How to enrich AUTOSAR based design processes with enabled System-level Analysis

Werner Damm
The Challenge

To have the cake and eat it:

Maintain target-hardware independence of capturing function networks
re-use across multiple platforms!

Yet assess realizability of new functions early (but this is dependent on the target ... )
frontloading!
Structure of Talk

Key Benefits of Autosar
Rich components
Speeds integration technology
Rich Component Based Design
Adding Speeds to Autosar: what it takes
Drivers for Change in automotive embedded systems development

**Flexibility**
- Decouple growth rate of \#functions from growth rate of \#electronic components
- Freedom in choosing boundary of in-house and external development

**Adaptability**
- Towards emerging technologies
- Towards emerging hardware platforms
- Maintainability: at life-time

**Cost**
- Decouple growth of \#functions from growth rate of development costs
- Decouple growth rate of number of supported platforms from development costs

**Quality**
- Maintain/Improve Quality while allowing growth of \#functions
Anticipated Changes in Processes

Strong push to virtual subsystem models Target independent
- Topic in Autosar

Strong push towards component based development
- Topic in Autosar
- Requires component characterizations dealing with non-functional aspects (e.g. real-time, safety, ...)

Need to support safety standards
- to support IEC 61508 customized to automotive domain: ISO WD 26 26 2 – compliant processes

Deployment analysis capabilities will be key competence
- for price-competetive offerings of tier 1 suppliers
- For realizability analysis of new functions for innovator OEMs
Autosar

- World wide private partnership of automotive OEMs, tier1/tier2 suppliers, SW-houses, vendors to push standardization of E/E architectures with more than 100 partners
- Standardized meta-model, release 3.0 available from www.autosar.org
Autosar

Decoupling functional design from implementation platform

© Autosar Consortium
Achievements and exploitation of the AUTOSAR development partnership, Convergence 2006
Key Benefits Autosar Design Methodology

Separation of application specification and development from implementation on specific hardware platforms

- a key enabler for enhanced cross platform re-use of applications
- allows cost-tuned hardware implementations for the specific set of functions and features integrated in a particular vehicle development.

- Supports a stepwise refinement of the overall system design
  - starting at a coarse grain system topology defined in the system template
  - stepwise enriched by hardware-related information based on mappings to the ECU network.

- The AUTOSAR metamodel gives a formal standardized specification of information required for a functional system design
  - provides skeleton for additional information to support further aspects of a full system-level analysis.
  - Allows to extend early assessment of key functional requirements by an analysis of non-functional requirements based on an extended AUTOSAR meta-model.
Linking and Autosar
The Story

- Heterogeneous model-based design
- Semantic integration of industry standard tools

Contract-based multi-criteria system-level analysis

Analysing compliance of Autosar Configurations against vertical assumptions
Technical Highlights of the IP Speeds

Speeds provides

- The capability of Modeling and Integration of Architectural Abstractions at all System Design Levels for multiple viewpoints including real-time and safety.

- A Rich Component Model allowing to completely encapsulate functional and non-functional aspects of a design in an assume-guarantee style with cross viewpoint dependencies, including the capability of expressing assumptions on lower design levels captured as architectural abstractions.

- A harmonized meta-model allowing a semantic integration of industry standard system- and software design tools supporting rich components based on an open tool integration standard, compatible with the Autosar Metamodel.

- A suite of compositional analysis and design space exploration methods supporting real-time and safety analysis.
Acknowledgements

This talk represents the collaborative effort of the Speeds consortium

End-users:
- Airbus, Bosch, Carmeq, EADS, IAI, Knorr-Bremse, MagnaSteyr, Saab

Vendors
- Esterel Technologies, Extesy, Geensys, Telelogic

Research Organizations
- INRIA, OFFIS, Parades, Verimag

The integrated project Speeds is funded by the European Commission under contract FP6-033471
Rich Components
„Rich Components“- Objectives

To provide a characterization of components of automotive electronic components

- supporting all phases, levels, and viewpoints of electronic system design
- Allowing complete re-use (across multiple platforms, across multiple organizations, and/or as part of design libraries)
- Allowing characterization of allowed/assumed environments of component (for all viewpoints)
- Basis for (de-facto) standardization, compatible with Autosar Component Model

As basis for tool-independent meta-model for capturing and validating function networks

- Supporting semantic based integration of industry standard System & SW design tools (UML, Matlab-Simulink/Stateflow, ASCET, …)
- Supporting view-point specific and cross viewpoint requirement capturing, modeling, analysis and design
Rich Component Model

Follows Design by Contract Paradigm

Assumptions
- Reflect current degree of knowledge of anticipated design context
- Determine boundary conditions on actual design context for each viewpoint under which component is promising its services
- Are decorated with confidence levels

Promises
- Are guaranteed if component is used in assumed design context

Organized per viewpoint
- Behaviour, Coordination, Safety, Real-Time, ...
- But allows specification of cross viewpoint dependencies
**Contracts**

**Assumption:**
- Status of vehicle available every t ms

**Promise:**
- \( \text{Status}(\text{vehicle}') = \text{braking}(p) \) implies
  - Signal(Brake\(p)\)) within t' ms

**Assumption:**
- Distance sensor values \(d\) available every t ms

**Promise:**
- \( \langle d \text{ at } t1\rangle < \langle d \text{ at } t0\rangle \) implies
  - Signal(Brake) within t' ms

...
Adding contracts: Example
Expressing assumptions on task activation

Use contract editor to identify relevant component and port

Choose pattern from real-time library to capture assumptions

Here:
Activation pattern for periodic systems
Adding contracts: Example
Expressing promise on maximal delay

Use contract editor to identify relevant component and port

Choose pattern from real-time library to capture assumptions
Here: Delay Pattern

Link assumption on task activation to promise on maximal delay
Heterogenous Rich Components
Metamodel Characteristics

Based on SysML

- Contracts (Assumptions, Promises)
- Viewpoints
- Layers (Functional Network, ECU, Physical Architecture, ..)

Available as Standalone Metamodel Implementation and SysML Profile

Equipped with formal semantics

- building on few „semantic atoms“: continuous evolution, discrete transitions, random choice
- semantic basis for contract compliance/dominance
for all viewpoints $v$: 
\[ \cap L(A(OutI.v.prj)) \subseteq \cap L(A(\text{InI}.v.\text{assm}_v)) \]
Use Cases supported by Speeds bus
Use Case 1: Designing Functional Network (Structure) in Tool 1

Classification of contracts w.r.t. viewpoints

**Functional Behaviour**

**Assumption:**
- in1 in range \([a..b]\)
- in2 in range \([a..b]\)

**Promise:**
- out = funct(in1, in2)

**Real-Time**

**Assumption:**
- every request for sensor values will be answered within \([t0..t1]\)

**Promise:**
- new actuator values will be transmitted every \(t\) ms
Use Cases supported by Speeds bus

Use Case 2: Designing a Component (Implementation) in Tool 2

Contracts for F3

Contract C1:
Assumption: .......
Promise: .......

Contract C2:
Assumption: .......
Promise: .......

Transmission ratio

1
Ne

2
gear

Torque converter

3
Nout

Tin

1
Ti

2
Tout

Nout

Nin

Process Advisor

Tool 2 - Component F3
Use Cases supported by Speeds bus

Use Case 3: Analysis 1 (Compatibility)

Check Compatibility of Contracts
- Defined in the Structure Definition
- Defined in the Implementation

Check Compatibility of Ports
- Defined in the Structure Definition
- Defined in the Implementation

Check ...
Use Cases supported by Speeds bus

Use Case 4: Analysis 2 (Advanced Analysis Techniques)
RCM in the Design Process (Focus Real-time)

OEM

Requirements

Specification of Functional Network

Design Space Exploration

Decomposition Of Contracts

Contracts (A, P)

Guaranteed by OEM (resp. other suppliers)

Final Deployment

Supplier

Component Design

Contracts (Ai, Pi)

To be Guaranteed by Supplier
RCM in the Design Process (Focus Real-time)

- Specification of Functional Network
- Design Space Exploration
- Decomposition of Contracts
- Guaranteed by OEM (resp. other suppliers)
- Requirements
- Component Design
- Final Deployment

Functional Layer

- F1
- F2
- F3
- F4

Architectural Layer

Initially: Unspecified or estimations based on experiences

Contracts Real-time View

"Global End-to-End Deadlines"
System level timing analysis

- Is applied to meta-model representation of designs at function layer
- Verifies end-to-end properties and/or timed sequence diagrams
- Is based on vertical assumptions (typically delay constraints)
- Assumptions annotated with confidence levels

Vertical assumption
- Delay(DS.d, AK.ds) = [0, 6] ms
- Delay(DS.d, bsp) = [0, 5] ms
- Delay(md, d) = [0, 1] ms

Vertical assumption
- Delay(vs, gd) = [0, 8] ms
- Delay(ds, bsp) = [0, 5] ms

ConfLevel=low

End-to-end deadline
- ConfLevel=low
- [0, 50] ms

ENV
Contract Compatability

Assumptions in contracts associated with ports constrain allowed environments
Contracts of virtual communication component characterize space of viable communication architectures
Example use case for contract compatibility: composition of components coming from different suppliers
Example compatibility check:
Is contract of component DS compliant to assumptions of communication component
SPEEDS – System-level hosted simulation
Design Space Exploration

Analysis of the architecture also w.r.t. other viewpoints, e.g. **Safety**
Given:
- Partially defined architecture
- Network of functions, partially known by re-use of already designed components

Early assessment of architectural solutions
Contracts to lower layer of abstraction used to define requirements for implementation of functions
Express contracts based on
- Experience
- Previous developments
- Model based estimations
Architectur exploration – Approach

Establish link between vertical assumptions at system level and vertical promises of detailed design

Check dominance

Solution 1

Vertical Promise on AK: AK.vs every 50ms with max. 3ms jitter

Solution 2

Vertical Promise on AK: AK.vs every 50ms with max. 2ms jitter
Assessments of architectural variants supported by automatic analysis methods

Minimize the number of ECUs

Robustness: Often parameters, e.g. WCET, are uncertain. Design optimization has to consider robustness under variations of WCET

Penalty of changes: late integration of new functions induces changes in the already implemented system, but number of changes has to be kept small. Must assess deployment wrt penalty for changes
Creating Deployable Units

Requirements

Specification of Functional Network

Design Space Exploration

Decomposition of Contracts

Functional Layer

Architectural Layer

Real-time View

Assm: ...
Promise: „Local End-to-End Deadlines“

Decomposition with local contracts based on architectural decisions

Contracts Guaranteed by Supplier (resp. other suppliers)

Requirements

Specifications Guaranteed by OEM
Generating requirements for suppliers
Design space exploration returns class of feasible architecture
After finding architecture and communication matrix (used slots in TDMA rounds on bus and ECUs), generate timing requirements for outsourced software components:

- Requirements for these delivered to supplier as part of contract
  - Assume class of architecture
  - Assume allowed behavior of environment (e.g. activation pattern)
  - Promise maximal delay and jitter
- Enables incremental integration by OEM based on contracts
  - If supplier fulfills its contract, integration will be successful
- Allows supplier to develop function in isolation without the need to consider functions coming from OEM or other suppliers
Getting the best of both worlds

RCM supports system level design
RCM enables combination of model based design and contract-based reasoning
AUTOSAR starts on functional network level
Combination of both covers the entire V-cycle starting from requirements and ending in implementation of operational architecture
AUTOSAR enabled system level analysis

System level analysis is based on vertical assumption, e.g. response times of components

AUTOSAR defines operational architecture

Special tailored analysis can be used to verify correctness of assumptions used for system analysis (e.g. scheduling analysis)
Integration with Autosar: what it takes

Principles

- Allow association of contracts to SWCs
- Timing Extensions of Autosar Meta-Model (under development), incl. timing characteristics of BSW
- Similarly: Extensions wrt other supported viewpoints, such as safety
- Analysis of System- and ECU configurations to derive promises characterizing configuration
- Translation of Autosar Meta-Model concepts into HRC meta-model
  - VFB, runnables, topology, tasks, ...
Translation of SW components

Software components of AUTOSAR and those of RCM are similar
Definition as well as refinement can be directly translated
AUTOSAR runnables are transformed into RCM components, Inter Runnable Variables become ports and connections

<<PortSpec>>

P_s

P_{DS}

AK

RPort behavior see next slides

 discrete in speed : float

P_s

P_{DS}

HRC
Triggering runnables – TimerEvents

AUTOSAR runnables are transformed into RCM components, Inter Runnable Variables become ports and connections.

Timer events triggering runnables can be mapped to a component which implements the timers behavior.

Periods of timer events can be mapped to promises of timer components.
Conclusion

Adding Speeds to Autosar provides a seamless design flow from requirements to Autosar based implementations supporting

- Multi-viewpoint contract based requirement capturing
- Semantic integration of industry standard design tools for capturing system-level requirements and specifications
- A suite of analysis and design methods, including
  - Contract compatibility and dominance checking
  - System-level safety and timing analysis
  - Design Space exploration
  - Hosted Simulation enhanced with run-time contract monitoring
- Assessment and optimization of implementation architecture as captured in Autosar configurations