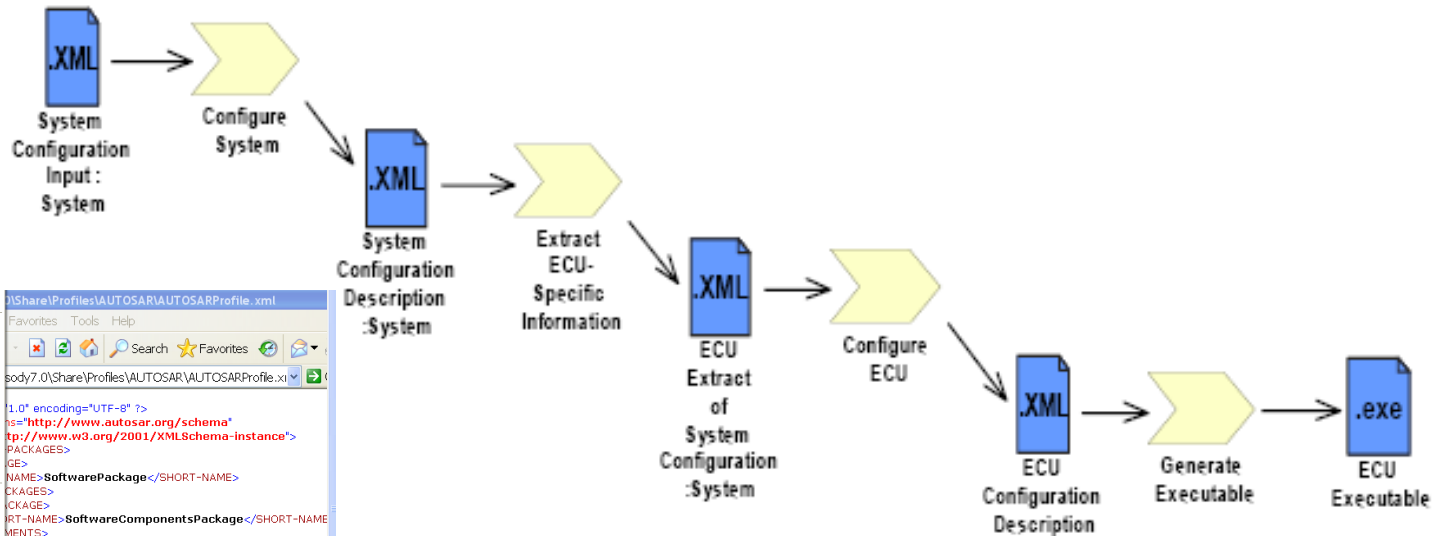
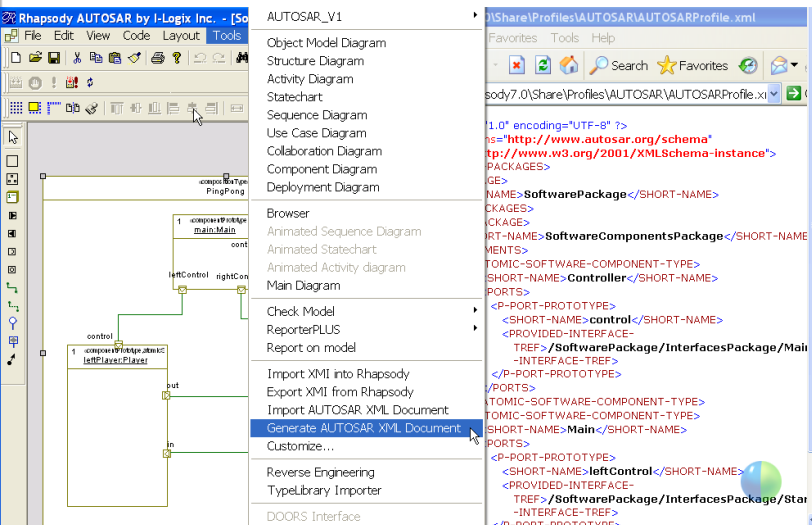


Figure 1: Overview AUTOSAR methodology



内容安排

- AUTOSAR 的由来
- AUTOSAR技术架构
- AUTOSAR开发流程
- Rhapsody对AUTOSAR的支持
- Rhapsody和Simulink的关系
- 总结

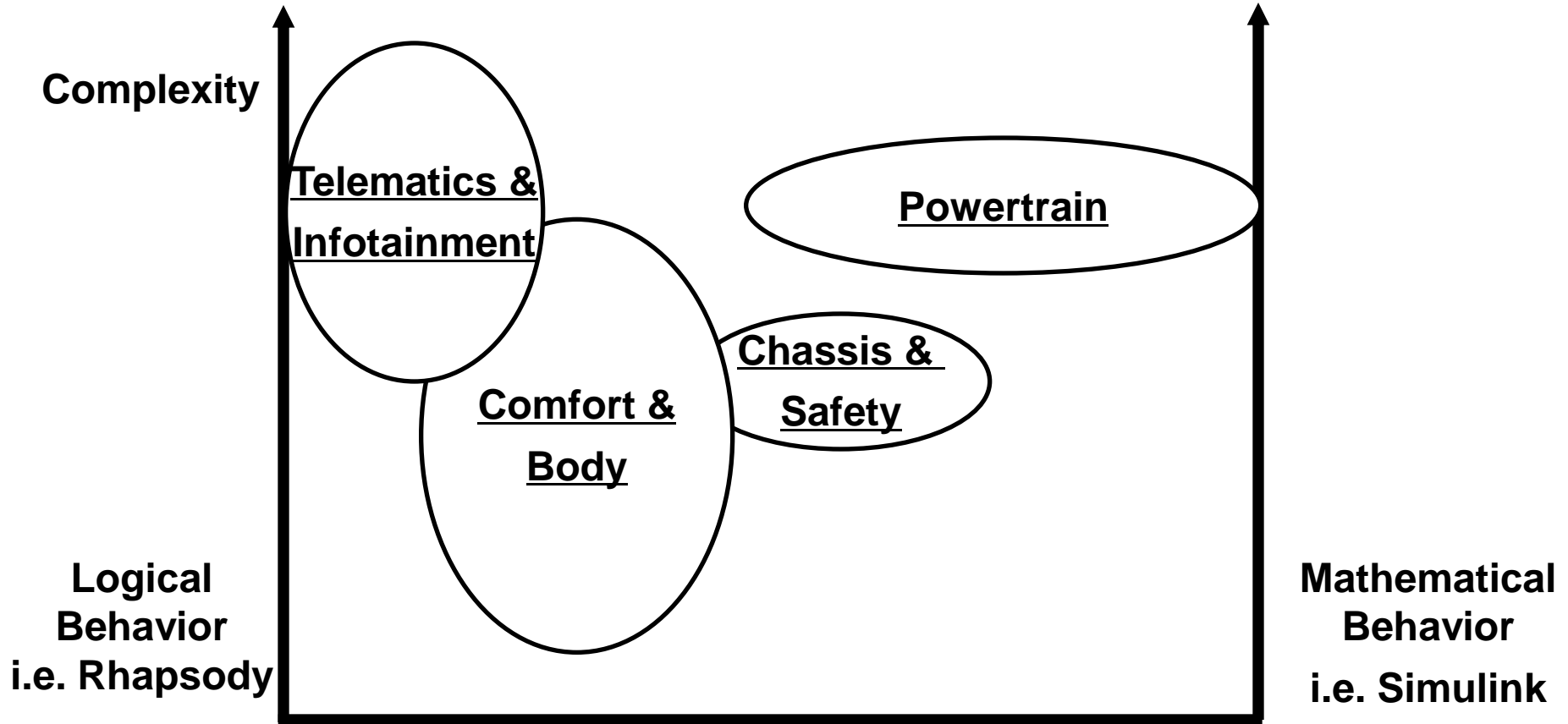


Rhapsody Vs Simulink

- Rhapsody
 - 架构和离散实时行为（基于状态机）--强调事件触发 (state machine modeling)
 - 对需求的建模 (functional analysis)
 - 针对嵌入式应用软件或系统 (architecture and behavior modeling, UML)
- Simulink
 - 控制行为建模—强调输入输出和控制逻辑（control equation modeling）
 - 基于数学的算法建模 (algorithm modeling)
 - 汽车物理特性的建模（plant modeling）



Rhapsody 更适合高复杂性（架构和行为）的软件部分



用Rhapsody 和Simulink联合设计一个简单的汽车巡航控制系统（Cruise Control System）

- 巡航控制系统由两部分组成 — 硬件（物理系统）和软件（逻辑系统）
 - 物理系统能够将 desiredSpeed（设定速度） 转化为actualSpeed（实际速度）
 - 逻辑系统能够监控actualSpeed， 设定desiredSpeed， 并且能通过驾驶员命令将desiredSpeed 输入到物理系统
- Simulink模型模拟硬件的主要部分 — 汽车
 - Simulink 可以对汽车物理特性和系统方程进行建模， 系统方程决定了汽车对驾驶员命令的相应
- 我们的任务是设计软件部分， 但是我们也需要硬件部分
 - 于是， 我们用Simulink模型来代表汽车的硬件部分,虽然它也是由软件构成的
- Rhapsody模型代表（实现）了汽车和驾驶员的接口部分， 以及系统功能的软件部分



Simulink – 物理设定

- § 假定汽车正在验证平滑的路面行驶。来自汽车引擎的 **Thrust (推力)** 从汽车的传动系统被输出，用来推动汽车行驶，并且对抗空气阻力和其它 **opposing forces (阻力)**



- ▶ 推力和汽车的功率成正比，和速度成反比

$$\text{Power} = \text{Force} \times \text{distance}$$

$u = pwr / vt$, 这里 u 是 thrust (N), pwr 是 power (W), vt 是 distance (M)

$$\textcircled{1} u \approx \frac{pwr}{v}$$

- ▶ 拖拽力, 正比于 v^2 , 在正常行驶情况下是最大的阻力

$$\textcircled{2} F_{drag} = -\frac{1}{2} C_d A \rho_{air} v^2 = b$$

- § 系统的数学模型用如下方程来表示:

$$\textcircled{3} \frac{\partial v}{\partial t} = \frac{u + b}{mass}$$

$$\textcircled{4} \int \frac{\partial v}{\partial t} = v$$

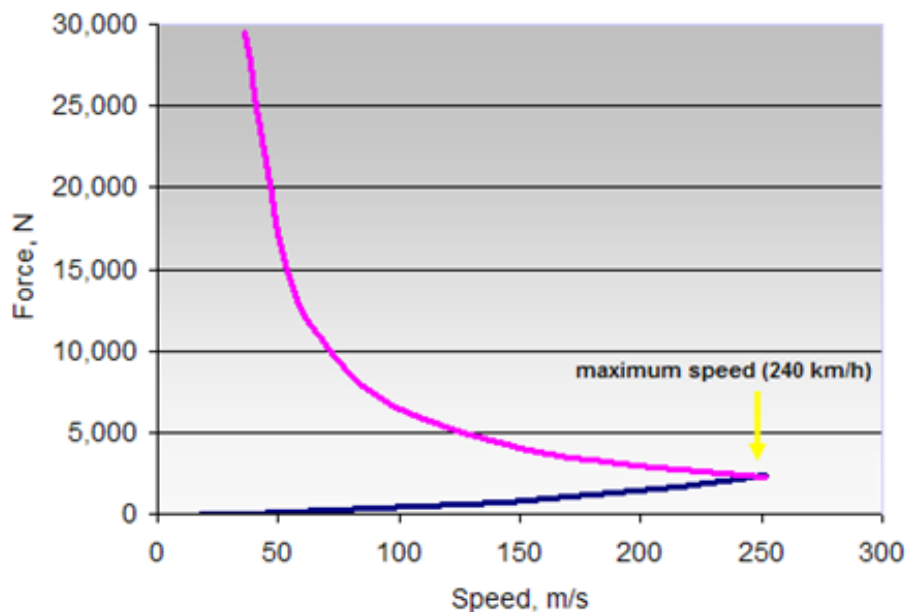


汽车的动态数据

§ 我们用一辆Audi A4 2.0T的物理数据做为我们做数学计算的基础数据

- ▶ 汽车达到最大速度 当 F_{drag} 等于推力

$$v_{max} = 240 \text{ km/h}$$



Data for Prototypical Vehicle	
power, bhp	200
power, W	149,140
power at wheels, W	147,649
mass, kg	1705
drag coef C_d	0.44
frontal area A , m^2	1.8
density of air, kg/m^3	1.2041



建立Simulink模型 (1)

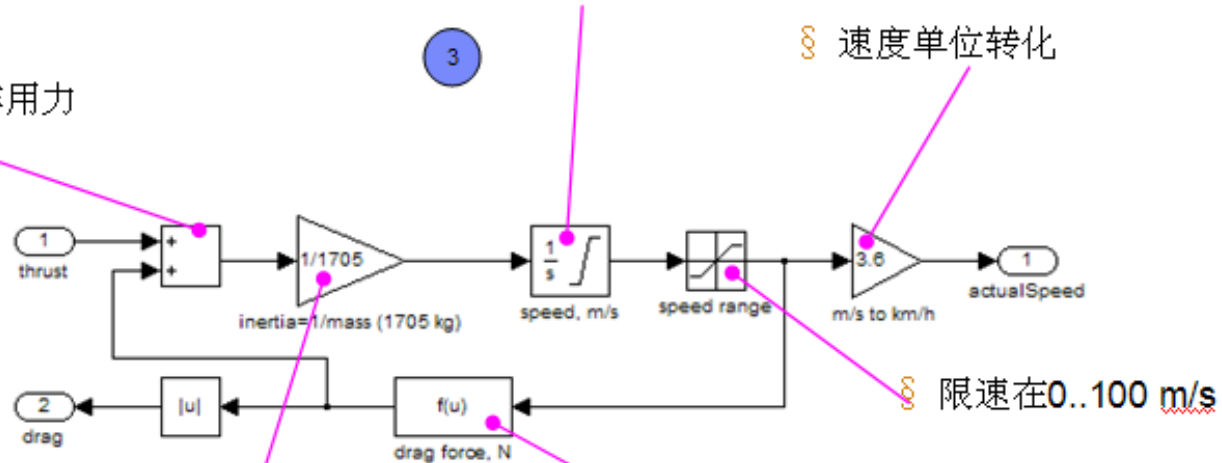
§ EnvironmentReaction 模型

▶ 作用于汽车的力决定了汽车的速度

§ 积分器用来对加速度求积分算出速度

§ 速度单位转化

§ 加法器相加所有作用力



拖拽力 2 用方程计算

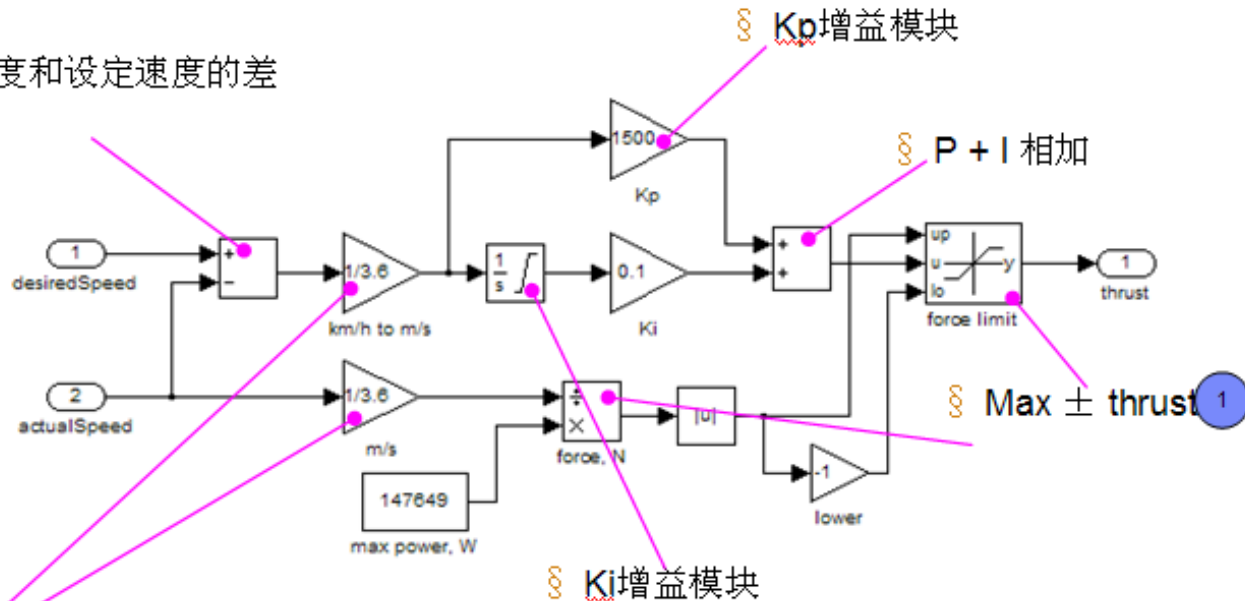
§ 通过作用力和质量的相除得到加速度

建立 Simulink 模型(2)

§ PropFeedback 模型

- ▶ PI控制器 根据实际速度和设定速度的差来设定车辆加速的推力，

§ 计算实际速度和设定速度的差

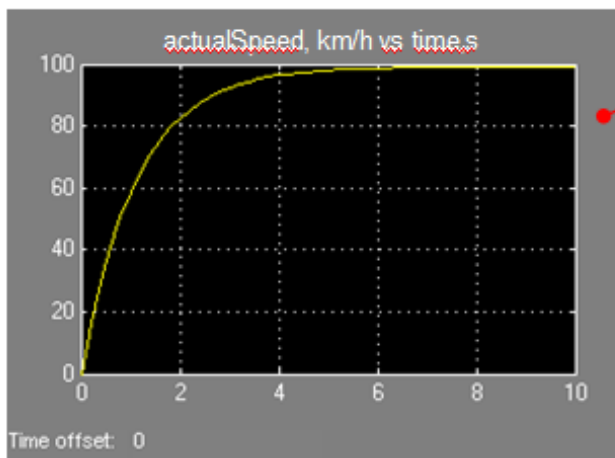
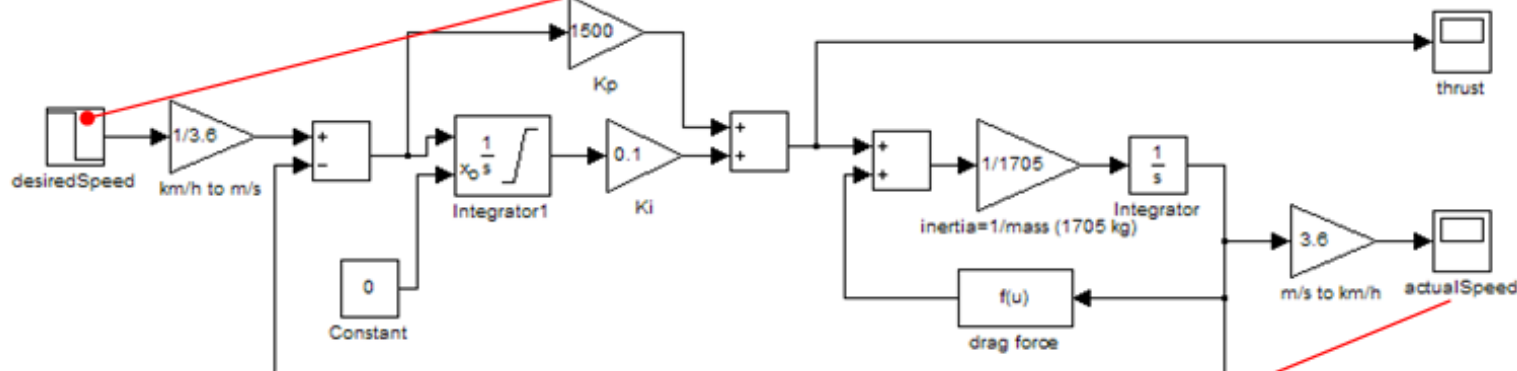


§ 速度单位转化

- ▶ K_p 和 K_i 需要被精确校准以符合设计标准 Rise time < 7 s, Overshoot < 10%, Steady state error < 2%

闭环响应

§ 通过合并 2个Simulink模型，速度从 0 到 100 km/h的闭环响应过程可以被确定和调优

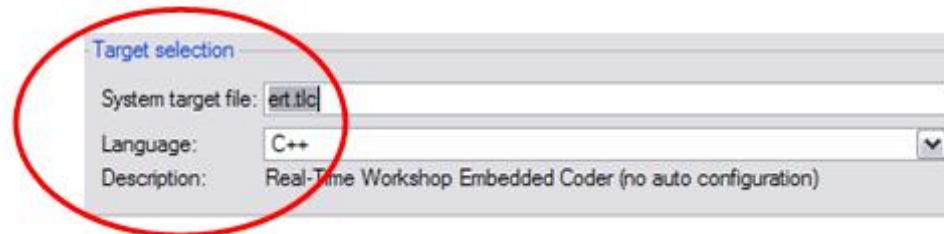
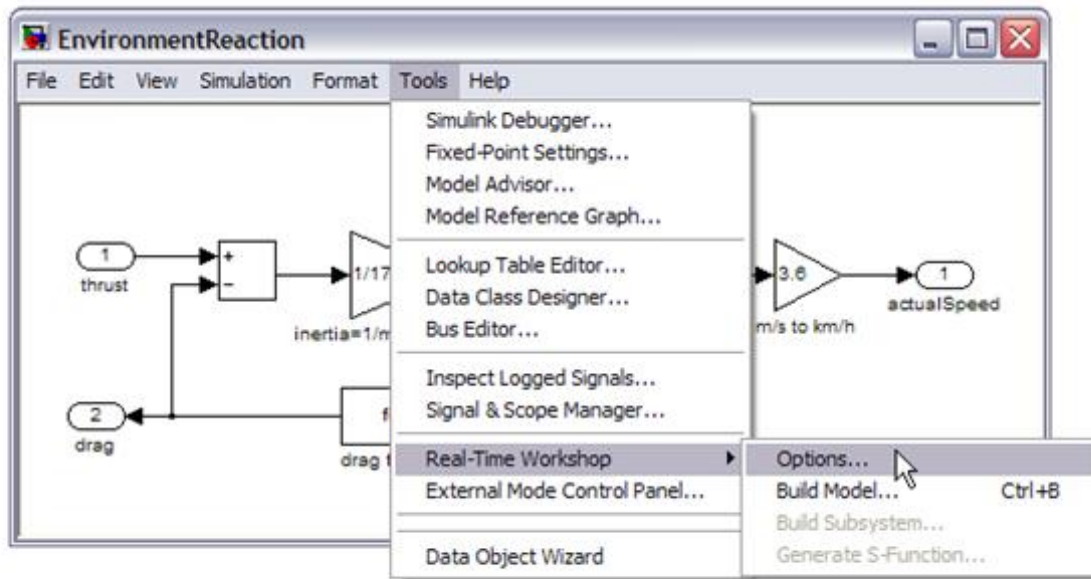


结果：
0 到 100 km/h 加速只用了 6 s

This meets the design criteria

从 Simulink 产生 C++ 代码

§ MatLab Real-Time Workshop

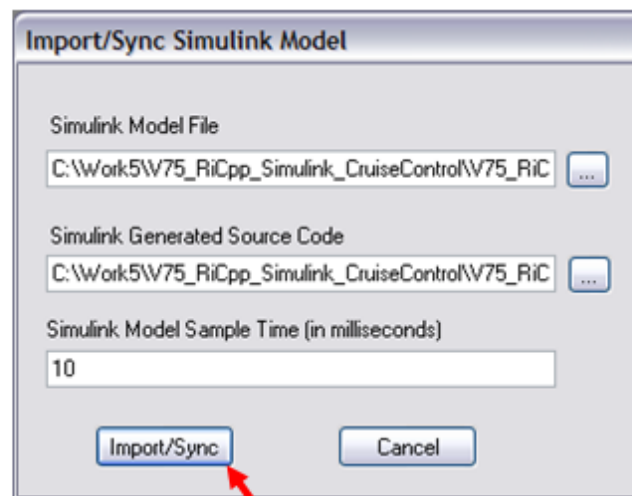
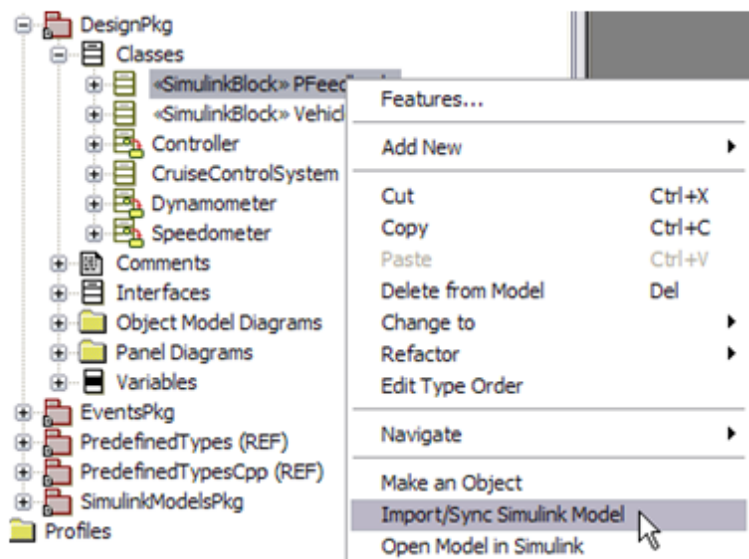


导入 Simulink Code 到 Rhapsody

§ 加一个 class 到 Rhapsody 模型 来表示每个 Simulink 模型

▶ Stereotype the class/block as a «SimulinkBlock»

§ 导入 Simulink code 到 class

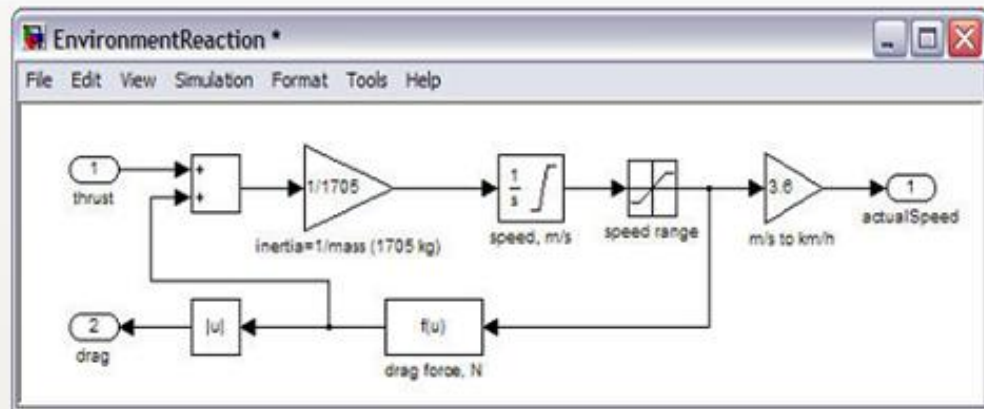


分别添加模型和代码信息, 单击 Import/Sync

SimulinkBlock的变化

§ 通过成功的 Import/Sync, flowPorts 会被自动添加到模型中去来表示Simulink的输入和输出

- ▶ flowPort的类型是real_T, 用来和Simulink的数据匹配
- ▶ flowPort的方向也自动做调整
- ▶ Operation也被自动添加到SimulinkBlock中

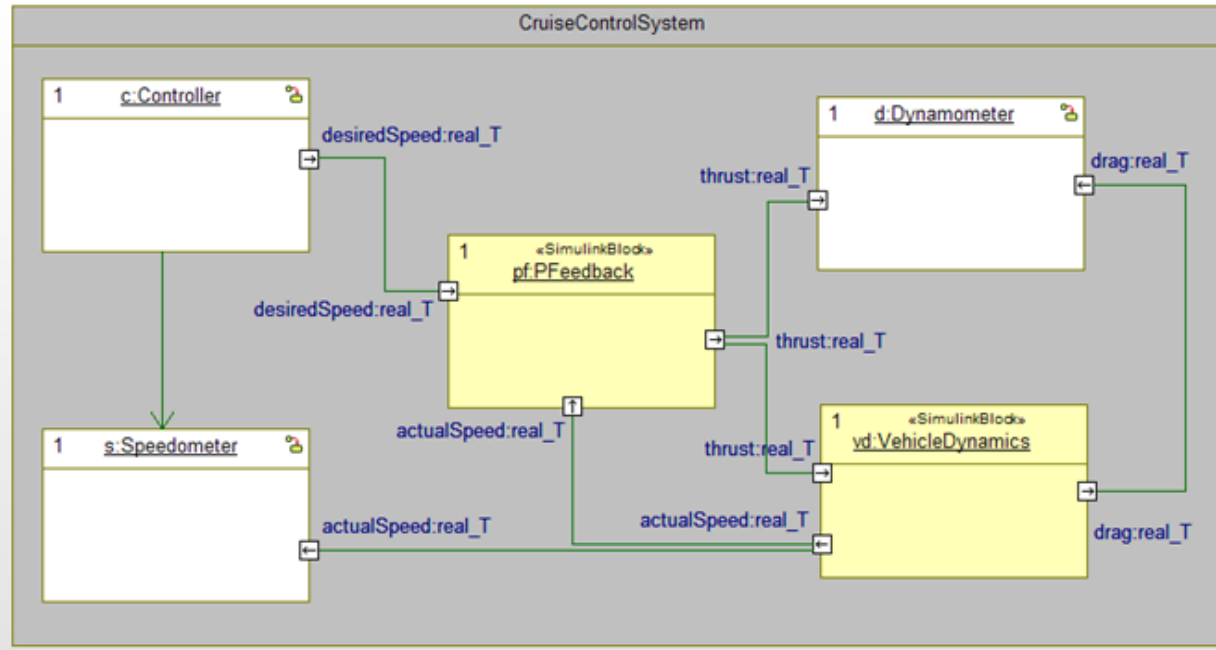
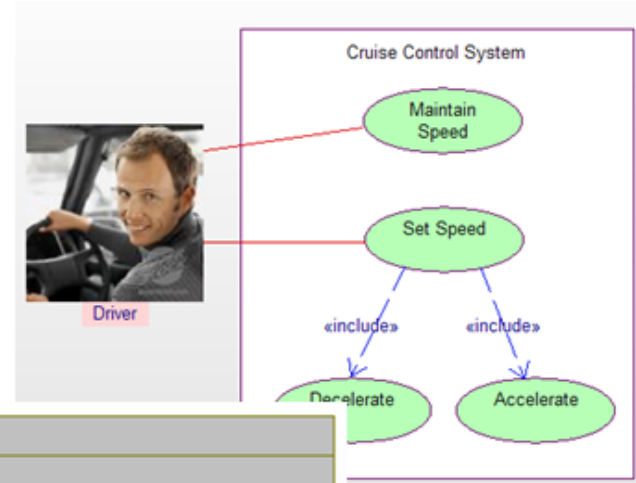


Rhapsody 模型 (1)

Rhapsody 模型 (1)

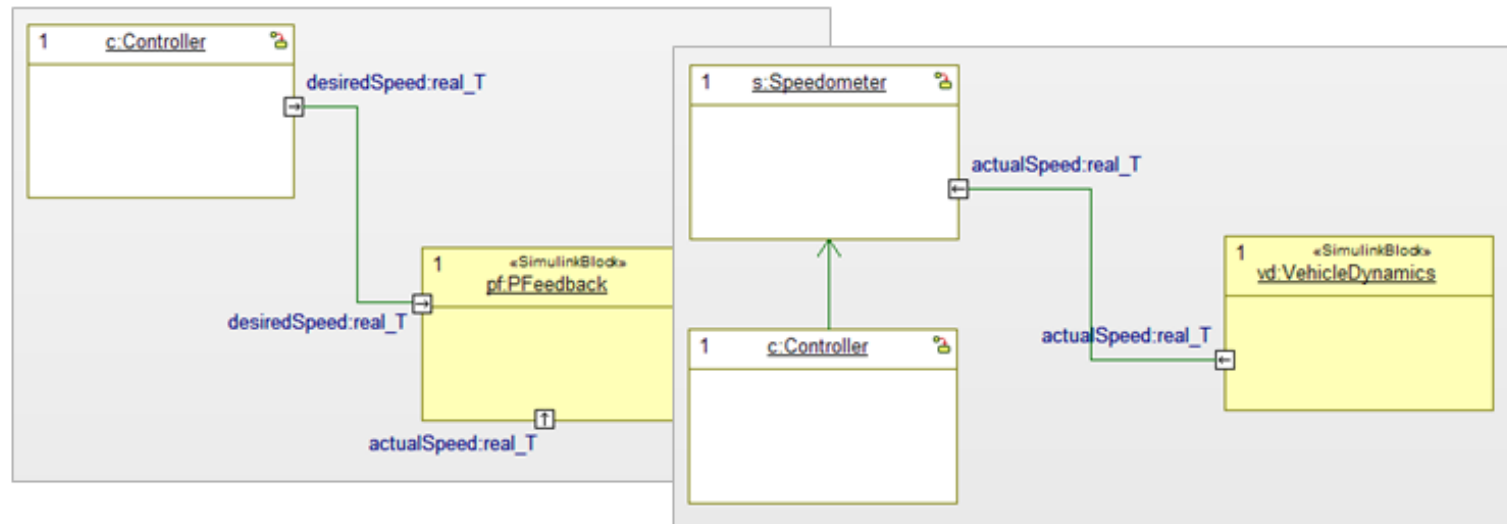
§ 系统设计

- ▶ **Controller** 负责实现巡航系统的功能
- ▶ **Speedometer** 和 **Dynamometer** 是感应器
- ▶ **SimulinkBlocks** 表示汽车物理模型

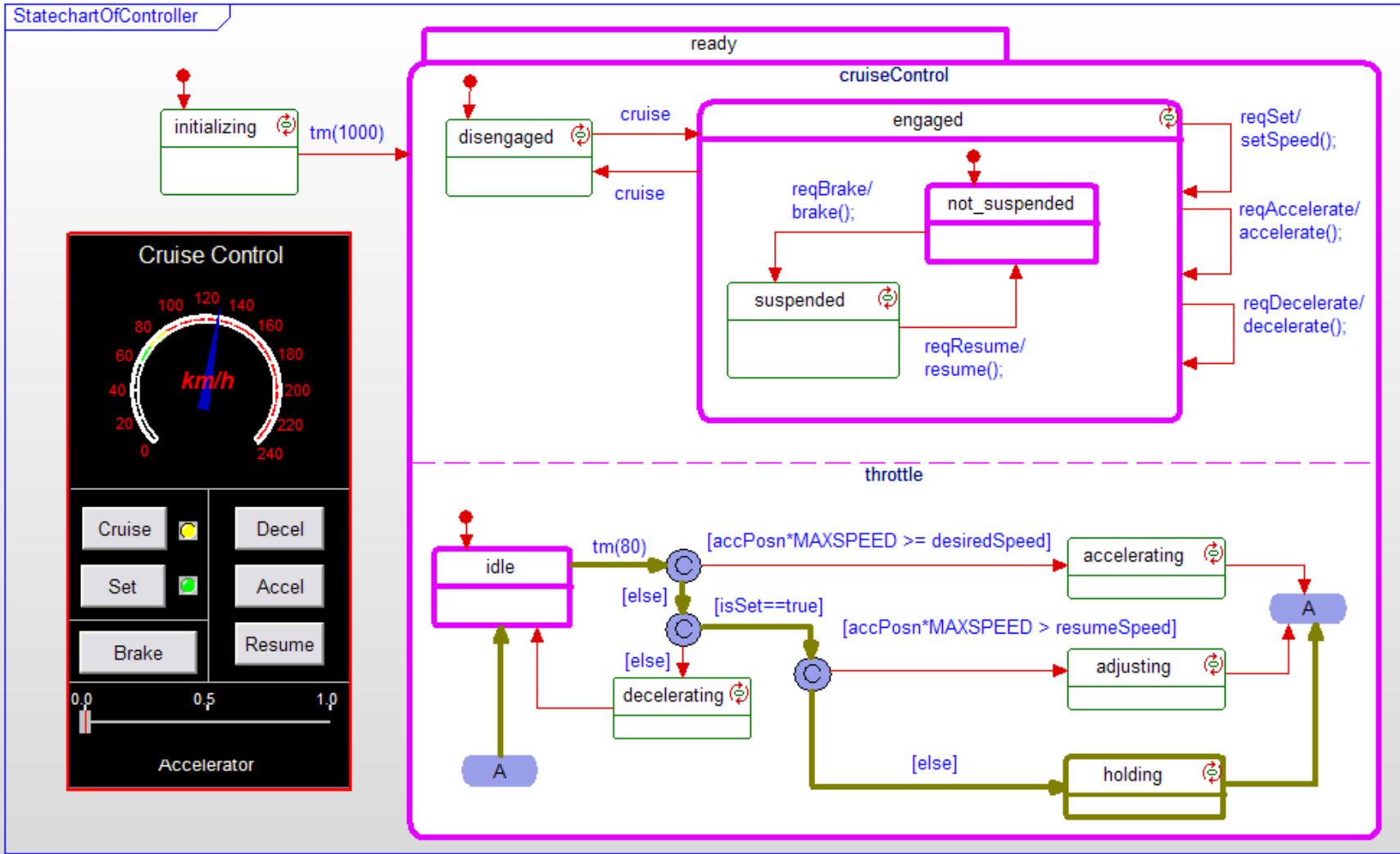


Rhapsody 模型 (2)

- § 连接 Controller desiredSpeed flowPort 到 pFeedback SimulinkBlock的响应输入 flowPort, 当 setDesiredSpeed() 被触发时, Controller.desiredSpeed的值就会被发送到 Simulink
- § 同样, actualSpeed 也被从 Simulink 输出到 Rhapsody
- § 我们可以用 UML 的状态机来描述 Controller 的动态行为



Controller的状态机



总结

- AUTOSAR作为一个新的汽车电子的开发标准，将在未来推动汽车电子研发的标准化、通用化和高效化。
- 目前AUTOSAR正处于前期试验状态，并没有大规模应用。各个厂家正在做自己的RTE和BSW，并且做一些试验性的评估。
- AUTOSAR的工具链比较广，从前期的系统描述和定义工具，到RTE、BSW定义和生成工具。需要整个产业链的整体协作。
- Rhapsody作为前期的系统描述和定义工具。目前最大的用途是用来根据AUTOSAR的标准接口来描述SW/C
- Rhapsody和Simulink是作用在不同领域的建模和代码生成工具，可以联合使用发挥最大功效



Innovate2011

IBM Rational 软件创新论坛



Software. Everywhere.

