Architecture Driven Development for Cyber-Physical Systems

Gopal Raghav, Swaminathan Gopalswamy
-Emmeskay Inc
-www.emmeskay.com
Introduction

Cyber Physical System (CPS)
- Systems consisting of computational and physical systems
- Tight coordination between the two

Behavioral model-based development (MBD) approach
- Address development of individual systems
- Inadequate for analyzing large scale system of systems

Architecture driven development (ADD) approach
- Facilitates development of large scale CPS

Focus of this presentation
Evolution of Embedded Systems

- Cyber Systems
- Logic
- Traditional / Structured S/W Development
- Cyber Physical Systems (CPS)
- Hybrid (Continuous + Logic)
- Analyze Behavior - Model Based Development
- Modern CPS
- System of Systems
- Development Process Complexity
- Ability to analyze before details are available
- Ability to manage variants
- -Architecture Driven Development (ADD)
What Next?

Major challenges to development of CPS
An architecture driven approach to development of CPS
Enabling technologies
Sample workflows illustrating the approach
Challenges to Development of CPS

- Concurrent Development
- System of Systems
- Distributed Development
- Variant Management
Interacting System of Systems

**Approach 1**
Use MBD to analyze individual CPS
Integrate CPS into larger system
**Problem**
Leads to integration challenges
Communication issues detected late

**Approach 2**
Apply MBD to system of systems
Involves details of all interacting CPS
**Problem**
Leads to scalability issues
• Difficult to apply MBD techniques to large scale systems
• Difficult to interpret results

Need to perform system analysis without involving details
**Solution:** ADD

**Auto-Flight System**
- Altitude Alert
- Flight Director
- Automatic Ground Spoilers
- Stall Warning
- Auto Pilot
- Wind Shear Protection
- Yaw Damper
- Auto Stabilizer
- Elevator Load
- Automatic Throttle System
- Auto Land

EAE Aerospace
An SAE International Group

Emmeskay Advanced Technology Solutions
Variant Management

Implementation Variants
– Technology, Level of Detail

Data Variants

Evolutionary Development
Work products are continuously changing
• Need for version control techniques
• Need to address dependencies between work products

Need to manage the different dimensions of variants
Apply select combinations to the CPS
Solution: ADD
Distributed Development

Need to breakdown systems for distributed development

System decomposition must be independent of component behavior
Solution: ADD

Globally Distributed Teams
Concurrent Development

Separation of Concerns - Multiple decomposition
Multiple views of the same system
Need to reconcile views later
Need to address integration issues
Solution: ADD
Architecture as a Solution

Our definition of architecture

- Topology of the system defining the hierarchy of subsystems, interfaces and connectivity between subsystems

Facilitates Distributed Development
Architecture Driven Modeling

- Add rich metadata
- Non-Functional analysis using architecture
  - Description
  - Schedulability
  - Safety
  - Security
  - Latency

- Functional analysis using behavioral models

Missile guidance model

Aerospace

EMMESKAY
Advanced Technology Solutions
Architecture as a Solution

Architecture provides the underpinning information

Behavioral models, data, requirements and other properties can be associated with and managed using the architecture

Constraints can be set to select combinations of the three variants

System Decomposition
- Independent of Component Behavior

Requirements Breakdown
Specification Flowdown

Implementation Variants

Data Variants

Evolutionary Variants - Versions
Architecture Views of Different “Concerns”

Multiple, Valid, Architectural views exist
Different views or perspectives are suited for different analysis/development purposes
Need mapping between the views
Multiple Views

Application Architecture View

Embedded Architecture View

Processor

Thread

Process

Snapshot from the tool IME

An SAE International Group
Architecture Driven Approach

- **Non-functional Requirements**
  - Safety, security, schedulability

- **RGA**
  - Requirements and Goals
  - Architecture - nodes can have variants

- **Functional Requirements**

- **SA**
  - System Architecture - nodes have variants

- **V&V**
  - Represent Verification and Validation Activities (V&VA)
  - Multiple perspectives with mapping between them

- **Behavioral Models**

- **Architectural Views**
  - Models generated are consistent with SA

- **Behavior of the CPS**
Technological Requirements

Architecture description methodology
- Need standardized descriptions of functional and embedded architectures
- Need to specify different components – functional, software, electronics, sensors and actuators
- Reconcile between architectural views while retaining their properties
- Need to specify variants
- Support incremental forms of analysis
  
  *SAE AADL (Architecture Analysis and Design Language)*

Architecture driven modeling
- Need to extract architectures from behavioral models and visualize them
- Need to migrate architecture descriptions into behavioral modeling domains such as Simulink
- Need to associate V&VAs with the underlying architecture and model variants
- Specify mapping and transition between multiple architectural views
  
  *IME (Integrated Modeling Environment) (Emmeskay, Inc)*
Technology Description

SAE AADL – SAE standard since 2004

- Tool neutral language to describe runtime architecture of embedded systems
- Model based analysis of embedded system architectures
  - Analysis of schedulability, safety, latency, etc
- Several component categories for software and hardware - extensible
  - System, Process, Thread, Sub-program, Data, Processor, Bus, etc.
  - Properties can be set on all components – built-in and user-defined

IME (Integrated Modeling Environment) – Commercial product

- Architecture creation, extraction, analysis and visualization
- Model and variants management, tool neutral model manipulation, model parameterization
- Enterprise wide collaboration, version control, user authentication and access control

- www.aadl.info
- http://www.emmeskay.com/tools/ime
Illustrations – Architecture Driven Workflows
Generation of Functional Models

**OSATE**

- End A_mission_guidance_system;
- System Implementation A_mission_guidance_system
  - Subcomponents
    - Airframe System T_airframe.1
    - Configuration System T_Configuration
  - Test Inputs/Results: System T_Test
  - Connections
    - Airframe_Re_3to_TeTestInput
    - Airframe_ALPHA_to_Controller a
    - Airframe_alpha_to_Controller b

**Simulink**

- Import into IME
- Visualize Architecture
- Configure Architecture
- Consistent Instantiation Tree
- Mine out Relevant Behavioral Models
- Compose Model and Export from IME
- Simulate and Author Behavioral Model

**IME**

- Visualize Architecture

**EMMESKAY**

An SAE International Group
Generation of AADL models of Embedded System from Functional Models

```plaintext
end A_missile_guidance_system;

system implementation A_missile_guidance_system

subcomponents

  Airframe: System T_Airframe.T_Control

AADL System T_Controlle TestInputs: results: system T_Te

connections

Airframe_Xe Z_e to TestInputsRe
Airframe_Altitude to TestInput
Airframe_alpha to Controller_a
Airframe_back to Controller_b

Export

Set the Processor, Process and Thread

View in different Perspective

Define Attributes: missile_guidance

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AADL_Process</td>
<td>controller_prs</td>
</tr>
<tr>
<td>AADL_Process</td>
<td>pohc</td>
</tr>
<tr>
<td>AADL_Thread</td>
<td>controller_thr</td>
</tr>
</tbody>
</table>
```
Multiple Views

Application Architecture View

Embedded Architecture View

Processor

Thread

Process

Application Software Component
Conclusions

Architecture serves as the powerful underpinning

Different aspects of CPS can be developed and integrated

Architecture is the starting point for both functional and non-functional analyses

Architecture facilitates management complexities posed by variants
Backup Slides
Goals

- Greater target accuracy
- Greater missile range
- Work in all weather conditions
- Weight and cost

System

- Home guidance system
  - Semi-active using radar detection
Missile Guidance System - SA

System

Guidance
- Generate normal accel demands
- Acquire initial target position

Targeting (Seeker/Tracker)
- Measure closing velocity between missile and target
- Estimate rate of change of slight line angle

Autopilot
- Control accel normal to the body
  - generate fin commands

Guidance Processor
Limit Acceleration

Target Acquisition
Tracker (Sightline Rate Estimator)

M1
M2

M3
M4

M5
M6