New Vistas on Automotive Embedded Systems

Edited and presented by
Alberto Sangiovanni-Vincentelli
UC Berkeley
Notable Quotes

• The Nihon Keizai Shimbun reported that Japan Ministry of Economy, Trade and Industry estimates that Japanese companies spend more than 100 billion yen (USD $903 million) per year developing automotive-related software. And it isn’t going to get any cheaper, with some analysts estimating costs escalating to 1 trillion yen (USD $9.1 billion) by 2014, according to the daily newspaper.

• So is the industry ultimately moving toward ‘plug-and-play’? Taking the idea of multiplexing to its logical extreme, a carmaker could potentially wait until relatively late in the vehicle’s development cycle before committing to specific electronic hardware yet avoid having to delay - or worse, tear up - its electrical architecture in the last minute.
Electronics, Controls & Software
Shifting the Basis of Competition in Vehicles

- More functions & features
- Less hardware
- Faster

Value from Electronics & Software

Electric Ignition
... 
Mechanical $

Software $
Electronics $
Other $

Fuel Cell
Wheel Motor

Electronic Brake
DoD
GDI
EGR

ACC
Rear Vision
Passive Entry
Side Airbags
Head Airbags

OnStar
OBD II
HI Spd Data
Rear aud/vid
CDs

Electric Fan
Electric Brake
...

1970s 1980s 1990s 2000s 2010s 2020s

100M Lines of Code (+9900%)
1M Lines
20 ECUs
50 ECUs (+196%) (+196%)
$400
$1162 (+196%)
...

Other $ 2%
Mechanical $ 76%
Electronics $ 13%
Software $ 9%

Source: Matt Tsien, GM

Shifting the Basis of Competition in Vehicles

- More functions & features
- Less hardware
- Faster

Value from Electronics & Software

Electric Ignition
... 

Software $
Electronics $
Other $

Fuel Cell
Wheel Motor

Electronic Brake
DoD
GDI
EGR

ACC
Rear Vision
Passive Entry
Side Airbags
Head Airbags

OnStar
OBD II
HI Spd Data
Rear aud/vid
CDs

Electric Fan
Electric Brake
...

1970s 1980s 1990s 2000s 2010s 2020s

100M Lines of Code (+9900%)
1M Lines
20 ECUs
50 ECUs (+196%) (+196%)
$400
$1162 (+196%)
...

Other $ 2%
Mechanical $ 76%
Electronics $ 13%
Software $ 9%

Source: Matt Tsien, GM
Top Priorities

◆ System-level architecture design approach
  ▪ To what extent can we decouple the dimensions of architecture (computation, communication, power, etc.)?
  ▪ What are the guiding principles of system-level architecture design?
  ▪ What are the tools to support system-level architecture design, modeling, simulation, and analysis?

◆ Next-generation architecture strategy
  ▪ What is the long-term architecture vision
    ▪ Independent of (not biased by) today’s architecture
    ▪ Not just evolution of Michigan A / Global A.
  ▪ What is the best approach to incrementally transition to the long-term architecture?
  ▪ Is Global A architecture good enough for the long term? How much better is possible?
AUTOSAR – ECU Software Architecture

AUTOSAR Runtime Environment (RTE)

AUTOSAR Software

Automatic Open System Architecture (AUTOSAR):

- Standardized, openly disclosed interfaces
- HW independent SW layer
- Transferability of functions
- Redundancy activation

AUTOSAR RTE:
by specifying interfaces and their communication mechanisms, the applications are decoupled from the underlying HW and Basic SW, enabling the realization of Standard Library Functions.

"New Vistas on Automotive Embedded Systems", ASV

ITR Review, Oct. 4, 2006
AUTOSAR Organization

AUTOSAR – Core Partners and Members
Status: 24th February 2005

Core Partner

Associate Members

Premium Members

General OEM

Generic Tier 1

Standard Software

Tools

Semi-conductors

Up-to-date status see: www.autosar.org

"New Vistas on Automotive Embedded Systems", ASV
ITR Review, Oct. 4, 2006
Metro: Separation of Concerns

Development Process:
- Analysis
- Specification
- Implementation

IPs
- C-Code
- Matlab
- ASCET

Behavior Components
- System Behavior
- System Architecture

Virtual Architectural Components
- CPUs
- Buses
- Operating Systems

Mapping
Performance Analysis
Refinement
Evaluation of Architectural and Partitioning Alternatives

"New Vistas on Automotive Embedded Systems", ASV
ITR Review, Oct. 4, 2006
Design Practice: Mismatch

- Functional Modeling and Code Generation assume uniprocessor implementation….
  - Modeling and stability analysis for control algorithms with Simulink
  - Code generation with RealTime Workshop
- But then code is distributed
- Architectural limitations
  - Shared buffers and clock drift between processors (ECUs)
  - Symptoms: Message loss and duplication
- Current mitigation
  - Limited analysis
  - In-vehicle testing: Expensive, not exhaustive
  - Oversampling: Brute force, too conservative
Stabilitrak Case Study with Lossy MoC

- Drive-by-wire application on distributed CAN platform
- System model accurately captures design space
  - Loss and duplication
  - Message latency
  - Priority inversion
- Metropolis library to support lossy MoC

Architecture Model: Abstraction Levels

"New Vistas on Automotive Embedded Systems", ASV
Matching Models of Computation

- The functional and architectural models should be described using the same model of computation.

- Architecture Characteristics:
  - Network of processes connected by point-to-point FIFOs
  - Non-blocking reads and writes
  - Messages may be lost or duplicated within FIFO

- Functional Model
  - Functional blocks operate concurrently
    - Single rate
    - No synchronization across processes
  - Non-blocking read, non-blocking write communication semantics

- Mapping: intersection of behaviors
  - Before mapping, nondeterministic loss and/or duplication of messages in functional model
  - After mapping, functional loss/duplication follows architecture
Finding a Compatible MoC

- Two initial options
  - “Handshaking” MoC which guarantees lossless delivery, but with latency overhead
  - “Lossy” MoC which exposes loss and duplication, but with limited functional verification capabilities
    - Point-to-point channels can lose or duplicate data
Results

- **Functional Model**
  - 14 functional processes
  - 48 signals

- **CAN controller configurations:**
  - Number of send buffers

- **Metric**
  - Message End-to-end Latency

### Graph: Message Latency sent from Supervisor ECU

- **With 1 send buffer:**
  1. Priority inversion: Message 7 < Message 1~6
  2. Average message latency = 4.936ms

- **With 2 send buffers:**
  1. No priority inversion
  2. Average message latency = 4.165ms
Automotive: Ongoing and Future Work

• Mapping Techniques for lossy MoC
  - Sensitivity criterion for message loss affects mapping decisions

• Alternative MoC that offers slightly stronger analysis capabilities
  - Guarantee that at most one message lost out of sequence of n messages

• Handshaking over unreliable network
  - Synchronous functional modeling
  - Reduce handshaking overhead based on timing analysis and/or allocation of tasks to ECUs
**Objectives:**
- Minimize the HC emissions of cold-start
- Reduce design-to-implementation controller cycle time.

**Challenges**
- Sensors not active, poor combustion, keep development cost low.

**Strategies**
- Design of AFR and HC observers, use of design of automated tools, use of modern controller design techniques

Experimental facilities
• **Goals:**
  - Minimize the HC emissions of cold-start
  - Reduce design-to-implementation controller cycle time.

• **Requirements**
  - Driveability: no noise or vibration, robustness to uncertain external conditions, low calibration effort, reliability in validation.

• **Strategies**
  - Design of AFR and HC observers, use of design of automated tools, use of modern controller design techniques.
Transmission Control

Goal:
• Improve drivability and fuel efficiency by automotive control.

Approach:
• Utilize dynamical model-based analysis and controller design.

Control Strategy:
• Multi-tiered approach to achieve shock-free gear shifting by smooth gear shifting control with engine/AT collaboration balancing between fuel economy & performance by optimal shift pattern scheduling

Prospected control structure for intelligent shifting
Objectives

• Hybrid System Analysis: study of a general semantics for simulator engines to execute hybrid system models.

• Study of representations of discontinuities and interactions between continuous-time dynamics and simultaneous discrete events

• The code generation project aims to produce application code automatically from graphical models in Ptolemy II

If an outgoing guard is true upon entering a state, because of the triggering semantics of transitions, the time spent in that state is identically zero. This state is called a "transient state."
Automotive Electronics: Driver Assistance

Product and Technology Overview

- **Ultra-Sonic**: Standard Parking
- **Long Range Radar**: ACC > 30km/h
- **Integrated Vision System**: Night Vision Support

- **Parking Space Measurement**: ACC plus 0 to ~200km/h
- **Lane Departure Warning**: Video Park Pilot

- **Semi-autonomous Parking Assistant**: Predictive Safety Systems (PSS)
Connected Drive

"New Vistas on Automotive Embedded Systems", ASV

ITR Review, Oct. 4, 2006
**Connected Car-to-Car**

**Implication on availability of a spontaneous car-to-car connectivity as part of the connected services concept**

Today, there are no vehicle functions (beyond advanced sensors) based on spontaneous car-to-car connectivity

- Opportunities in advanced / new functionalities
- Implications on architectures
- Methods and tools for the design of such systems
System Content

Software Code to Compute

\[ S_1 = g_1(E_1, E_2, \ldots, x_1, x_2, \ldots, p_1, \ldots) \]

Software Code to Compute

\[ E_1 = g_1(x_1, x_2, \ldots, p_1, \ldots) \]

Hardware

- 3x3x3, (3 mm$^3$, 3 grams, 3 $)
- Tyre compatible packaging
- Use of existing vehicle infrastructure
- No de-standardization