Architecture Description Languages
- Representing Embedded Systems

SNART Presentation August 2009
Henrik Lönn/Cecilia Ekelin
VTEC within the Volvo Group
Outline

- Motivation
- Overview of modelling approaches
- EAST-ADL2
- MARTE
Motivation

Challenges in engineering automotive embedded systems:

- **Product Related Challenges**
  - Functionality increase
  - Complexity increase
  - Increased Safety-criticality
  - Quality concerns

- **Challenges Related to Development Process**
  - Supplier-OEM relationship
  - Multiple sites & departments
  - Product families
  - Componentization
  - Separation of application from infrastructure
  - Safety Requirements, ISO 26262
Representation Needs

- Engineering information must be precise and complete
- Separation of need and solution
- Models should distinguish between abstraction levels
  - Separation of Concerns
  - Early System Integration
- Integrated Information Handling
  - Effective Documentation management
  - Traceability
  - Tool Integration
- Model Based Development
  - Simulation
  - Analysis
  - Synthesis
A Selection of Modelling Technologies

- AADL – Aerospace domain, SW centric ADL
- Autosar – Automotive standard for SW platform and representation
- CCM – Component Model for CORBA
- EAST-ADL – Automotive ADL for System level modelling and down
- SysML – UML profile supporting generic systems engineering needs
- MARTE – UML profile for real-time embedded systems
- SAVE CCM – Software-centric automotive component model
- SPEEDS HRC – Development approach using existing standards/tools
- TADL – Formalizing timing requirement for the automotive domain
- ….. WRIGHT, ACME, EEA AIL, Titus, HRT-HOOD,
EAST-ADL2 and MARTE

- Architecture Description Language, ADL
  - Wikipedia: “computer language used to describe software and/or system architectures”
- EAST-ADL is and ADL that seek to cover the engineering information related to Automotive Electronics/Software development
- MARTE provides UML constructs to support the development of real-time and embedded systems
EAST-ADL2 Characteristics

- A system model organized in submodels on 5 abstraction levels
- The contents on an abstraction level forms a complete representation of the vehicle embedded system, with respect to the concerns of that abstraction level
- The EAST-ADL2 model is information integration: a structural representation of the vehicle EE system relying on several external tools and models (depending on company, vehicle domain, modellig purpose, etc.)
EAST-ADL Structure

- **Vehicle Level**
  - Feature content in a VehicleFeatureModel

- **Analysis Level**
  - Functional Analysis Architecture capturing the abstract functional behavior

- **Design level**
  - Hardware entities/topology
  - Concrete Functional structure & behavior
  - Function-to-ECU allocation

- **Implementation Level**
  - AUTOSAR constructs

![Diagram of EAST-ADL Structure]

**Data exchange over ports**
Principle of Realization

- Entities on lower abstraction level realizes Entities on higher abstraction level

Vehicle Level
Analysis Level
Design Level
Implementation Level
Operational Level
EE Architecture

Volvo Technology
6200, Henrik Lönn/Cecilia Ekelin
10 August 2009
Vehicle Feature Model

- A Vehicle is given by a set of Features
- Features are *stakeholder* requested functional or non-functional characteristics of a vehicle
- A Feature describes that "what", but shall not fix the "how"
- A Feature might be refined by further requirements
- From a top-down architecture approach the features are the configuration points to create a vehicle variant
Analysis Architecture

- Functional description of the EE system is found on Analysis and Design Level
  - The Analysis Architecture captures the abstract functional definition while avoiding implementation details
  - Analysis Architecture defines the system boundary
  - Realizes the functions based on the feature, the environment model, the requirements and stakeholders
  - The analysis level is purely functional, no designs or implementations are considered at this stage
  - Basis for safety analysis
Design Architecture

- The Design Architecture captures the concrete functional definition with a close correspondence with the final implementation
  - differentiation between the functionality realized by HW, middleware and SW
  - Functional Design Architecture is the functional definition of application software
  - Function-to-hardware allocation is defined
Implementation Architecture

- The Implementation Architecture represents the software-based implementation of the system
  - Functional blocks are realized by software components
  - Abstract hardware architecture is realized by hardware components
- AUTOSAR is used to capture the implementation
  - The application software architecture is defined according to AUTOSAR software component template
  - The software platform is defined in line with the AUTOSAR basic software
  - The hardware is defined according to the AUTOSAR ECU and topology elements.
Traceability between abstraction levels

- ADLRealization 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**

Data exchange over ports
Plant Model

- The plant model is visualized as a single, global entity in the system model.
- Structurally, it is contained inside each of the “architectures”.
- Although there is one occurrence in each architecture one typically use the same type definition for each
Cross-cutting aspects

• EAST-ADL supports
  – Variability
  – Requirements
  – Verification and Validation
  – Safety
  – Error propagation
Function interactions – end-to-end

- Model structure supports interaction with the environment and end-to-end functional definitions.
Hardware Design Architecture

Hardware architecture to allow hardware design and functional allocation or of HW entities defined for end-to-end function analysis.

- Analysis Level
  - Analysis Architecture
    - Functional Analysis Architecture
  - Design Architecture
    - Functional Design Architecture
    - Middleware Abstraction
    - Hardware Design Architecture
- Implementation Level
  - Implementation Architecture
    - AUTOSAR System
- Operational Level
  - Operational Architecture

Volvo Technology
6200, Henrik Lönn/Cecilia Ekelin
19 August 2009
EAST-ADL2 Complements AUTOSAR

- EAST-ADL2 is an information structure including aspects beyond the Software Architecture
  - Requirements, traceability, feature content, variability, safety, etc.
- Provides means to define what the software does
  - An AUTOSAR specification defines the software architecture and information required for SW integration - but is neutral to its functionality
- Provides means to model strategic properties
  - Key vehicle aspects is captured independently of the software architecture
- Supports modelling of error behavior and the representation of safety-related information and requirements
EAST-ADL2 – AUTOSAR Mapping
The MARTE UML2 profile

- UML profile supporting development of Real-Time and Embedded systems
- Focus on Components and Non-functional properties
- In the process of OMG standardization (V1.0 in 2009)
- Annexes in MARTE describe the mappings to EAST-ADL and AADL
- MARTE is extensive and highly complex
MARTE Characteristics

MARTE define the language constructs only!

- Common patterns, basic building blocks, standard NFP annotations
- Generic constraints that do not force specific execution models, analysis techniques or implementation technologies

It does not cover methodologies aspects:

- Interface-Based Design, Design Space Exploration
- Means to manage refinement of NFP measurement models
- Concrete processes to storage, bind, and display NFP context models
- Mapping to transform MoCCs into analysis
MARTE Contents

UML Profile with stereotypes in 3 packages:

- **Foundations** for RT/E systems modeling and analysis:
  - Core Elements
  - Non functional properties, NFPs
  - Time
  - Generic resource modeling
  - Generic component modeling
  - Allocation

- Constructs for specification and **design**:
  - RTE model of computation and communication
  - Software resource modeling
  - Hardware resource modeling

- Constructs for annotating model for **analysis**:
  - Generic quantitative analysis
  - Schedulability analysis
  - Performance analysis
1-2-3 Use of MARTE

**MARTE** is to the RTES domain as UML to the System & Software domain: a family of large and open specification formalisms!

1. In the context of a UML model
   …which may be an EAST-ADL2, AADL, AUTOSAR, etc. model…
2. Apply MARTE profile and assign MARTE stereotypes to UML elements

3. MARTE now provides RTES-relevant constructs and a semantics for a proper meaning of the model. An agreed interpretation of the model is now possible, and also the use of MARTE-compliant analysis and synthesis tools
MARTE Example

- Structural model and its sequence diagram with timing annotations
- UML model with MARTE stereotypes and graphical appearance
- MARTE events and constraints define the temporal properties
MARTE Example 2

- Allocation of application to OS to HW
Finally….  

- System Modeling is a necessary approach to master challenges of embedded systems development  
  - Complexity, Safety, Lead times, Communication, Separation of concerns, …  
- EAST-ADL is an Architecture Description Language that seeks to cover a relevant set of modelling concepts for automotive embedded systems, complementary to AUTOSAR  
- MARTE provides building blocks for UML-based modelling of Real-Time and Embedded Systems