Andreas Löscher
TU Chemnitz
AUTOSAR = Automotive Open System Architecture

AUTOSAR Consortium

AUTOSAR Standard for software components in automobiles

"cooperate on standards – compete on implementation"
History

- Founded in 2003 by BMW, Bosch, Continental, Daimler Chrysler, Siemens VDO and Volkswagen
- Till 2004 Citroën Automobiles, Ford Motor, General Motors, Peugeot and Toyota Motors joined the group
- Development of a standard for software components in automobiles
Contributors
Why a standard?

- Non existence of a standard
- Complexity of systems and software
- Growth of costs
- Lack of scalability and flexibility
Goals

- Manage increasing complexity associated with growth in functional scope
- Improve flexibility for product modification, upgrade and update
- Improve scalability of solutions within and across product lines
- Improve quality and reliability
- Enable detection of errors in early design phases
Schedule

Top Level Schedule for AUTOSAR in phase II

- Basic Software & RTE
  - Specification R3.0: Finalizat.
  - Specification R4.0: Maintain R3.0 Specifications
  - CT Pilot: Conformance Test Preparation
  - Methodology & Templates R3.0: Finalizat.
  - Methodology & Templates R4.0: Maintain, R4.0
  - Specification Application Interfaces R3.0: Finalizat.
  - Specification Application Interfaces R4.0: Maintain R4.0

- Application Interfaces
  - Specification 3.0 Ready: 28.09.07
  - Full Release 3.0: 21.12.07
  - Specification 3.1 Ready: 30.05.08
  - Full Release 3.1: 27.06.08
  - Spec R4.0 MS3 Ready: 12.12.08
  - Validation & CT 4.0: 24.07.09
  - Full Release 4.0: 27.11.09

- AUTOSAR Phase II (2007 – 2009)

Full Release 4.0 shifted
Virtual Functional Bus

- Central Concept of AUTOSAR
- Abstract Communication Layer
- Encapsules underlying architecture (CAN, Flexray)
- Provides Sender-Receiver and Server-Client communication
- The concrete implementation of the VFB on an electronic control unit (ECU) is the runtime environment (RTE)
Virtual Functional Bus

AUTOSAR Runtime Environment (RTE)

AUTOSAR Software

- Application Software Component
- Actuator Software Component
- Sensor Software Component

Basic Software

ECU-Hardware

API 2
VFB & RTE relevant

API 1
RTE relevant

API 3
Private Interfaces inside Basic Software possible

Operating System

Standardized Interface

Services

Standardized Interface

Communication

Standardized Interface

Microcontroller Abstraction

Standardized Interface

Complex Device Drivers
Virtual Functional Bus

- AUTOSAR Software Component
  - Interface
  - ECU Firmware
  - Standard Software

- Application Software Component
  - AUTOSAR Interface
  - Complex Device Drivers

- Actuator Software Component
  - AUTOSAR Interface
  - ECU Abstraction

- Sensor Software Component
  - AUTOSAR Interface
  - Standardized AUTOSAR Interface
  - Services

- Application Software Component
  - AUTOSAR Interface
VFB – Sender Receiver Pattern

- Simple information distribution
- Logically atomic transfer
- Datatype is known at configuration time
- "last-is-best" also provides an invalidate service
Server provides a service and the client uses this service

Client starts the communication by requesting the service

The server waits for incoming request

An error is sent if the service can not be provided
Non-functional properties

Timing
Timing Model

- Generic Timing Framework
- Based on events

Reality \rightarrow Implementation \rightarrow Model
Events

- Occurs at a concrete moment
- An event has a type which must be specified during the configuration of the system
- Events of a type can occur multiple times
- The expected behaviour of an event can be described as model
Recurring Event Model

- Models sporadic events
- Highest and lowest inter occurrence time
- The next event must occur in this interval beginning from its last occurrence
Periodic Event Model

- Model periodic events
- Events do not need to occur exactly after the defined period
- A jitter can be defined
- The difference between the maximal and minimal delay must be smaller than the jitter
Timing Chains

- Only simple behaviour can be modeled with single events
- Timing chain are composed of many segments
- Every segment is associated with an event
- Like events the desired behaviour is described as model which the implementation must fulfil
MinMaxOneToOne Model

- Timing chain is activated once for every stimulus
- Response time must be between specified minimum and maximum
- Used to describe event driven systems
MaxAge Model

- Only one Parameter MaxAge
- Not necessary to react on every stimulus
- Provided information must be new enough
- Every stimulus leads to an activation or a later activation leads to a response within the MaxAge of the former stimulus
Communication

ECU1

DataElementAvailableOnPPort
ClientInvocationAvailableOnRPort
ServerResponseAvailableOnPPort

VFB

DataElementAvailableOnRPort
ClientInvocationAvailableOnPPort

ECU2
Evaluation

- Only basic features
- Not even part of the standard
- What is with shared resources, systematical timing analysis?
- Industry partners demand a more feature rich timing model
- The current timing model is too simple
- The problem has been recognized and a new timing framework will be introduced in further releases
Industry driven project to provide a proper timing framework

Uses a Timing Augmented Description Language (TADL) called MARTE

Provides many features which AUTOSAR does not like, systematical timing analysis procedure on model level, shared resources, timing behaviour of communication in respect to the schedule of the software components
Non-Functional Properties

Reliability
System Description

- Contains all information about the components of a system
- Contains constraints for mapping and communication
Software components (SWC) are developed independent from the used ECU

In some cases SWCs must be mapped in a distinct way

e.g. Components which are already mapped during a earlier design step

A component can be forced to be mapped on a distinct ECU

A software component must not be mapped on a specific ECU
Mapping Constraints

- More dynamic constraints possible
- Separated and not separated clusters
- Separated Clusters must not be mapped on the same ECU
- Clusters must be mapped on the same ECU
Communication Constraints

- Similar to the mapping constraints
- Specific path, Forbidden path, Common paths, Divided paths
Higher reliability can only be achieved through redundancy

No implicit declaration of high reliable components

Explicit declaration is possible (separated clusters, divided paths)

May change in further version
Review

- Does AUTOSAR provide enough methods to describe non-functional properties?

- Timing Framework:
  - Generic Timing Framework is too simple
  - No shared resources or support for analytical analysis
  - The possibilities provided here are not enough
  - The problem has been recognized and there will be another timing framework in further releases
  - TIMMO project is promising as replacement
Review

- Reliability
  - The constraints that can be defined in the system description are enough to build reliable systems
  - No implicit declaration of high reliable components
How to build a system?

- Per hand?
  - Many Templates that must be filled out
  - Difficult to maintain, create and analyze

- Tools
  - TargetLink from dSpace
  - Ascet from ETAS
  - AUTOSAR provides its own UML model
How to build a system?
Discussion
Appendix
Software Components Description

- Software Components Description (SCD) contains all of the information of the software components of a system
- Multiple implementations possible
- Dependencies
- Resource Consumption
- Execution Time
Resource Consumption

- Resources are represented through memory
- Static allocated and dynamic allocated memory
- Static allocated memory is known on configuration time
- ECU must provide this memory if the software component is mapped on it
- Dynamic allocated memory is more complicated
Dynamic allocated Memory

- Divided in stack and heap usage
- AUTOSAR provides the possibility to describe worst case usage, measured usage, rough estimated usage for stack and heap memory
- Test patterns can be given for measured usage with a resulting maximum and minimum usage
- Segmentation can be a problem with heap memory
- Dynamic allocated memory should be used with caution
Execution Time

- Similar to the resource consumption
- Worst Case Execution Time, Simulated Execution Time, Rough Estimated Execution Time
- Context in which the execution time is achieved can be described
AUTOSAR 3.1 Specification on www.autosar.de


Illustrations on Slide 4, 9, 10 and 33 as well as the AUTOSAR logo are taken from the AUTOSAR 3.1 specification