AADL Meta Model & XML/XMI

Peter Feiler
Software Engineering Institute
phf@sei.cmu.edu
Outline

• Meta Modeling Approach
• Declarative AADL Model
• AADL Instance Model
XMI/XML Based Tool Interoperability

AADL Front-end

- Textual AADL
  - Parser
  - Name Resolution
  - Semantic Checking

- Graphical AADL
  - Graphical View

- Co-resident tool
  - AADL Instance Model
    - Declarative AADL Model
    - Graphical Layout Model

- AADL XML Tool Interface
  - Persistent XML document
    - Tool-specific XML Representation
    - Convert

- Tool-specific representation
AADL Meta Model

• Defined in Eclipse Modeling Framework (EMF)
  – Collection of meta model packages with graphical views
  – Separate from, but close to UML profile of AADL

• XML as persistent storage
  – XMI specification from Ecore meta model
  – Generated XML schema

• In-core AADL model
  – Generated methods for AADL model manipulation
  – Edit history, deep copy, object editor, graphical editor
  – Methods to support
    • AADL extends hierarchy
    • feature “inheritance”
    • property value “inheritance”
AADL Meta Model Packages

- Core: defines the concepts of component type, implementation, subcomponent, AADL packages and modes.
- Component: defines the concrete classes for the different categories of components, including the constraints on their containment.
- Feature: defines the features of component types.
- Connection: defines the connections between component features.
- Flow: defines flow related elements of the AADL.
- Property: defines the elements for associating property values and for introducing new property types and properties via property sets.
Outline

• Meta Modeling Approach
• Declarative AADL Model
• AADL Instance Model
AADL Meta Model Class Hierarchy

Inheritance of meta model attributes and associations

- Programmatically available for common methods & operations

- ComponentClassifier
  - ComponentType
  - ComponentImpl
  - feature::PortGroupType

- ModeMember

- NamedElement
  - name: EString

- PropertyHolder
AADL Meta Model Fragment

Reference associations Can be cross XML document

Association labels become XML tags & Access methods
AADL Text Example

package edu::cmu::sei::XMIExample
public
  system GPS
  features
    init: in event port;
    signal: out data port GPS_Signal;
  end GPS;
  system implementation GPS.basic
  end GPS.Basic;
  data GPS_Signal
  end GPS_Signal;
end edu::cmu::sei::XMIExample;
AADL XML Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<core:AadlSpec xmi:version="2.0" ………>
  <aadlPackage name="edu::cmu::sei::XMLExample">
    <aadlPublic>
      <systemType name="GPS">
        <features>
          <eventPort name="init"/>
          <dataPort name="signal" direction="out"
            dataClassifier="//aadlPackage[@name=edu::cmu::sei::XMLExample]/aadlPublic/dataType[@name=GPS_Signal]"/>
        </features>
      </systemType>
      <systemImpl name="GPS.basic" compType="//aadlPackage[@name=edu::cmu::sei::XMLExample]/aadlPublic/systemType[@name=GPS]"/>
      <dataType name="GPS_Signal"/>
    </aadlPublic>
  </aadlPackage>
</core:AadlSpec>
```
AADL Inheritance

- Inheritance of component type features
  - By type extensions
  - By implementations
  - By subcomponents
- Inheritance of component implementation elements
  - By implementation extensions
  - By subcomponents
- Inheritance of property values
  - By type extensions
  - By implementations
  - By subcomponents
  - By contained components
  - By instance model
Port references require Context due to inheritance.
Flow Sequence Meta Model

Represents a connection reference or a pair of subcomponent/flow spec references.
Property Values As Objects
Outline

• Meta Modeling Approach
• Declarative AADL Model
• AADL Instance Model
AADL Instance Model Objectives

• Derivable from declarative AADL model
  – System implementation as root
  – Application & execution platform as subcomponents
  – Traceability to declarative model

• Self-contained compact system model
  – Compact representation
  – Separately loadable XML document
  – Semantic connections
  – Profile of locally cached property values

• Modal system instances
  – Legal mode combinations for system operation modes
  – System operation mode specific property values

• Recording of instance analysis results
AADL Instance Model

System implementation

- Application Subcomponent
- Execution Platform Subcomponent

System instance

- Application system instance
- Execution platform system instance

Process component instance

- Thread component instance

Traceability & Transparent access to declarative model

Feature instance

Connection instance
Instance Meta Model

Concrete component instance class
Category recorded as attribute

Reference to declarative model

Feature category recorded as attribute
Semantic Connections

- Semantic access connection between requires feature & accessed component
- Semantic connection between ports
- Mode transition connection between event port & transition
Instance Property Values

All instance model objects can have property associations

Cached property association indicator

Declarative model reference as reference element sequence
Modal & Configurable System Instances

Set of legal System Operation Modes
Supports SOM-specific property values

Mode instances per modal component instance
Modal Analysis Result Sets

Modal analysis results in form of sets of modal property associations

Stored as separate XML documents
Summary

• AADL Meta Model specified in Ecore
• Modular, extensible meta model specification
• XML Schema & XMI specification generated from meta model
• Declarative model & instance model
• Self-contained instance models