Support for ISO 26262 in the EAST-ADL/AUTOSAR context

Dr. Henrik Lönn
Volvo Technology

henrik.lonn@volvo.com
Evolution of Vehicle Electronics
EAST-ADL Overview

EAST-ADL defines an Engineering information structure

- Feature content
- Functional content
- Software architecture
- Requirements
- Variability
- Safety information
- V&V Information
- Behavior

Data exchange over ports
Allocation
EAST-ADL+AUTOSAR Representation

- **Features of the vehicle**
  - Chassis
  - Steer
  - Brake
  - Cruise

- **Abstract functions**
  - BrakeAlgorithm
  - AbstractABSFrontLeft
  - VehicleSpeed

- **Hardware topology, concrete functions, allocation to nodes**
  - BrakePedal
  - BrakeFrontLeft
  - WheelSensorFrontLeft

- **Software Architecture**
  - AUTOSAR Application SW
  - SWComposition
  - VehicleSpeed
  - BaseBrake
  - SensorSWC
  - BrakePedal
  - LocalDeviceManager
  - WheelSensorFL
  - ActuatorSWC
  - Brake
  - <<Realize>>
EAST-ADL Extensions

SystemModel
- VehicleLevel
  - TechnicalFeatureModel
- AnalysisLevel
  - FunctionalAnalysisArchitecture
- DesignLevel
  - FunctionalDesignArchitecture
  - HardwareDesignArchitecture
- ImplementationLevel
  - AUTOSAR Application SW
  - AUTOSAR Basic SW
  - AUTOSAR HW

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

Data exchange over ports
Allocation
EAST-ADL Extensions

SystemModel
- VehicleLevel
  - TechnicalFeatureModel
- AnalysisLevel
  - FunctionalAnalysisArchitecture
- DesignLevel
  - FunctionalDesignArchitecture
  - HardwareDesignArchitecture
- ImplementationLevel
  - AUTOSAR Application SW
  - AUTOSAR Basic SW
  - AUTOSAR HW

Extensions ...

Environment Model

Data exchange over ports
Allocation
EAST-ADL vs AUTOSAR

EAST-ADL
For Features, Functional Architecture and Topology

AUTOSAR
For Software Architecture and Execution Platform
EAST-ADL vs AUTOSAR

- Different Abstraction Levels:
  - EAST-ADL complements AUTOSAR with “early phase” information

- Engineering Information Scope:
  - EAST-ADL complements AUTOSAR with more concepts
    - Requirements Engineering
    - Variant Management
    - Behaviour (nominal/error)
    - Timing
    - Safety

- Same Meta-Metamodel
  - Enterprise Architect model used for both
  - Same file exchange ARXML-EAXML
  - Same tool infrastructure possible ARTOP-EATOP
EAST-ADL
Related Projects
ISO 26262 reference life cycle

<table>
<thead>
<tr>
<th>1. Vocabulary</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.6 Safety management during item development</td>
</tr>
<tr>
<td>2.7 Safety management after release for production</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2. Management of functional safety</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6 Specification of the technical safety requirements</td>
</tr>
<tr>
<td>4.7 System design</td>
</tr>
<tr>
<td>4.8 Item integration and testing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3. Concept phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 Item definition</td>
</tr>
<tr>
<td>3.6 Initiation of the safety lifecycle</td>
</tr>
<tr>
<td>3.7 Hazard analysis and risk assessment</td>
</tr>
<tr>
<td>3.8 Functional safety concept</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. Product development at the system level</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5 Initiation of product development at the system level</td>
</tr>
<tr>
<td>4.7 System design</td>
</tr>
<tr>
<td>4.8 Item integration and testing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5. Product development at the hardware level</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.5 Initiation of product development at the hardware level</td>
</tr>
<tr>
<td>5.6 Specification of hardware safety requirements</td>
</tr>
<tr>
<td>5.7 Hardware design</td>
</tr>
<tr>
<td>5.8 Hardware architectural metrics</td>
</tr>
<tr>
<td>5.9 Evaluation of violations of the safety goals due to random HW failures</td>
</tr>
<tr>
<td>5.10 Hardware integration and testing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. Product development at the software level</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.5 Initiation of product development at the software level</td>
</tr>
<tr>
<td>6.6 Specification of software safety requirements</td>
</tr>
<tr>
<td>6.7 Software architectural design</td>
</tr>
<tr>
<td>6.8 Software unit design and implementation</td>
</tr>
<tr>
<td>6.9 Software unit testing</td>
</tr>
<tr>
<td>6.10 Software integration and testing</td>
</tr>
<tr>
<td>6.11 Verification of software safety requirements</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8. Supporting processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.5 Interfaces within distributed developments</td>
</tr>
<tr>
<td>8.6 Specification and management of safety requirements</td>
</tr>
<tr>
<td>8.7 Configuration management</td>
</tr>
<tr>
<td>8.8 Change management</td>
</tr>
<tr>
<td>8.9 Verification</td>
</tr>
<tr>
<td>8.10 Documentation</td>
</tr>
<tr>
<td>8.11 Confidence in the usage of software tools</td>
</tr>
<tr>
<td>8.12 Qualification of software components</td>
</tr>
<tr>
<td>8.13 Qualification of hardware components</td>
</tr>
<tr>
<td>8.14 Proven in use argument</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9. ASIL-oriented and safety-oriented analyses</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.5 Requirements decomposition with respect to ASIL tailoring</td>
</tr>
<tr>
<td>9.6 Criteria for coherence of elements</td>
</tr>
<tr>
<td>9.7 Analysis of dependent failures</td>
</tr>
<tr>
<td>9.8 Safety analyses</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>10. Guideline on ISO 26262</th>
</tr>
</thead>
</table>
Six ISO26262 Concerns

1. Concept Phase – Safety Goals
   - Risk assessment

2. Concept Phase – Functional Safety Concept
   - Topology-independent Solution

   - Preliminary System solution

4. Product Development – Hardware and Software
   - Detailed hardware and software architecture

5. Safety Element out of Context
   - Matching ASIL with ASIL

6. Supplier-OEM Exchange
   - Matching ASIL with ASIL
Safety Requirement Hierarchy

- **Hazard analysis and risk assessment**: Hazard analysis and risk assessment
  - Specification of safety goals

- **Functional safety concept**: Specification of functional safety requirements

- **Specification of technical safety requirements**: Specification of technical safety requirements

- **Hardware safety requirements**: Hardware safety requirements

- **Software safety requirements**: Software safety requirements

Hazardous situations: S, E, C

ASIL attribute

Inherited ASIL attributes
ISO 26262 - What to handle for each phase

Concept phase

- 3-7 Hazard analysis and risk assessment
  - Hazard analysis and risk assessment
  - Specification of safety goals

Product development

- 3-8 Functional safety concept
  - Specification of functional safety requirements

- 4-6 Specification of technical safety requirements
  - Specification of technical safety requirements

- 5-6 Specification of hardware safety requirements
  - Hardware safety requirements

- 6-6 Specification of software safety requirements
  - Software safety requirements

Focus on functional objectives and not technological solutions

Realization by high level architectural elements without notion of HW

Introducing HW & SW in architecture

Implementation of SW/HW
What to handle on each abstraction level

- **Vehicle Level**: Focus on functional objectives and not technological solutions.
- **Analysis Level**: Realization by high level architectural elements without notion of HW.
- **Design Level**: Introducing HW & SW in architecture.
- **Implementation Level**: Implementation of SW/HW.
- **Operational Level**:
1. Safety Goals: Vehicle Level

- Part 3.7 artifacts in EAST-ADL

<table>
<thead>
<tr>
<th>Vehicle Level</th>
<th>SystemModel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VehicleLevel</td>
</tr>
<tr>
<td></td>
<td>TechnicalFeatureModel</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Analysis Level</th>
<th>Environment Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AnalysisLevel</td>
</tr>
<tr>
<td></td>
<td>FunctionalAnalysisArchitecture</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Design Level</th>
<th>Environment Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DesignLevel</td>
</tr>
<tr>
<td></td>
<td>FunctionalDesignArchitecture</td>
</tr>
<tr>
<td></td>
<td>HardwareDesignArchitecture</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Implementation Level</th>
<th>Environment Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ImplementationLevel</td>
</tr>
<tr>
<td></td>
<td>AUTOSAR Application SW</td>
</tr>
<tr>
<td></td>
<td>AUTOSAR Basic SW</td>
</tr>
<tr>
<td></td>
<td>AUTOSAR HW</td>
</tr>
</tbody>
</table>

[Diagram showing SystemModel, AnalysisLevel, DesignLevel, ImplementationLevel with related artifacts and extensions]
Item Definition

**Dependability Requirements**
- **Item EPB**
- **Item SB**

**Vehicle Level** DemoVehicleVL

- **Technical Feature Model**
  - **Vehicle Root**
  - **Chassis**
  - **Brakes**
    - **Service Brake**
    - **Parking Brake**
  - **Cruise Control**
  - **Active Suspension**

**Requirements**
- **Requirement**
  - PB force shall be applied when parking brake function is active

**Dependability**
- **Item** ItemEPB
- **Item** ItemSB
Item Definition
Preliminary Hazard Analysis

Vehicle Feature Model
- Feature: ParkingBrake
- Feature: ServiceBrake

Dependability
- Item: ItemPB
- Item: ItemSB

Requirement
- Brake force shall be applied when brakes are activated

Safety Goal
- + EPB_Goal1
- + Brake force shall not be below 40% of driver request
- + ASIL=ASIL C
- + safeState: none

Other Components
- Hazard: SuddenLossOfBraking
- Hazardous Event:
  - + SuddenLossOfBrakinginSlope
  - + Controllability=C3
  - + Severity=S3
  - + Exposure=E4
  - + ASIL= ASIL C
- Operating Mode: BrakeActivated
- Environment Situation: Slope
- Traffic Situation: AdjacentVehicle
- Operating Situation Use Case: HighwayDriving

Non-Fulfilled Requirement
- Feature: ParkingBrake
  - Item: ItemPB
2. Functional Safety Concept: Analysis Level

- Part 3.8 artifacts in EAST-ADL
Modelling Safety Constraints

• Cover needs for Safety Requirements in ISO 26262
  • ASIL attribute
  • Allocable on architectural element

"How sure do I have to be that this safety requirement will be fulfilled”

i.e. there is a remaining risk that something unwanted of concern still happens.

What failures in what architectural elements do I want to restrict

Let’s call this a Failure (for Safety Goals we call the unwanted to avoid a Hazard)
Timing Modelling was the first constraint set defined for EAST-ADL and AUTOSAR (TIMMO project)
Safety Modelling – Basic Concept

“How sure can I be to avoid something unsafe, and where in the architecture does this apply”
Functional Safety Concept

- **TechnicalFeatureModel**
  - ServiceBrake
  - ParkingBrake

- **FunctionalAnalysisArchitecture**
  - BrakeFunction
    - BrakePedal
    - ServiceBrakeCtrl
    - BrakeGovernor
    - BrakeActuator

- **Dependability**
  - ItemServiceBrake
  - ItemParkingBrake
  - SafetyGoal
    - ASIL=C
    - ASIL=SG1
  - ASIL
    - C

- **Requirement**
  - Brake force shall not be below 40% of driver request
  - Brake command shall not deviate more than 60% from requested braking level
  - Brake request shall not deviate more than 60% from pedal command
  - BrakeActuator force shall not deviate more than 60% from requested level

- **Satisfy**
  - DeriveReq
  - RefineReq

- **SafetyConstraint**
  - ASIL=C
Functional Safety Requirement

- **BrakeActuator force shall not deviate more than 60% from requested level**

**SafetyConstraint**
- ASIL=C

**FaultFailure**
- BrakeOmission Value=Dev60%

**Requirement**
- BrakeActuator force shall not deviate more than 60% from requested level

**RefineReq**
3. Technical Safety Concept: Design Level

- Part 4 artifacts in EAST-ADL
Functional Safety Concept

**Technical Safety Concept**

**Dependability**

- Functional Safety Concept
  - Service Brake
    - Requirement
      - Brake Pedal shall not request deviating braking level
    - Derive Req
      - Technical Safety Concept
      - Service Brake
        - Requirement
          - Fault Tolerant Time Interval shall be at least 100 ms

**Functional Design Architecture**

- Brake Function
  - Pedal Sensor
    - Brake Request
  - Pedal Sensor Lo Res
    - Pedal Collector
      - Brake Request 2
      - Satisfy
      - Derive Req
        - Technical Safety Requirement
          - Brake Pedal Sensors shall be independent
      - Satisfy

**Functional Analysis Architecture**

- Brake Function
  - Brake Pedal
    - Driver PB Request
    - Satisfy
    - Park Brake Ctrl
    - Service Brake Ctrl
    - Brake Governor
      - Brake Actuator
    - Realize
4. HW & SW Requirements: Implementation Level

- Part 5 artifacts in AUTOSAR (and IP-XACT)
- Part 6 artifacts in AUTOSAR
AUTOSAR Elements

Functional Design Architecture

Brake Function

Brake Pedal -> Brake Request
Realize

Service Brake Ctrl

Brake Governor -> Brake Actuator
Satisfy

AUTOSAR

Realize

Satisfy

Dependability

Requirement
- Brake command shall not deviate more than 60% from requested braking level

Requirement
- Brake Pedal Sensors shall be independent

Requirement
- Pedal Collector Output shall not deviate more than 60% from requested level

Safety Constraint
- ASIL = C

Refine Req

Derive Req

Technical Safety Concept Service Brake

Derive Req

Safety Constraint
- ASIL = C
5. Safety Element out of Context

<table>
<thead>
<tr>
<th>Vehicle Level</th>
<th>Analysis Level</th>
<th>Design Level</th>
<th>Implementation Level</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Environment Model</strong></td>
<td><strong>SystemModel</strong></td>
<td><strong>Environment Model</strong></td>
<td><strong>SystemModel</strong></td>
</tr>
<tr>
<td><strong>Architecture</strong></td>
<td><strong>Architecture</strong></td>
<td><strong>Architecture</strong></td>
<td><strong>Architecture</strong></td>
</tr>
<tr>
<td><strong>Hazard</strong></td>
<td><strong>ASIL X</strong></td>
<td><strong>SafetyGoal</strong></td>
<td><strong>ASIL X</strong></td>
</tr>
<tr>
<td><strong>Item</strong></td>
<td><strong>FaultFailure</strong></td>
<td><strong>SafetyConstraint</strong></td>
<td><strong>ASIL X</strong></td>
</tr>
<tr>
<td><strong>ErrorModel</strong></td>
<td><strong>ASIL Y</strong></td>
<td><strong>SafetyConstraint</strong></td>
<td><strong>ASIL Y</strong></td>
</tr>
</tbody>
</table>

E.g. Technical Safety Concept without Functional Safety Concept:
Allocated Safety Constraints can play the role of Technical Safety Requirements when Functional Safety Concept is available.
6. Supplier-OEM interaction: A/D/I Level

Dependability aspects: Safety Constraints Match between subsystems

Nominal aspects: Interfaces match between subsystems
### Activities vs. Abstraction Levels

<table>
<thead>
<tr>
<th>Level</th>
<th>Activities</th>
</tr>
</thead>
</table>
| Vehicle Level | Define Features and requirements  
Identify Feature Flaw and Hazard  
Identify Scenarios and Hazardous Event  
Define Safety Goal |
| Analysis Level| Define Functional Architecture  
Define Functional Safety Requirements and Concept  
Define Error Model and Fault Failure  
Define Safety Constraints |
| Design Level  | Define Concrete Functional and Hardware Architecture  
Define Technical Safety Requirements and Concept  
Define Error Model and Fault Failure  
Define Safety Constraints |
| Implementation Level | Define Software and detailed Hardware Architecture  
Define Software and Hardware Requirements  
Define Error Model and Fault Failure  
Define Safety Constraints |
Conclusion

- EAST-ADL is a language for Automotive EE engineering information
  - Shared ontology/terminology across companies and domains
  - EAXML exchange format to secure tool interoperability
  - Allows joint efforts on methodology, modelling and tools
- Supports several aspects (timing, variability, behavior, V&V, etc. through extensions)
- EAST-ADL is aligned with AUTOSAR modelling elements and modelling infrastructure
- EATOP platform can foster tool prototyping
- EAST-ADL Association is a structure to coordinate and harmonize language progress
- **Collaborative aspect of EAST-ADL is particularly relevant for ISO26262**