Tutorial Part II Objectives

- Provide overview of propose UML Profile for SAE AADL
- Review key features of UML Domain Model for SAE AADL
- Demonstrate profile AADL through use on existing avionics system
Outline

- Extending UML
  - Draft UML Domain Model for AADL
  - AADL/UML Profile
  - AADL/Example
  - Summary of Part II

AADL/UML Relationship

- Extensible AADL Annexes
- UML Working Groups
- To Be submitted to OMG for Adoption
- AADL Core
- AADL UML Profile
- UML 2.0
- UML 1.4 Detailed design
- RT Scheduling
- Performance
- Timeliness
- Safety
- Dependability
- Investigating overlap & integration
Extending UML

- UML provides modeling concepts & notations for typical software modeling projects
- Users may need
  - Additional features and/or notations
  - Non-semantic information attached to models
- UML core concepts can be extended or specialized by users
  - 3 built-in extension mechanisms
    • Stereotype
    • Constraint
    • Tagged Value
  - Can be used separately or together
- Can extend UML metamodel by explicitly adding new metaclasses & other meta-constructs
  - Depends on modeling tools or use of meta-metamodel facility

Benefits of Extending UML

- Architects can represent system architecture graphically using commonly available UML tools
- UML tool developers can add advance support for AADL to existing tools rather than developing new tools
  - e.g. safety analysis
- Software designers can take defined architecture & refine software components
  - rather than common practice of re-creating architecture in software development tools
- System integrators should have easier time integrating
  - Software components generated by UML tools, or hand-code based on UML specification
  - Executive and architectural glue code that is generated by AADL tool
  - Target hardware.
Outline

- Extending UML
- Draft UML Domain Model for AADL
- AADL/UML Profile
- AADL/Example
- Summary of Part II

UML Domain Model of AADL v0.99 (Balloted)

Overview

Model Data Types - Basic Concepts - Annexes

Core Concepts

global

OMG Model-Integrated Computing Workshop 2004, Edward Colbert & Peter Feiler

Copyright 2001-4. Absolute Software Co., Inc.

19 October 2004

OMG Model-Integrated Computing Workshop 2004, Edward Colbert & Peter Feiler

Copyright 2001-4. Absolute Software Co., Inc.

19 October 2004
UML Domain Model of AADL v0.99 (Balloted)

Components (cont.)

Component Classifier

+features

+owningClassifier

+definedFeatures

+inheritedFeatures

+properties

+definedProperties

+inheritedProperties

+definition

+flows

+definedFlows

+inheritedFlows

Feature Specification (from Features)

Extensible Element (from Extensible & Redefinable Elements)

Namespace (from Namespace)

Annex Subclause (from Annexes)

Flow (from Flows)

Property Specification (from Property Definitions)

Component Types

Component-Type

Composite-Type

Hardware-Type

System-Type

Software-Type

Processor-Type

Memory-Type

Data-Type

Subprogram Type

Device-Type

Process-Type

Thread-Type

Thread Group-Type

Copyright 2001-4. Absolute Software Co., Inc.

19 October 2004
The SAE AADL Standard: an Architecture
Analysis & Design Language for Embedded
Real-time Systems, Part 2

OMG Model-Integrated Computing Workshop 2004,
Edward Colbert & Peter Feiler

UML Domain Model of AADL v0.99 (Balloted)

Feature

Redefinable-Element
(from Extensible & Redefinable Elements)

+featureRedefinition

+redefinedFeature

Provide-Feature

Required-Subcomponent-Access

Port-Specification
(from Ports & Portgroups)

Subprogram-Specification
(from Subprogram Features)

Parameter-Specification
(from Subprogram Features)

Subprogram-Specification
(from Subprogram Features)

Property-Specification
(from Feature-Specification)

Feature-Specification

0..1
0..*
+redefinedFeature

0..1{subset redefinedElement}

+definingFeature

0..1{subset definingElement}

+definedFeaturePropertySpec

0..1{subset definedPropertySpecification, ordered}

+property

0..*{subsets appliedProperty, union, ordered}

+type

+equivalentBasicPortSpecification

0..*{subset equivalentPortSpecification, derived, ordered}

+complementaryBasicPortSpec

0..*{subset complementaryPortSpecification, derived, ordered}

Ports

Basic-Port-Specification

Data-Port-Specification

Data-Type
(from Component Types)

Event-Port-Specification

Event-Data-Port-Specification

Copyright 2001-4. Absolute Software Co., Inc.

19 October 2004
UML Domain Model of AADL v0.99 (Balloted)

Sample Constraints (Invariants)

- Component Classifier
  - a component classifier is owned by exactly 1 namespace, if it is not defined in a package then it is owned by the anonymous namespace.
  - this classifier’s category must appear in the applicableCategories of all definedPropertiesComponentType

- Component Implementation
  - name must be unique relative to other implementations of the same component type
  - category must be same as the category of the type it realizes component type
  - Subcomponents is the union of inheritedSubcomponents and defineSubcomponents
  - inheritedSubcomponents is the union of subcomponents of the Component Implementation this implementation extends and requiredComponent
  - requiredComponents has the same set of components as the requiredComponents of the type this implementation realizes

Outline

- Extending UML
- Draft UML Domain Model for AADL
- AADL/Profile
- AADL/Example
- Summary of Part II
Outline

- Extending UML
- Draft UML Domain Model for AADL
- AADL/UML Profile
- AADL/Example
- Summary of Part II
Flight Manager in AADL Graphical Notation

Flight Manager Example Using UML Profile
Classifier View: Navigation Package

Flight Plan Processing

Aircraft Performance Calculation

Integrated Navigation

Sensor Processing

Navigation

Fuel Flow

Nav sensor data

Nav signal data

20Hz

10Hz

20Hz

5Hz

2Hz

From Partitions

To Partitions

Sensor_Processing_Type

AADLPorts

Nav_Signal_Data : in
Signal_Data_Type

Nav_Sensor_Data : out
Sensor_Data_Type

AADLProperties

Period = 50ms  -- i.e. 20Hz
Dispatch_Protocol = Periodic

Integrator_Type

AADLPorts

Nav_Data: out
Integrated_Data_Type

Nav_Signal_Data: in
Sensor_Data_Type

AADLProperties

Source_Data_Size =32 B

Integrator_Type_Impl1

AADLProperties

Period = 100ms -- i.e. 10Hz
Dispatch_Protocol = Periodic

Navigation

<<AADLThread>>

Sensor_Processing_Type

AADLPorts

Nav_Signal_Data : in Signal_Data_Type

Nav_Sensor_Data : out Sensor_Data_Type

AADLProperties

Period = 50ms -- i.e. 20Hz
Dispatch_Protocol = Periodic

<<AADLData>>

Sensor_Data_Type

AADLProperties

Source_Data_Size =10 B

<<AADLData>>

Integrated_Data_Type

AADLProperties

Source_Data_Size =32 B

<<AADLData>>

Signal_Data_Type

AADLProperties

Source_Data_Size =100 B

<<AADLThread>>

Integrator_Type

AADLPorts

Nav_Data: out Integrated_Data_Type

Nav_Signal_Data: in Sensor_Data_Type

AADLProperties

<<AADLThreadImplementation>>

Integrator_Type_Impl1

AADLProperties

Period = 100ms -- i.e. 10Hz
Dispatch_Protocol = Periodic

<<AADLRealization>>
Flight Manager Using UML Profile
Classifier View: Navigation Package (Possible Alternative Representation)

```text
<<AADLThread>>
Sensor_Processing_Type
--------------------------
AADLProperties
Period = 50ms -- i.e. 20Hz
Dispatch_Protocol = Periodic

<<AADLData>>
Signal_Data_Type
--------------------------
AADLProperties
Source_Data_Size = 10 B

<<AADLThread>>
Integrator_Type
--------------------------
AADLProperties
Period = 100 -- i.e. 10Hz
Dispatch_Protocol = Periodic
```

Flight Manager Example Using UML Profile
Classifier View: Guidance Package

```text
<<AADLThread>>
Controller_Type
--------------------------
AADLFeatures
<<AADLPort>> Nav_Data: in Navigation::Integrated_Data_Type
<<AADLPort>> FP: in Flight_Management::Flight_Plan_Type
<<AADLPort>> Nav_Signal_Data: out Guidance_Data_Type

AADLProperties
Period = 50ms -- i.e. 20Hz
Dispatch_Protocol = Periodic

<<AADLData>>
Guidance_Data_Type
--------------------------
AADLProperties
Source_Data_Size = 100 B
```
Flight Manager Example Using UML Profile
Classifier View: Aircraft Performance Package

<<AADLPackage>> Aircraft_Performance

<<AADLData>> AC_Performance_Data_Type

Calculate_Performance

AADLFeatures

<<AADLParameter>> Plan : in Flight_Management::Flight_Plan_Type
<<AADLParameter>> Nav_Data : in Navigation::Integrated_Data_Type
<<AADLParameter>> Fuel_Flow : in Fuel_Management::Fuel_Data_Type
<<AADLParameter>> Performance : out AC_Performance_Data_Type

AADLSubprogramImplementation>> Calculate_Performance.F15_Implementation.

AADLProperties
Compute_Execution_Time = 25ms

<<AADLRealization>> Calculate_Performance.Harrier_Implementation.

AADLProperties
Compute_Execution_Time = 35ms

<<AADLData>> AC_Performance_Data_Type

<<AADLRealization>><<AADLRealization>>

Flight Manager Example Using UML Profile
Classifier View: Aircraft Performance Package (cont.)

<<AADLPackage>> Aircraft_Performance

<<AADLData>> Performance_Data_Type

AADLFeatures

<<AADLSubprogram>> Calculate : Calculate_Performance

AADLSubprogramImplementation>> Calculate_Performance.F15_Implementation.

AADLProperties
Source_Data_Size = 20 KB

<<AADLData>> Performance_Data_Type.Harrier_Implementation

AADLFeatures

<<AADLSubprogram>> Calculate : Calculate_Performance.Harrier_Implementation

AADLProperties
Source_Data_Size = 40 KB
Flight Manager Example Using UML Profile
Classifier View: Aircraft Performance Package (cont.)

Flight Manager Example Using UML Profile
Behavior View: Aircraft Performance Package’s Calculator_Type.Base Thread Impl.
Flight Manager Example Using UML Profile
Classifier View: Flight Management Package

Classifier View: Flight Management Package (cont.)
Generic Flight Mgr. Impl.
Flight Manager Example Using UML Profile
Classifier View: Flight Manager Implementation (cont.) – Nesting & Association

Flight Manager Example Using UML Profile
Classifier View: Flight Management Package (cont.)
F15 & Harrier Flight Mgr. Impl.
Outline

- Extending UML
- Draft UML Domain Model for AADL
- AADL/UML Profile
- AADL/Example

SAE AADL Summary

- AADL is Architecture Description Language & tools for predictable systems
  - e.g. time, reliability, fault-tolerance
  - Based on 15 years of DARPA research
- AADL provides means to:
  - Specify software & hardware architecture
  - Incrementally develop from prototype to specification
  - Analyze architecture rigorously
  - Implement final system
    - Integrating components with hardware & automatically generated system executive & glue code
  - Evolve system rapidly
    - Within development
    - Across lifecycle
- Extensible for Safety & Security analysis
SAE AADL Summary (cont.)

UML Profile

- Symbiotic Relationship
  - System Architects can graphically using commonly available UML tools to represent software & hardware architecture
    - Can make use of UML tools that provide capabilities like simulation of state models
  - UML tool developers can easily integrate AADL tools for advance analysis techniques rather than developing new tools
    - e.g. safety analysis
  - Software designers can take defined architecture & refine software components
    - Rather than common practice of re-creating architecture in software development tools
  - System integrators should have easier time integrating
    - Software components generated by UML tools, or hand-code based on UML specification
    - Executive & architectural glue code that is generated by AADL tool
    - Target hardware

Value of AADL-Based Development

- Early Prediction & Verification (Tool-Supported)
  - performance
  - reliability
  - Fault-tolerance

- Component Compliance Verification (Tool-Supported)
  - functional interface
  - resource requirements
  - system safety

- System Integration & Verification (Tool-Supported)
  - workstation testing
  - system performance
  - system safety verification
Benefits

- Model-based system engineering benefits
  - Predictable runtime characteristics addressed early & throughout life cycle
  - Greatly reduces integration and maintenance effort

Benefits of AADL as SAE standard

- AADL as standard provides confidence in
  - Language stability,
  - Broad adoption,
  - Common Definitions,
  - Strong tool support

References