Introduction to AUTOSAR

Massimo Violante
Politecnico di Torino
Dip. Automatica e Informatica
Torino, Italy

资料整理
火龙果软件
荐

Outline

- Introduction
- ECU software architecture
- The AUTOSAR approach
- Software components
- AUTOSAR methodology

资料整理
火龙果软件
荐

Outline

- Introduction
- ECU software architecture
- The AUTOSAR approach
- Software components
- AUTOSAR methodology

资料整理
火龙果软件
荐

What is AUTOSAR?

- ww.autosar.org:
  - "AUTOSAR (AUTomotive Open System ARchitecture) is an open and standardized automotive software architecture, jointly developed by automobile manufacturers, suppliers and tool developers."
- Consortium founded in 2003:
  - BMW, DaimlerChrysler, Bosch, Continental, Volkswagen and Siemens VDO
  - Later Ford, General Motors, Toyota and PSA (Peugeot Citroen) joined the consortium as core members
- The AUTOSAR standard will serve as a platform for future vehicle applications

资料整理 火龙果软件 荐

What is AUTOSAR?

Why do we need AUTOSAR?

- Increasing complexity of software in automotive systems
- There is no standardized software architecture

In a single car*	1994	2008
Control Units	40	60
MIPS	45	1150
MHz	85	2000
MCU-Storage (Program + Data)	1,1 MB+160 kB	19 MB+1,25 MB
Transistors	21 m.	340 m.

(* without Infotainment)
Source: ElektronikNet.de http://www.elektroniknet.de/home/automotive/autosar/multiple-applikationen-in-steuergeraeten/

资料整理 火龙果软件 荐

Main drivers for AUTOSAR

- Management of E/E complexity associated with growth in functional scope
- Flexibility for product modification, upgrade and update
- Scalability of solutions within and across product lines
- Improved quality and reliability of E/E systems

资料整理
火龙果软件
荐

Main goals of AUTOSAR

- Fulfillment of future vehicle requirements:
  - Availability and safety, SW upgrades/updates and maintainability
- Increased scalability and flexibility to integrate and transfer functions
- Higher penetration of "Commercial off the Shelf" SW and HW components across product lines
- Improved containment of product and process complexity and risk
- Cost optimization of scalable systems

资料整理
火龙果软件
荐

AUTOSAR Software Components

- AUTOSAR is based on the concept of separation between infrastructure and application
  - Infrastructure: services providing an execution environment to abstract hw details
  - Application: vehicle functions of interest
- Application consists of interconnected software components
- Atomic software component: a software component can not be distributed over several ECUs
- AUTOSAR software component implementation is independent from the infrastructure

资料整理 火龙果软件 荐

Outline

- Introduction
- ECU software architecture
- The AUTOSAR approach
- Software components
- AUTOSAR methodology

资料整理
火龙果软件
荐

Layered Software Architecture

资料整理
火龙果软件
荐

Layered Software Architecture

- Microcontroller Abstraction Layer:
  - Lowest software layer of the Basic Software
  - Makes higher software independent of Microcontroller

资料整理
火龙果软件
荐

Layered Software Architecture

- ECU Abstraction Layer:
  - Interfaces the drivers of Microcontroller Abstraction Layer
  - Makes higher software layers independent of ECU hardware layout
  - Offers access to I/O Signals

资料整理
火龙果软件
荐

Layered Software Architecture

- Service Layer:
  - Highest layer of the Basic Software
  - Offers Memory Services, Diagnostic Services, ECU state management
  - Provides basic services for application and basic software modules

资料整理
火龙果软件
荐

Layered Software Architecture

- Runtime Environment:
  - Middleware Layer
  - Provides communication services for the application software
  - Makes AUTOSAR Software Components independent from the mapping to a specific ECU

资料整理 火龙果软件 荐

Layered Software Architecture

- Application Layer:
  - "component style"
  - Software components communicate with other components and/or services via the RTE

资料整理 火龙果软件 荐

Detailed view

资料整理 火龙果软件 荐

Outline

- Introduction
- ECU software architecture
- The AUTOSAR approach
- Software components
- AUTOSAR methodology

资料整理
火龙果软件
荐

Standardization

- The software implementing the automotive functionality is encapsulated in software components
- Standardization of the interfaces is central to support scalability and transferability of functions
- Any standard-conformant implementation of a software component can be integrated with substantially reduced effort in a system

资料整理
火龙果软件
荐

AUTOSAR architecture

资料整理 火龙果软件 荐

AUTOSAR architecture

- Software component (SW-C)
  - Encapsulate an application that runs in the AUTOSAR infrastructure
  - It has well-defined standard interfaces
  - Interfaces and other aspects needed for integration of SW-C a standard description format is provided (SW-C description)

资料整理
火龙果软件
荐

AUTOSAR architecture

- Virtual Function Bus (VFB)
  - Sum of all communication mechanisms and interfaces to the basic software provided by AUTOSAR on a technology independent level

资料整理 火龙果软件 荐

AUTOSAR architecture

- System constraints and ECU description
  - Description format for: system, ECU resources and configuration
  - Needed for deploying SW-C on a network of ECUs

资料整理
火龙果软件
荐

AUTOSAR architecture

- Mapping on ECU
  - Methodology and tool support for implementing SW-Cs on ECUs, performing for each ECU configuration and generation of:
    - Runtime Environment (RTE) → it implements the VFB on each ECU
    - Basic Software (RTOS services)

资料整理 火龙果软件 荐

Outline

- Introduction
- ECU software architecture
- The AUTOSAR approach
- Software components
- AUTOSAR methodology

资料整理
火龙果软件
荐

Software component

- It encapsulate part of the functionality of the application
  - AUTOSAR does not specify the granularity of SW-C
- It is an atomic software component → it is statically assigned to one ECU
- AUTOSAR does not specify how the SW-C should be implemented
  - Handwritten/automatically generated from a model

资料整理
火龙果软件
荐

SW-C description

- It contains:
  - The operations and data elements that the software component provides and requires → PortInterface concept
  - The resources needed by the software component (memory, CPU-time, etc.),
  - Information regarding the specific implementation of the software component

资料整理
火龙果软件
荐

SW-C implementation

- It is independent from:
  - The type of microcontroller of the ECU and the type of ECU on which the component is mapped
    - The AUTOSAR infrastructure takes care of providing the software component with a standardized view on the ECU hardware
  - The location of the other components with which it interacts. The component description defines the data or services that it provides or requires. The component doesn't know if they are provided from components on the same ECU or from components on a different ECU
  - The number of times a software component is instantiated in a system or within one ECU

资料整理
火龙果软件
荐

SW-C description levels

- Virtual functional bus level
- RTE level
- Implementation level

资料整理
火龙果软件
荐

SW-C description levels

- Virtual functional bus level
  - Components are described with the means of
    - Datatypes and interfaces
    - Ports and connections between them
    - Hierarchical components
  - At this level, the fundamental communication properties of components and their communication relationships among each other are expressed
- AUTOSAR terms
  - Port Interfaces and Ports
  - Compositions

资料整理
火龙果软件
荐

Example

- SeatHeatingControl
- Input:
  - Passenger is sitting
  - Setting of the seat temperature
  - Information from a central power management system
- Output:
  - DialLED associated to the seat temperature dial
  - Heating element
  - It can be calibrated, needs the status of the ECU on which the component runs and requires access to local non-volatile memory

资料整理 火龙果软件 荐

Port Interface

- It defines the contract that must be fulfilled by the port providing or requiring that interface
  - Client-server: the server is provider of operations and several clients can invoke those operations
  - Sender-receiver: a sender distributes information to one or several receivers, or one receiver gets information (events) from several senders. A mode manager can notify mode switches to one or several receivers
  - Parameter Interface: it allows SW-C access to either constant data, fixed data or calibration data

资料整理 火龙果软件 荐

Port Interface

- Non volatile Data Interface: provide element level access (read only or read/write) to non volatile data
- Trigger Interface: it allows software components to trigger the execution of other software components
- Mode Switch Interface: the mode switch interface is used to notify a software component of a mode. The mode manager provides modes that can be used by mode users to adjust the behavior according to modes or synchronize activities to mode switches.

资料整理
火龙果软件
荐

Client-server

- A client-server interface defines a set of operations that can be invoked by a client and implemented by a server
- Example
  - The interface "HeatingElementControl" defines a single operation called "SetPower" with a single ingoing argument called "Power". The operation can return an application error called "HardwareProblem"

<<ClientServerInterface>>
HeatingElementControl

ApplicationErrors:
HardwareProblem

Operations:
SetPower(
IN ARGUMENTint32 Power,
POSSIBLEERROR=HardwareProblem)

资料整理 火龙果软件 荐

Sender-receiver

- A sender-receiver interface defines a set of data-elements that are sent and received over the VFB
- Example
  - Simple sender-receiver interface called “SeatSwitch” containing a single data-element called “PassengerDetected”

<<SenderReceiverInterface>>
SeatSwitch

DataElements:
boolean PassengerDetected

资料整理
火龙果软件
荐

Ports

- The ports of a component are the interaction points between components
- A port of a component is either a PPort or an Rport
  - PPort provides the elements defined in a port-interface
  - RPort requires the elements defined in a port-interface
- A port is thus typed by exactly one port-interface

资料整理
火龙果软件
荐

Port types

PPort	sender-receiver	No	The component provides values for the data-elements
RPort	sender-receiver	No	The component reads or consumes values for the data-elements
PPort	client-server	No	The component provides (=implements) the operations defined in the interface
RPort	client-server	No	The component requires (=uses or invokes) the operations defined in the interfaces

资料整理 火龙果软件 荐

Port types

PPort	parameter (this includes providing calibration data)	No	The component provides parameter data (either fixed, const or variable)
RPort	parameter (this includes requiring calibration data)	No	The component requires parameter data (either fixed, const or variable)
PPort	sender-receiver	Yes	The component provides data-elements and mode-groups to an AUTOSAR Service
RPort	sender-receiver	Yes	The component reads/consumes data-elements and mode-groups from an AUTOSAR Service

资料整理 火龙果软件 荐

Port types

PPort	client-server	Yes	The component provides (=implements) operations for an AUTOSAR Service
RPort	client-server	Yes	The component invokes operations from an AUTOSAR Service
RPort	sender-receiver	Yes	A component requires access to non volatile data provided by an NV

资料整理
火龙果软件
荐

Port types

PPort	Sender-receiver	Yes	The NV Block Component provides access to non volatile data
PPort	Trigger	No	Application component with trigger source
RPort	Trigger	No	Application component with trigger sink

资料整理
火龙果软件
荐

Port types

RPort	Trigger	Yes	Service with trigger sink
PPort	Trigger	Yes	Service with trigger source
RPort	mode switch	No	Mode Switch user

资料整理
火龙果软件
荐

Port types

PPort	mode switch	No	Mode Switch manager
RPort	mode switch	Yes	Mode Switch user
PPort	mode switch	Yes	Mode Switch manager

资料整理
火龙果软件
荐

Client-server

- When a PPort of a component provides a client-server interface, the component provides an implementation of the operations defined in the interface
- Example:
  - "SeatHeating" implements the operation "SetPower" and makes it available to other components through the port "Setting"
  - The component "SeatHeatingControl" uses the operation "SetPower" and expects such an operation to be available through the port "HeatingElement"

资料整理 火龙果软件 荐

Example

资料整理
火龙果软件
荐

Sender-receiver

- A component providing a sender-receiver interface generates values for the data elements defined in the interface
- Example:
  - The component "SeatSwitch" generates values for the Boolean value "PassengerDetected" through its port "Switch"
  - Similarly, the component "SeatHeatingControl" can read the data-element "PassengerDetected" through its port "SeatSwitch"

资料整理
火龙果软件
荐

Example

资料整理
火龙果软件
荐

Compositions

- Ports between components that need to communicate with each other are hooked up using assembly-connectors
- Such an assembly-connector connects one RPort with one PPort

资料整理
火龙果软件
荐

Example

DialLED is sent to LED

资料整理
火龙果软件
荐

Example

The operation invoked over Position is requested to the server over the Setting port

资料整理
火龙果软件
荐

SW-C description levels

- RTE level
  - Component behavior is described in terms of:
    - RTE events
    - Schedulable units
  - For instance, for an operation defined in an interface on the VFB, the behavior specifies which of those units is activated as a consequence of the invocation of that operation

资料整理
火龙果软件
荐

Runnables

- SW-C actual implementation consists of a set of runnable entities (aka runnables”)
- A runnable entity is a sequence of instructions (provided by the component) that can be started by the Run-Time Environment

资料整理
火龙果软件
荐

Example

资料整理 火龙果软件 荐

Example

资料整理 火龙果软件 荐

Example

Invoked when other SW-C requires "Setting" P-Port. It will requires the RTE to provide the data over Rport "SeatSwitch"

资料整理 火龙果软件 荐

Runnables

- A runnable entity runs as a task on the ECU microprocessor
- The task provides the common resources to the runnable entities such as a context and stack-space
- Typically the operating-system scheduler has the responsibility to decide during run-time when which task can run on the CPU (or multiple CPUs) of the ECU
  - There are many standard strategies that schedulers can use (e.g. priority-based preemptive, round- robin, time-triggered...).

资料整理 火龙果软件 荐

RTEEvents

- Mechanisms through which the component's implementation (for example "C" functions) is invoked:
  - Fixed-time schedules (for example: many components need to run "cyclically")
  - Events related to the communication mechanisms (for example some components might want to be notified upon the reception of data from other components)
  - Events related to physical occurrences (i.e. a triggered event)

资料整理
火龙果软件
荐

SW-C description levels

- Implementation level
  - For each runnable the corresponding behavior is defined (as handwritten or automatically generated code)
  - The requirements for the RTE are defined in terms of:
    - Which runnables need to be called cyclically
    - Which runnables need to be called in response to events related to communication or other sources
    - How the component would like to access the information in its ports or invoke the operations that it requires from other components
    - Any other resources the component requires, such as AUTOSAR services or local memory

资料整理
火龙果软件
荐

VFB view

资料整理 火龙果软件 荐

RTE view

资料整理 火龙果软件 荐

Implementation view

资料整理 火龙果软件 荐

Outline

- Introduction
- ECU software architecture
- The AUTOSAR approach
- Software components
- AUTOSAR methodology

资料整理
火龙果软件
荐

AUTOSAR methodology

- Design steps go from the system-level configuration to the generation of an ECU executable

资料整理 火龙果软件 荐

Methodology

- The System Configuration Input must to be defined
  - The software components and the hardware are defined and selected
  - The overall system constraints are identified
- XML templates:
  - Software Components: each SW-C requires a description of the software API e.g. data types, ports, interfaces
  - ECU Resources: specifications regarding the CPU, memory, peripherals, sensors and actuators
  - System Constraints: bus signals, topology and mapping of SW-C

资料整理
火龙果软件
荐

Methodology

- System Configuration Descriptions contains
  - The mapping of SW-C to ECU
  - Bus topology
- Further steps have to be performed on each ECU
- Extract ECU-Specific Information extracts the information from the system configuration description for a specific ECU

资料整理 火龙果软件 荐

Methodology

- Configure ECU configures:
  - RTE
  - Basic Software
- It generates:
  - Task scheduling
  - Required basic software modules
  - Configuration of the basic software
  - Assigment of runnables to tasks
- The ECU software can be built from this information

资料整理
火龙果软件
荐

methodology

- The executable code for the ECU si generated:
  - Code generation
  - Code compilation
  - Code linking

资料整理
火龙果软件
荐

Flow for each SW-C

资料整理 火龙果软件 荐

Flow for each SW-C

- List of runnables
- Definition of input-data/runnable association

资料整理 火龙果软件 荐

Flow for each SW-C

The AUTOSAR Component API Generator reads the Component Internal Behavior Description of the appropriate software component and generates the Component API accordingly. The Component API contains all header declarations for the RTE communication.

资料整理 火龙果软件 荐

Flow for each SW-C

Actual implementation of the SW-C resulting in:

- Component implementation (C code)
- Component Internal Behavioral description (documentation)
- Component Implementation Description (e.g. compiler settings, optimizations, etc.).

资料整理 火龙果软件 荐

Flow for each SW-C

资料整理 火龙果软件 荐