Electronic Spark Timing (EST) System (1 ECU)

- 2000 functions enabled by software (70-100 ECUs)

90% innovation
50-70% development cost
Why more software?

Enabling innovation easier and cheaper

Faster time to market

Decreasing development cost

http://www.cvel.clemson.edu/auto/systems/auto-systems.html
Automotive supply chain software integration

**Architecture-driven:**
- (Partially) Automated
- Early detection of errors
- Less effort/cost to change

**Document-centric:**
- Manual
- Error prone
- Costly to change

Adapted from [http://www.edibasics.hu/edi-resources/edi-by-industry/automotive.htm](http://www.edibasics.hu/edi-resources/edi-by-industry/automotive.htm)
Automotive architecture modeling

- Top-down system development i.o. bottom up
- Separation of concerns in different architectural models/views
- Model-driven i.o. document-centric approach
- Improved design quality by detecting errors early
- ...
Automotive companies and ADLs

- Automotive Modeling Language (AML)
- COnponent Language (COLA)
- EAST-ADL
- Timing Augmented Description Language (TADL)
- The ICT MAENAD project EAST-ADL2
EAST-ADL

- Advancing Traffic Efficiency and Safety through Software Technology 2 (ATESST) project
- Refined EAST-ADL2 language, profile, methodology, tools
- It provides means to represent the embedded system in several abstraction levels.
- Main source: http://www.east-adl.info/
EAST-ADL and AUTOSAR

Features of the vehicle

Abstract functions

Hardware topology, concrete functions, allocation to nodes

Software Architecture as represented by AUTOSAR

http://maenad.eu/
EAST-ADL Abstraction Levels

TechnicalFeatureModel

ExampleFeatureTree

DoorLock

BaseBrake

ABS

BrakeLight

Realization relations

FunctionalAnalysisArchitecture

LockButton

VehicleSpeedCalc

LockController

LockActuator

WheelSpeedSensor

BrakePedal

BrakeController

BrakeWheelCtrl

BrakeActuator

LockRequest

VehicleSpeed

LockActivation

WheelSpeed

PedalBrkRequest

BrakeForce

BrakeRequest
EAST-ADL Abstraction Levels

SystemModel

- DesignLevel
  - Functional Design Architecture
    - Functions
    - MW
    - Sensors/Actuators
  - Hardware Design Architecture
- ImplementationLevel
  - AUTOSAR SWC Template
  - AUTOSAR ECU Resource Template
  - AUTOSAR System Template

EnvironmentModel

Design Level

Implementation Level
Example of function-to-component Mapping

Function C2

Design Level

Function E4

Function E5

Function C1

Function E2

Function E3

Implementation Level

Function C1

Function C2

Runnable R1

Runnable R4

Runnable R3

Runnable R5

Runnable R2

ApplicationSWC A1

ApplicationSWC A2

ApplicationSWC A3

In_D : C_1

out_A : SCS1

In_A : SCS1

In_B : SCS2

out_B : SCS2

out_D : C_1

n Function to 1 SW Component
EAST-ADL Metamodel Structure

Vehicle Level
- VehicleLevel
  - TechnicalFeatureModel

Analysis Level
- AnalysisLevel
  - FunctionalAnalysisArchitecture

Design Level
- DesignLevel
  - FunctionalDesignArchitecture
  - HardwareDesignArchitecture

Implementation Level
- ImplementationLevel
  - AUTOSAR Application SW
  - AUTOSAR Basic SW
  - AUTOSAR HW

Data exchange over ports
Allocation

Extensions ...
- Requirements
- Variability
- Timing
- Dependability

TU/e
Technische Universiteit Eindhoven
University of Technology
Traceability between abstraction levels

Realization relations identify which abstract element is realized by a more concrete entity.

- Functions on analysis level realizes features on vehicle level
- Functions on design level realizes functions on analysis level
- SW components or runnables on implementation level realizes functions on design level
EAST-ADL Tooling

UML-based Tooling
- CEA Papyrus
- Integrated Eclipse application with 5 ATESTS plugins
- MagicDraw UML
- ...

DSL AUTOSAR Tooling
- MentorGraphics VSA

DSL Generic Tooling
- MetaEdit+
- TopCased

http://www.east-adl.info/Tooling.html
EAST-ADL Summary

- Defines several abstraction levels and mapping between them
- Extensions to traditional ADLs:
  - Requirements
  - Variability
  - Timing
  - Dependability
  - Safety (alignment with ISO26262)
  - Environment modeling
- Not well applied yet in automotive industry
SysML and UML

- Not required by SysML
- SysML’s extensions to UML
- UML reused by SysML (UML4SysML)
SysML Diagram Taxonomy

- Activity Diagram
- Sequence Diagram
- State Machine Diagram
- Use Case Diagram
- Block Definition Diagram
- Internal Block Diagram
- Package Diagram

- Same as UML 2
- Modified from UML 2
- New diagram type
Blocks are Basic Structural Elements

- Provides a unifying concept to describe the structure of an element or system
  - System
  - Hardware
  - Software
  - Data
  - Procedure
  - Facility
  - Person

- Multiple standard compartments can describe the block characteristics
  - Properties (parts, references, values, ports)
  - Operations
  - Constraints
  - Allocations from/to other model elements (e.g. activities)
  - Requirements the block satisfies
  - User defined compartments

<table>
<thead>
<tr>
<th>Compartment Label</th>
<th>«block» BrakeModulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>allocatedFrom</td>
<td>«activity» Modulate BrakingForce</td>
</tr>
<tr>
<td>values</td>
<td>DutyCycle: Percentage</td>
</tr>
</tbody>
</table>
Using Blocks

- Based on UML Class from UML Composite Structure
  - Supports unique features (e.g., flow ports, value properties)
- Block definition diagram describes the relationship among blocks (e.g., composition, association, specialization)
- Internal block diagram describes the internal structure of a block in terms of its properties and connectors
- Behavior can be allocated to blocks
**Block Definition vs. Usage**

**Definition**
- Block is a definition/type
- Captures properties, etc.
- Reused in multiple contexts

**Usage**
- Part is the usage of a block in the context of a composing block
- Also known as a role

---

**Block Definition Diagram**

```
<<block>>
Library:: Electronic Processor

<<block>>
Anti-Lock Controller

<<block>>
Library:: Electro-Hydraulic Valve

d1

<<block>>
Traction Detector

<<block>>
Brake Modulator
```

**Internal Block Diagram**

```
<<block>>
Anti-Lock Controller

d1 : Traction Detector

m1 : Brake Modulator
```
Internal Block Diagram (ibd) Specifies Interconnection of Parts

Enclosing Block

Connector

Item Flow

Part

Port

Internal Block Diagram Specifies Interconnection of Parts
Reference Property Explained

- S1 is a reference part*
- Shown in dashed outline box

*Actual name is reference property
 SysML Ports

• Specifies interaction points on blocks and parts
  • Integrates behavior with structure
  • portName:TypeName

• Kinds of ports
  • Standard (UML) Port
    – Specifies a set of required or provided operations and/or signals
    – Typed by a UML interface
  • Flow Port
    – Specifies what can flow in or out of block/part
    – Typed by a block, value type, or flow specification
    – Atomic, non-atomic, and conjugate variations

Standard Port and Flow Port Support Different Interface Concepts
Port Notation

**Standard Port**

- **provided interface** (provides the operations)
- **required interface** (calls the operations)

**Flow Port**

- **item flow**

---

**Flow Port**

- **part1:**
- **part2:**
State Machines

• Typically used to represent the life cycle of a block
• Support event-based behavior (generally asynchronous)
  • Transition with trigger, guard, action
  • State with entry, exit, and do-activity
  • Can include nested sequential or concurrent states
  • Can send/receive signals to communicate between blocks during state transitions, etc.

• Event types
  • Change event
  • Time event
  • Signal event
Operational States (Drive)

Transition notation: trigger[guard]/action
Adaptive Cruise Control (ACC) in SysML

Modeling the ACC system for an E-truck with a top-down approach in SysML
Requirements Diagram

The CCS must allow a driver to enable the vehicle to maintain a desired speed.

- **Engage CC**
- **D Angle CC**
- **Suspend CC**
- **Resume CC**
- **Set Desired Speed**
- **Increment Speed**
- **Decrement Speed**
- **Shift Gear**

Source: Artisan Software Tools
Use Case diagram

- Provides means for describing basic functionality in terms of usages of system by actors
- Generally elaborated via other behavioral representations to describe detailed scenarios

Source: Artisan Software Tools
System architecture
System integration

- Software
- Hardware
Running ACC_UI on Freescale board
SysML summary

- SysML provides a general purpose modeling language to support specification, analysis, design and verification of complex systems
  - Subset of UML 2 with extensions
  - 4 Pillars of SysML include modeling of requirements, behavior, structure, and parametrics

- Intended to improve communications, tool interoperability, and design quality

- Multiple tools available
  - IBM –Rhapsody
  - Sparx Systems -Enterprise Architect
  - Atego –Artisan Studio etc.
AUTOSAR (AUTomotive Open System Architecture)

- An open and standardized automotive software architecture

- Architecture
- Methodology
- Application Interfaces
AUTOSAR Milestones

- **2003**: AUTOSAR founded
- **2005**: First release
- **2006**: Basic SW complete
- **2009**: Feature enrichment
- **2011**: Derived applications
- **2012**: New development methods
- **2013**: Release 4.1.1

*Features:
- Multicore support
- Functional safety
- Ethernet
- ...*
AUTOSAR Layered Architecture

http://autosar.org/
**AUTOSAR Methodology**

**Virtual Integration**
Virtual Functional Bus - Independent of hardware

**Introduction of HW Attributes**
Holistic view of the entire system, both software and hardware

**ECU Configuration**
Separation of system into its ECUs with a common SW platform (infrastructure).

[Diagram showing the AUTOSAR methodology with components and interactions.]

[Link to AUTOSAR: http://autosar.org/]

AUTOSAR Application Interface

**Syntax** of Interfaces:
- Meta-model, Software Component Template
- Supporting transferability within the network

**Semantics** of Interfaces:
- Physical properties, units, etc.
- Supporting re-use across product lines
- In scope of AUTOSAR workpackages specifying application interfaces

http://autosar.org/
AUTOSAR Use Case

Use case ‘Front-Light Management’ in AUTOSAR

http://autosar.org/
AUTOSAR Benefits

Fully AUTOSAR compliant ECU

Scenario A
The supplier provides the ECU to a different OEM.

Scenario B
Integration of features, delivered from different sources.

Scenario C
The hardware changes.

http://autosar.org/
Automotive Standards

- **ISO 26262:**
  - Absence of unreasonable risk due to hazards caused by malfunctioning behavior of E/E systems

- **IEC 61508:**
  - Part of the overall safety related to the equipment under control (EUC) that depends on the correct functioning of the safety-related system.

- **MISRA C:**
  - Software development standard
ISO 26262 is “State of the Art” For Automotive Developed with OEM

KoenLeekens, ISO-26262 introduction, 2012
Safety in V cycle

[Diagram of the V cycle process]

4-7 System design

4-8 Item integration and testing

6-6 Specification of software safety requirements

6-11 Verification of software safety requirements

6-7 Software architectural design

6-10 Software integration and testing

6-8 Software unit design and implementation

6-9 Software unit testing

Scope of Part 4

Scope of Part 6

Scope of Part 4

Scope of Part 6
Safety Analysis in ISO 26262
MISRA C

• MISRA C is a software development standard for the C programming language developed by MISRA (Motor Industry Software Reliability Association).

• Its aims are to facilitate code safety, portability and reliability in the context of embedded systems, specifically those systems programmed in ISO C.

• As with many standards the MISRA C guideline documents are not free to users or developers.
2014 record recall year

Software problem that could cause

- the cars to **stop suddenly**
- accelerate **without warning**
- **overheats/damages** power electronics
- ...

|| YEAR | TOTAL RECALLS ISSUED | TOTAL NO. OF VEHICLES AND EQUIPMENT RECALLED IN MILLIONS |
|------|-----------------------|--------------------------------------------------------|
| 1990 | 269                   | 18.5                                                   |
| 1991 | 282                   | 14.4                                                   |
| 1992 | 217                   | 13.6                                                   |
| 1993 | 264                   | 11                                                     |
| 1994 | 290                   | 9.9                                                    |
| 1995 | 348                   | 19                                                     |
| 1996 | 341                   | 19.5                                                   |
| 1997 | 312                   | 16.7                                                   |
| 1998 | 408                   | 19.2                                                   |
| 1999 | 440                   | 55.6                                                   |
| 2000 | 626                   | 44.6                                                   |
| 2001 | 527                   | 22.4                                                   |
| 2002 | 506                   | 25.3                                                   |
| 2003 | 600                   | 22.9                                                   |
| 2004 | 698                   | 33                                                     |
| 2005 | 645                   | 20.4                                                   |
| 2006 | 613                   | 14.1                                                   |
| 2007 | 713                   | 20.6                                                   |
| 2008 | 781                   | 22.6                                                   |
| 2009 | 571                   | 18                                                      |
| 2010 | 723                   | 23                                                      |
| 2011 | 657                   | 17.5                                                   |
| 2012 | 657                   | 18.1                                                   |
| 2013 | 714                   | 27                                                      |
| 2014 YTD | 500 | **56** |

Source: National Highway Traffic Safety Administration
‘...And I’m saving even more on gas now that I’m afraid to drive it.’
Example applications:

- Power Train
- Chassis-Safety
- Body Electronics
- Infotainment

Functional domains:

- Engine
- Transmission
- Braking
- Steering
- Airbag
- Door Modules, Anti-theft
- Lighting, Wipers
- HVAC, Cluster
- Telematics, GPS
- Car Infotainment
- PND

Hybrid Innovations for Trucks (HIT) project

Safety-Critical Domain Certification

InMotion, Solar Team, “Cars in Context” TU/e projects
HIT Results

- Architecture modeling approaches
  - ADL evaluation
  - Architecture framework

- Quality framework
  - Validated metrics for Simulink
  - Metrics tool
  - Visualization
Model clones may have the effect of increasing code size and duplication of errors.
Number of duplicates affects modifiability as well.
Contribution to GMM based Editors

- Generic MetaModel
- Domain Concepts from Standard (ISO 26262) or Project
- SafeCase
- SM Editor
- Link to Metamodel Transformation
- Automatic generated
- Structured English

1. Add domain concepts
2. Link to
Summary

• In the automotive industry, more and more software and electronics system require system and software architecture methods.

• Automotive specific and generic purpose ADLs are being developed and applied.

• Many stakeholders, functionalities, safety and environment requirements require automotive specific standards.
Interesting Topics for Future Research

- System/software architecture for ITS and autonomous cars
- Safety, security mechanisms

Johan J. Lukkien, TU/e
Contact information:

Yanja Dajsuren
Tel: +31(0)402475052
Email: y.dajsuren@tue.nl
Address:
MF 6.085, Eindhoven University of Technology
5612 AZ Eindhoven, The Netherlands