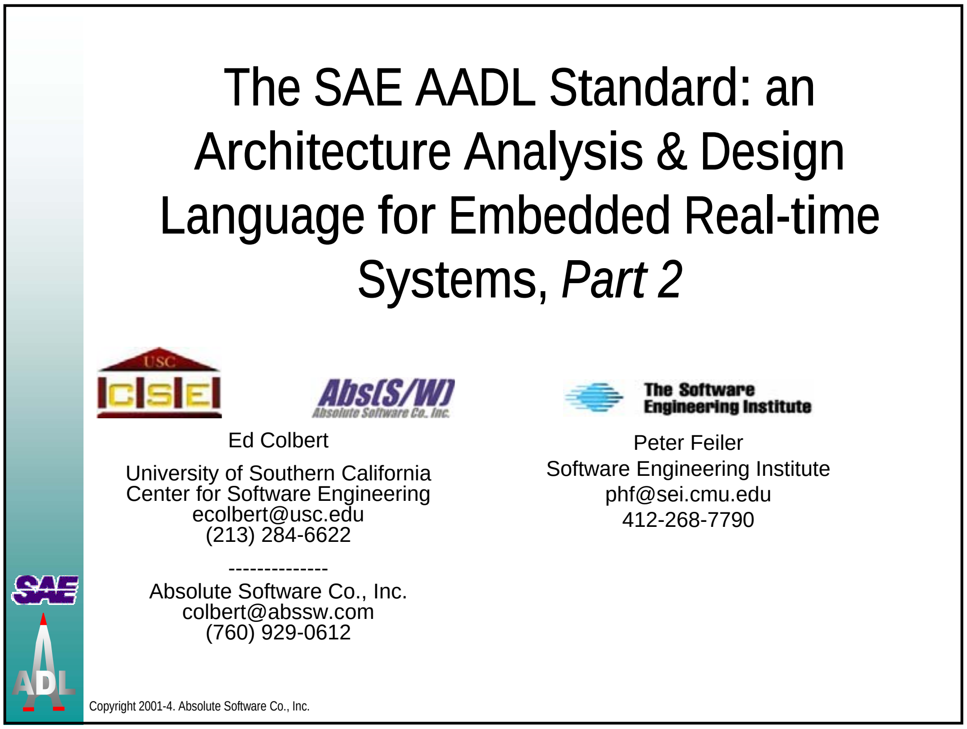


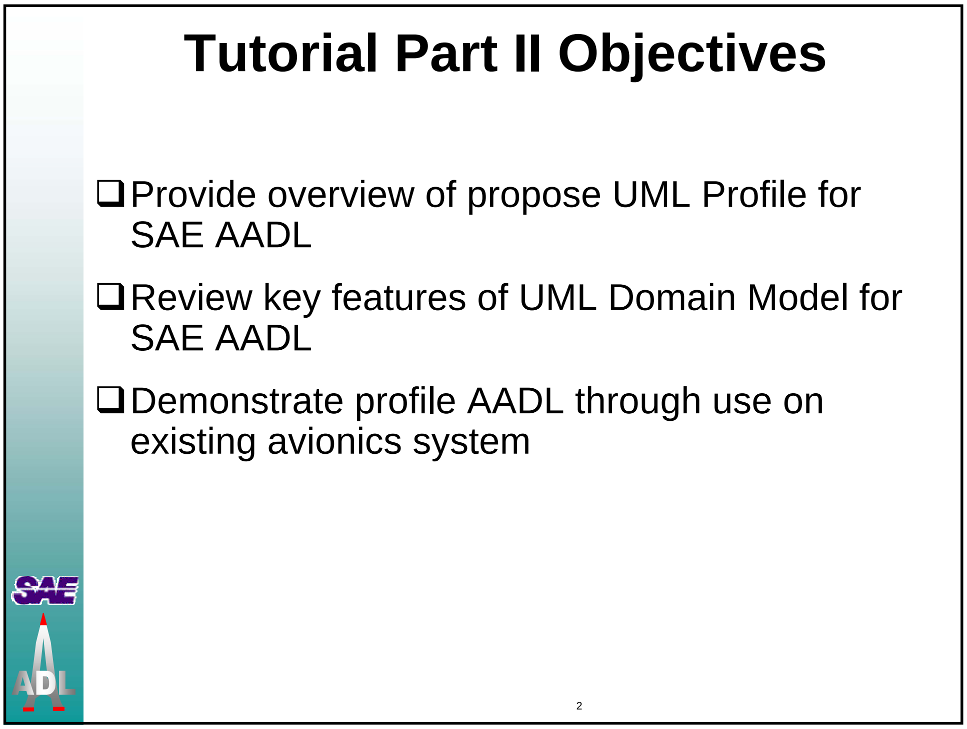
The SAE AADL Standard: an Architecture Analysis & Design Language for Embedded Real-time Systems, *Part 2*

Ed Colbert

University of Southern California
Center for Software Engineering
ecolbert@usc.edu
(213) 284-6622

Absolute Software Co., Inc.
colbert@abssw.com
(760) 929-0612

Peter Feiler
Software Engineering Institute
phf@sei.cmu.edu
412-268-7790



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

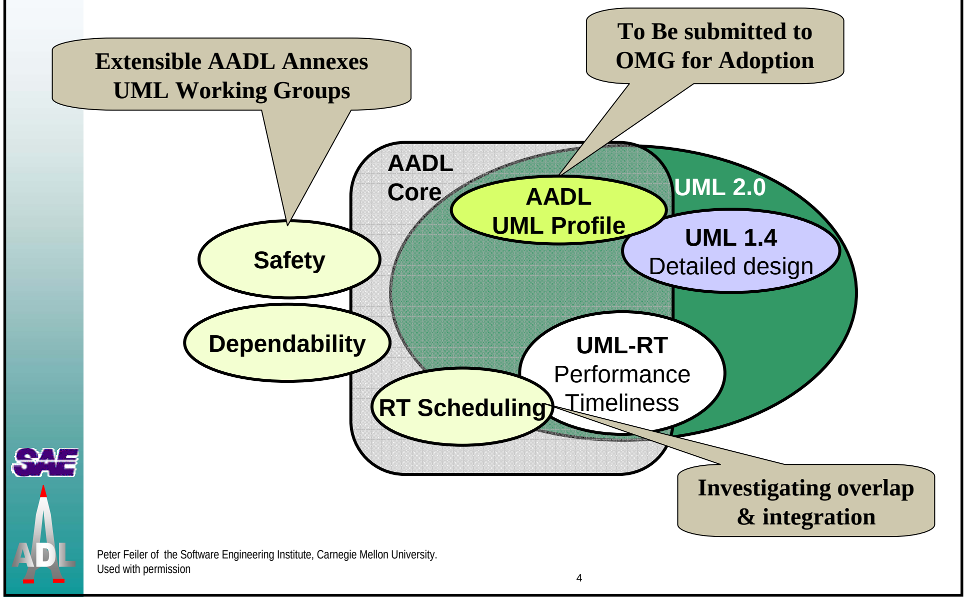
❑ Draft UML Domain Model for AADL

❑ AADL/UML Profile

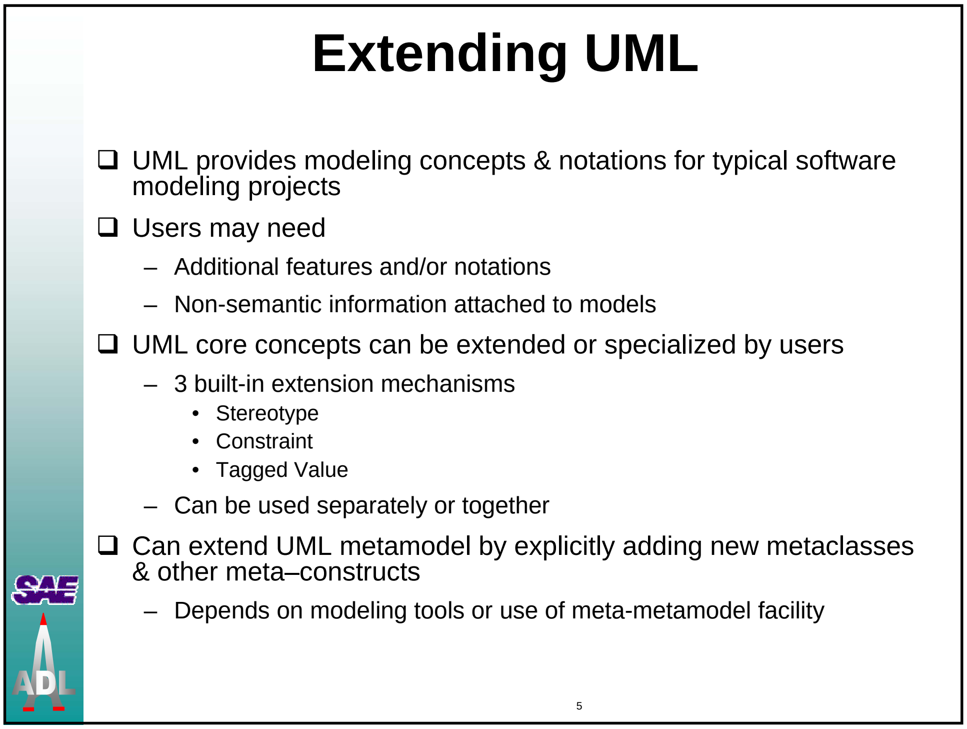
❑ AADL/Example

❑ Summary of Part II

AADL/UML Relationship



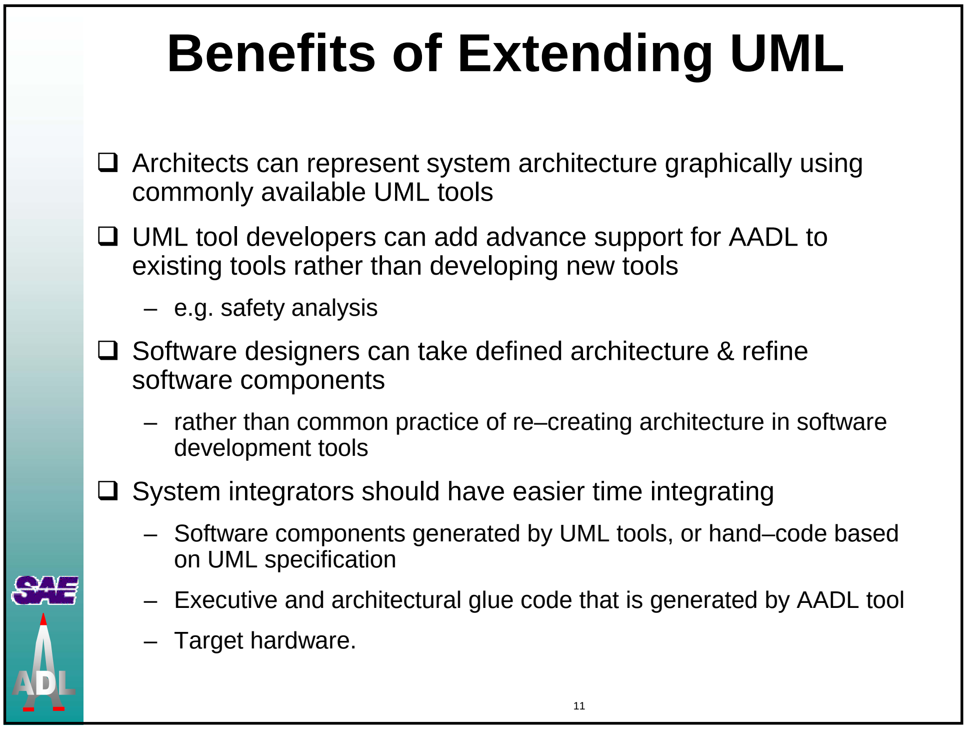
Peter Feiler of the Software Engineering Institute, Carnegie Mellon University.
Used with permission



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

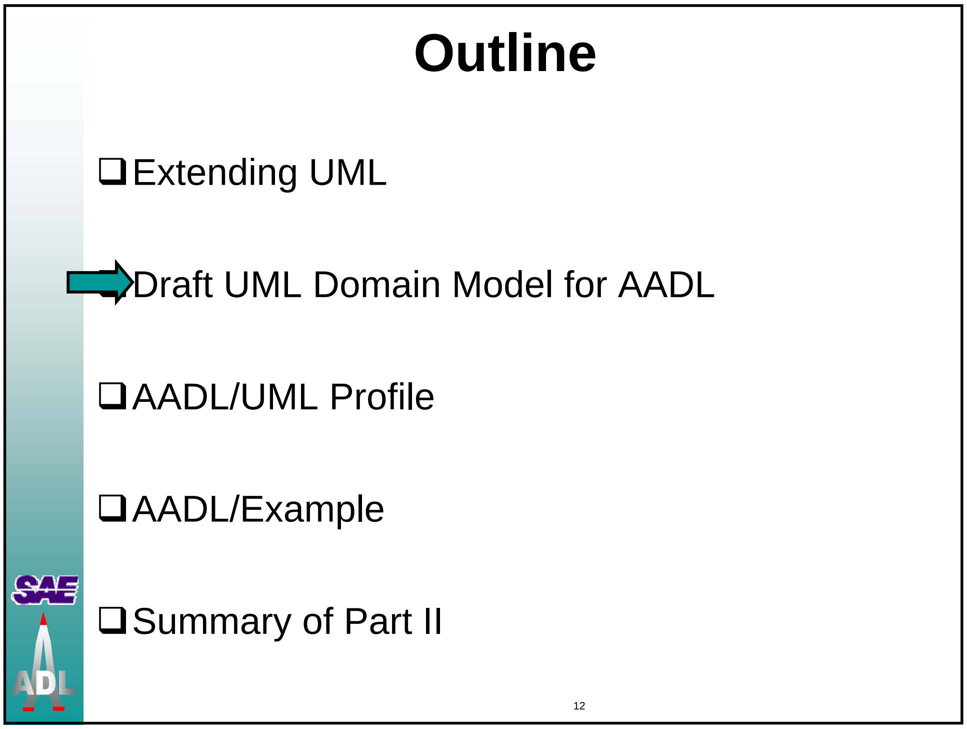
5



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.

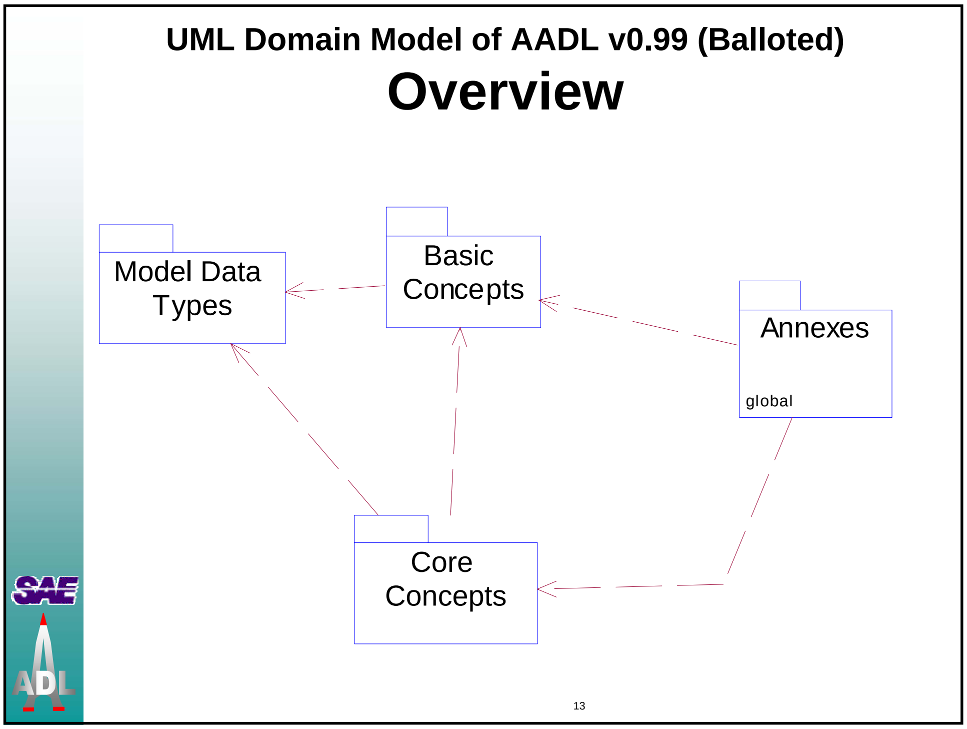
11



Outline

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

12



UML Domain Model of AADL v0.99 (Balloted)

Overview

Model Data Types

Basic Concepts

Annexes

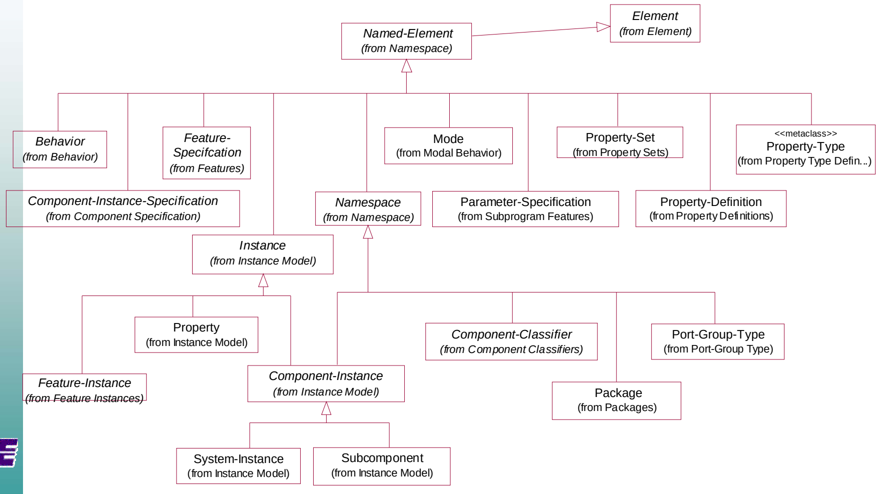
global

Core Concepts

13

UML Domain Model of AADL v0.99 (Balloted)

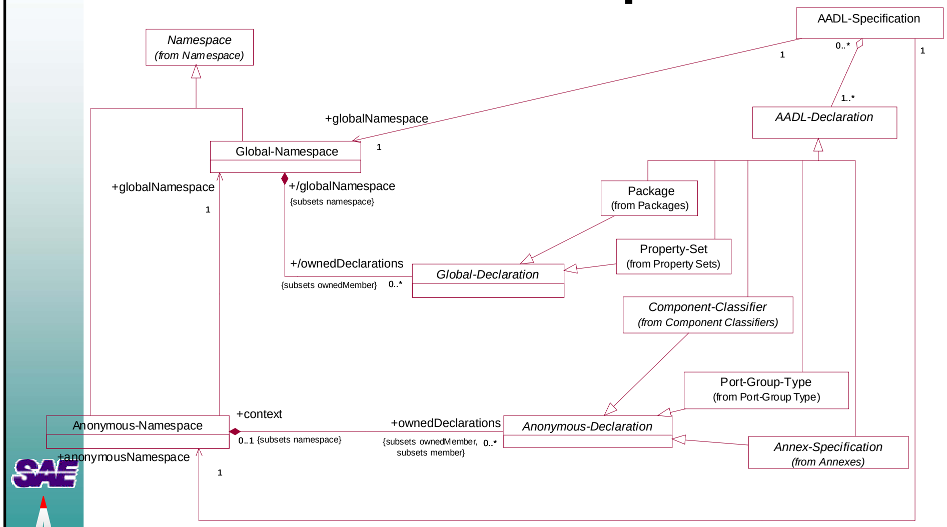
Core Concepts



14

UML Domain Model of AADL v0.99 (Balloted)

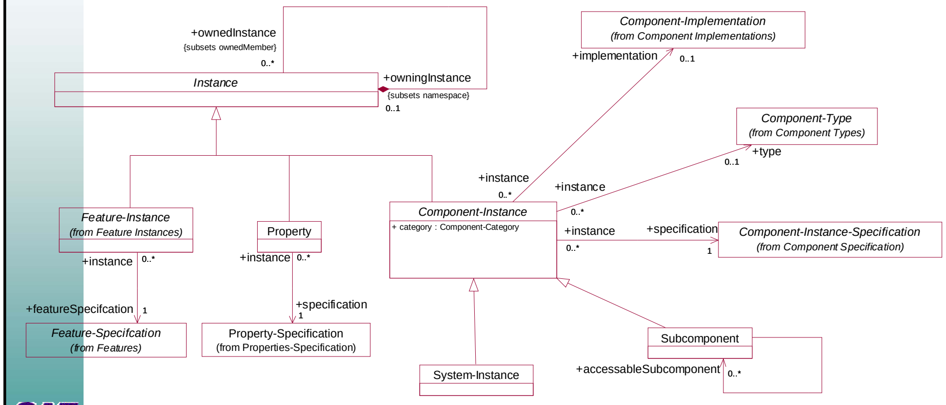
AADL Architecture Specification



15

UML Domain Model of AADL v0.99 (Balloted)

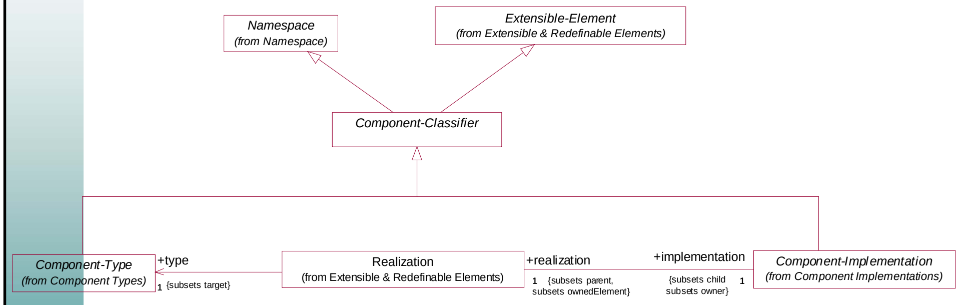
System Instance



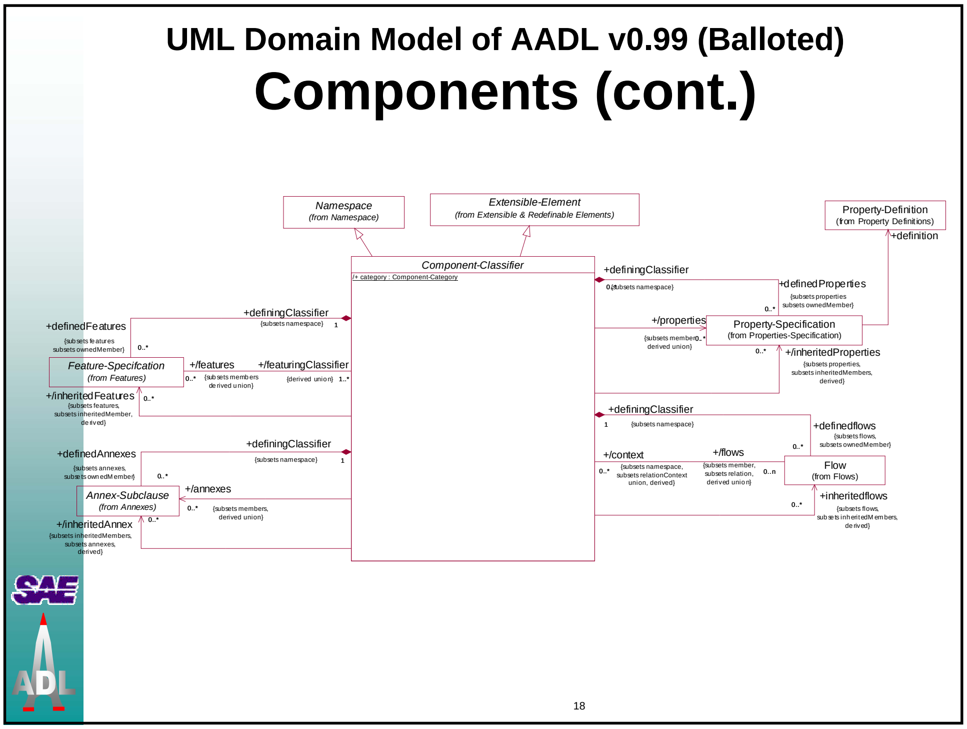
+ownedInstance
{subsets ownedMember}
0..*
Instance
+owningInstance
{subsets namespace}
0..1
Component-Implementation
(from Component Implementations)
+implementation
0..1
Component-Type
(from Component Types)
+type
0..1
+instance
0..*
+instance
0..*
Feature-Instance
(from Feature Instances)
Property
Component-Instance
+ category : Component-Category
+instance
0..*
+specification
1
Component-Instance-Specification
(from Component Specification)
+instance
0..*
+instance
0..*
+featureSpecifcation
1
+specification
1
Feature-Specifcation
(from Features)
Property-Specification
(from Properties-Specification)
Subcomponent
System-Instance
+accessableSubcomponent
0..*

UML Domain Model of AADL v0.99 (Balloted)

Components

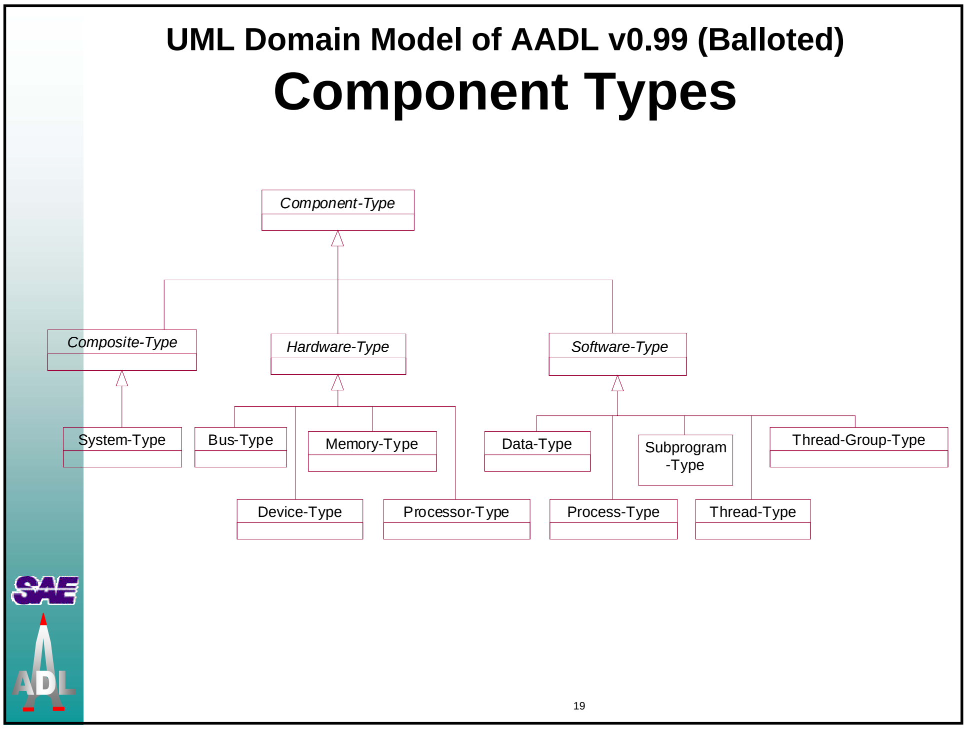


Namespace
(from Namespace)
Extensible-Element
(from Extensible & Redefinable Elements)
Component-Classifier
Component-Type
(from Component Types)
+type
1 {subsets target}
Realization
(from Extensible & Redefinable Elements)
+realization
1 {subsets parent, subsets ownedElement}
+implementation
{subsets child subsets owner} 1
Component-Implementation
(from Component Implementations)



UML Domain Model of AADL v0.99 (Balloted)

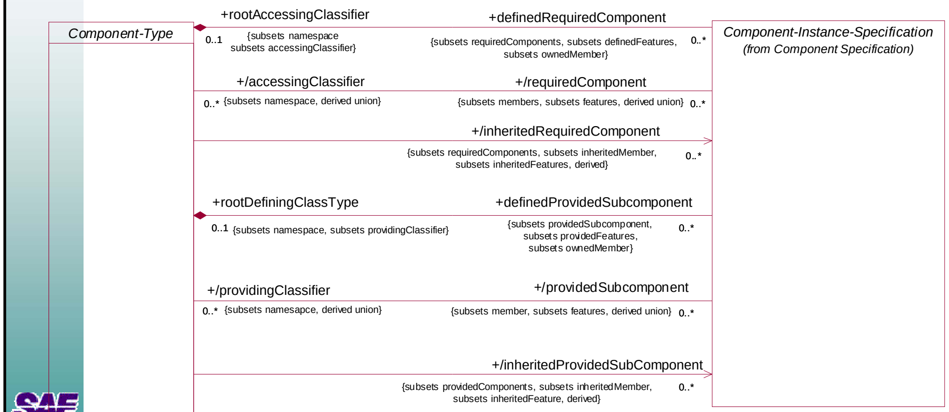
Components (cont.)



UML Domain Model of AADL v0.99 (Balloted)

Component Types

UML Domain Model of AADL v0.99 (Balloted)
Component Types (cont.)



Component-Type

Component-Instance-Specification
(from Component Specification)

+rootAccessingClassifier
0..1 {subsets namespace
subsets accessingClassifier}

+definedRequiredComponent
{subsets requiredComponents, subsets definedFeatures,
subsets ownedMember} 0..*

+/accessingClassifier
0..* {subsets namespace, derived union}

+/requiredComponent
{subsets members, subsets features, derived union} 0..*

+/inheritedRequiredComponent
{subsets requiredComponents, subsets inheritedMember,
subsets inheritedFeatures, derived} 0..*

+rootDefiningClassType
0..1 {subsets namespace, subsets providingClassifier}

+definedProvidedSubcomponent
{subsets providedSubcomponent,
subsets providedFeatures,
subsets ownedMember} 0..*

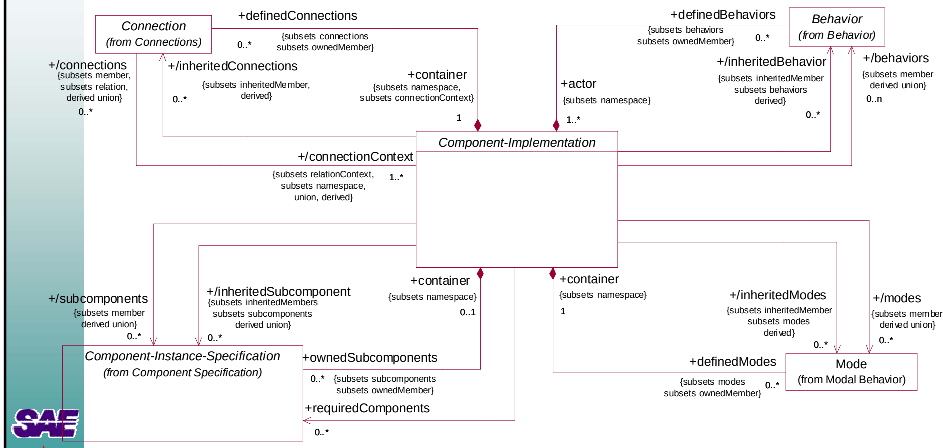
+/providingClassifier
0..* {subsets namesapce, derived union}

+/providedSubcomponent
{subsets member, subsets features, derived union} 0..*

+/inheritedProvidedSubComponent
{subsets providedComponents, subsets inheritedMember,
subsets inheritedFeature, derived} 0..*

20

UML Domain Model of AADL v0.99 (Balloted)
Component Implementation



Connection
(from Connections)

Behavior
(from Behavior)

Component-Implementation

Component-Instance-Specification
(from Component Specification)

Mode
(from Modal Behavior)

+definedConnections
0..* {subsets connections
subsets ownedMember}

+/connections
{subsets member,
subsets relation,
derived union}
0..*

+/inheritedConnections
{subsets inheritedMember,
derived}
0..*

+container
{subsets namespace,
subsets connectionContext}
1

+/connectionContext
{subsets relationContext,
subsets namespace,
union, derived} 1..*

+definedBehaviors
{subsets behaviors
subsets ownedMember} 0..*

+actor
{subsets namespace}
1..*

+/inheritedBehavior
{subsets inheritedMember
subsets behaviors
derived}
0..*

+/behaviors
{subsets member
derived union}
0..n

+/subcomponents
{subsets member
derived union}
0..*

+/inheritedSubcomponent
{subsets inheritedMembers
subsets subcomponents
derived union}
0..*

+container
{subsets namespace}
0..1

+ownedSubcomponents
0..* {subsets subcomponents
subsets ownedMember}

+requiredComponents
0..*

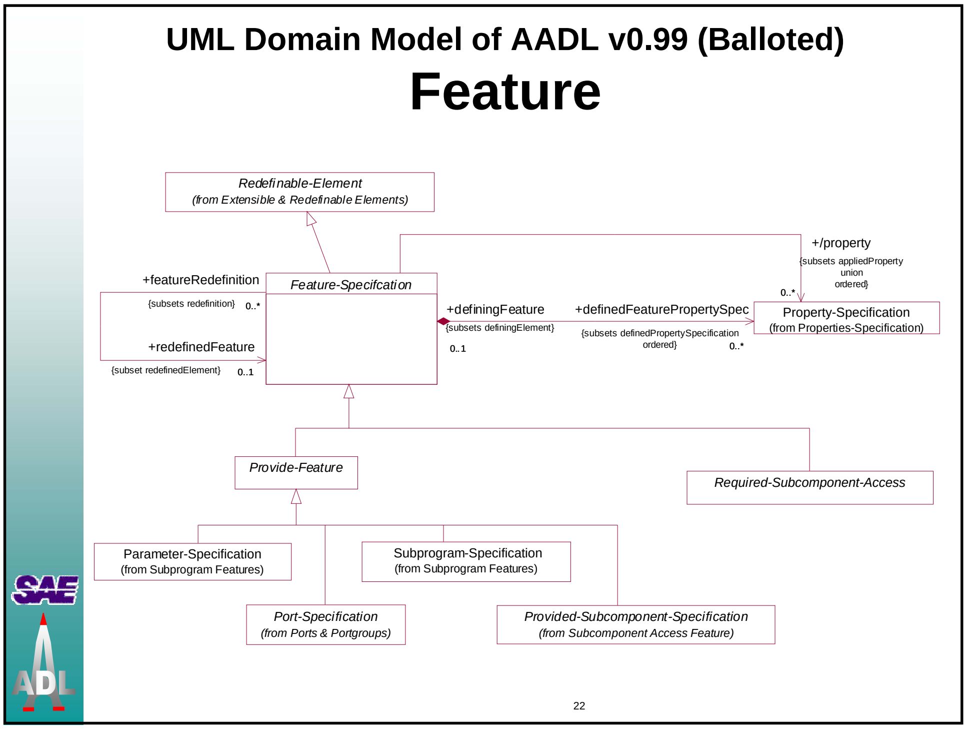
+container
{subsets namespace}
1

+/inheritedModes
{subsets inheritedMember
subsets modes
derived}
0..*

+/modes
{subsets member
derived union}
0..*

+definedModes
{subsets modes
subsets ownedMember} 0..*

21



UML Domain Model of AADL v0.99 (Balloted)

Feature

Redefinable-Element
(from Extensible & Redefinable Elements)

Feature-Specifcation

+featureRedefinition
{subsets redefinition} 0..*

+redefinedFeature
{subset redefinedElement} 0..1

+definingFeature
{subsets definingElement}
0..1

+definedFeaturePropertySpec
{subsets definedPropertySpecification
ordered} 0..*

+/property
{subsets appliedProperty
union
ordered}
0..*

Property-Specification
(from Properties-Specification)

Provide-Feature

Required-Subcomponent-Access

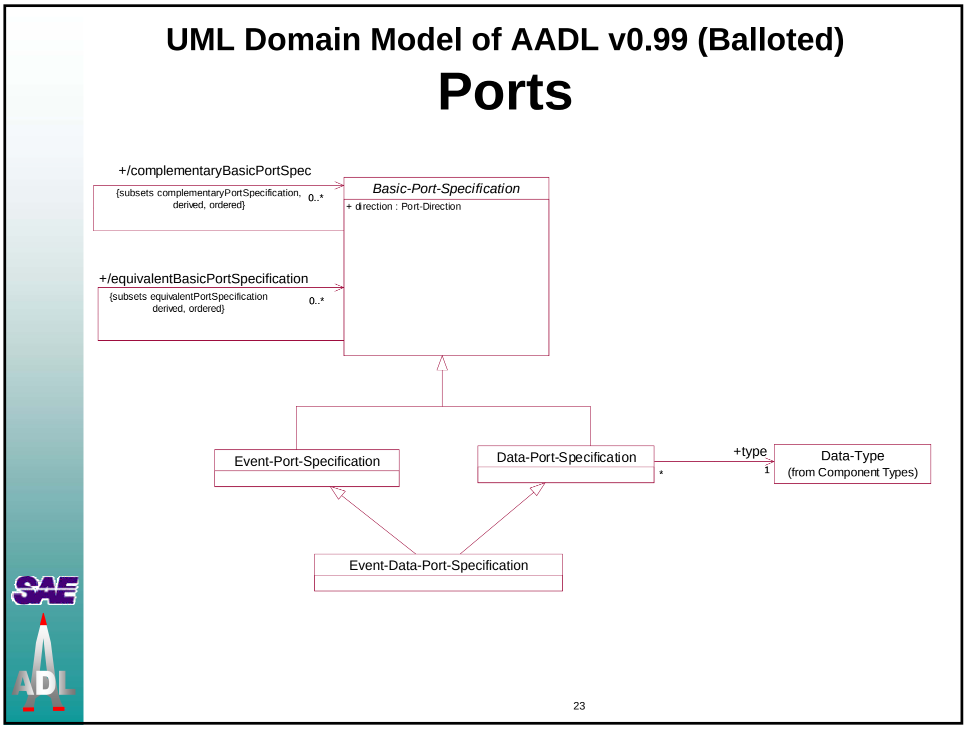
Parameter-Specification
(from Subprogram Features)

Subprogram-Specification
(from Subprogram Features)

Port-Specification
(from Ports & Portgroups)

Provided-Subcomponent-Specification
(from Subcomponent Access Feature)

22



UML Domain Model of AADL v0.99 (Balloted)

Ports

+/complementaryBasicPortSpec
{subsets complementaryPortSpecification,
derived, ordered} 0..*

Basic-Port-Specification
+ direction : Port-Direction

+/equivalentBasicPortSpecification
{subsets equivalentPortSpecification
derived, ordered} 0..*

Event-Port-Specification

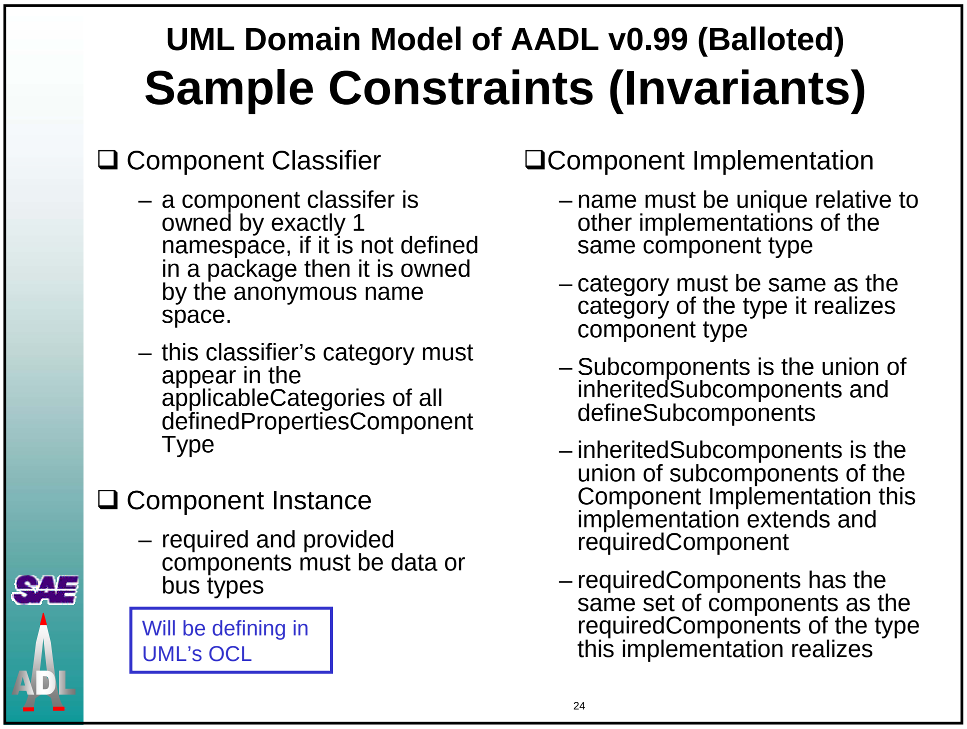
Data-Port-Specification

+type
* 1

Data-Type
(from Component Types)

Event-Data-Port-Specification

23



UML Domain Model of AADL v0.99 (Balloted)

Sample Constraints (Invariants)

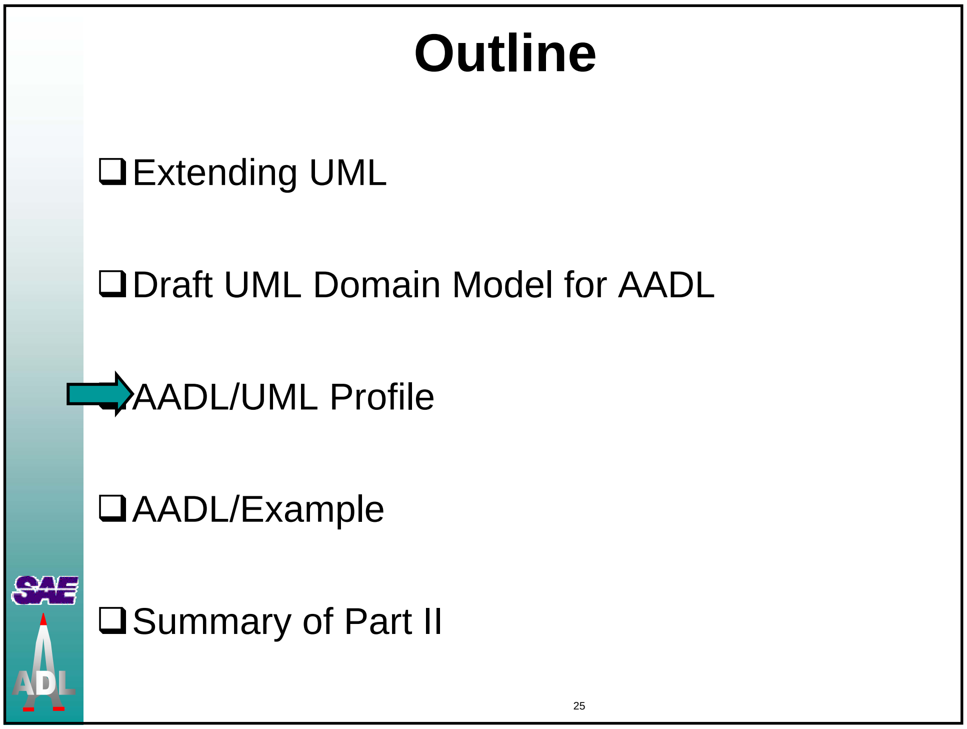
❑ Component Classifier
- a component classifer is owned by exactly 1 namespace, if it is not defined in a package then it is owned by the anonymous name space.
- this classifier's category must appear in the applicableCategories of all definedPropertiesComponent Type

❑ Component Instance
- required and provided components must be data or bus types

Will be defining in UML's OCL

❑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

24



Outline

❑ Extending UML

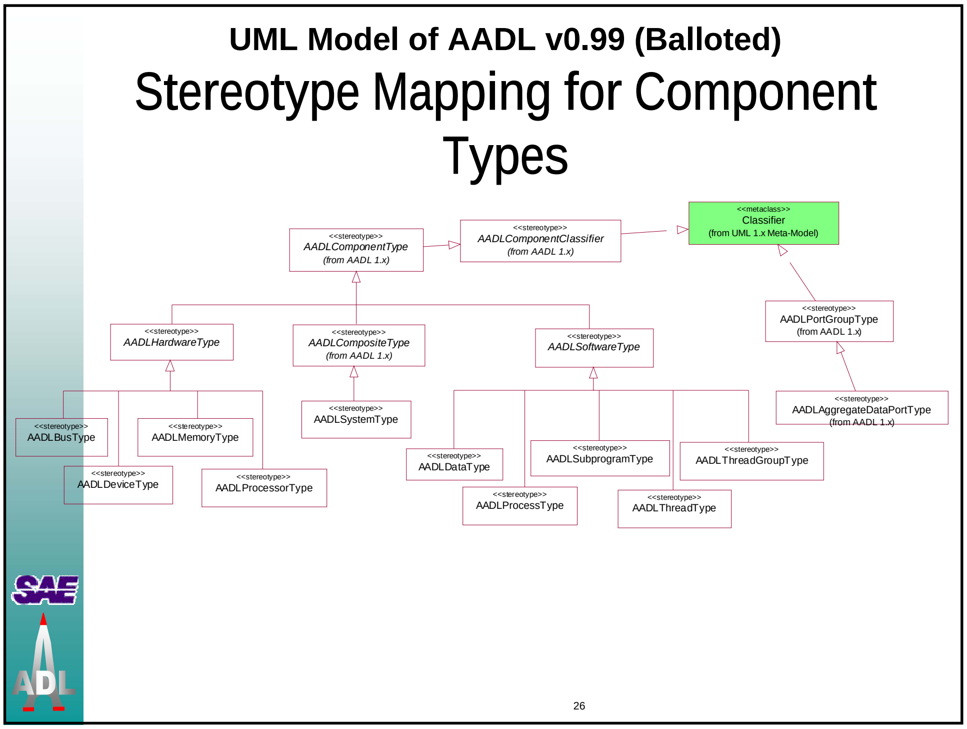
❑ Draft UML Domain Model for AADL

❑ AADL/UML Profile

❑ AADL/Example

❑ Summary of Part II

25



UML Model of AADL v0.99 (Balloted)

Stereotype Mapping for Component Types

<<metaclass>>
Classifier
(from UML 1.x Meta-Model)

<<stereotype>>
AADLComponentClassifier
(from AADL 1.x)

<<stereotype>>
AADLComponentType
(from AADL 1.x)

<<stereotype>>
AADLHardwareType

<<stereotype>>
AADLCompositeType
(from AADL 1.x)

<<stereotype>>
AADLSoftwareType

<<stereotype>>
AADLPortGroupType
(from AADL 1.x)

<<stereotype>>
AADLAggregateDataPortType
(from AADL 1.x)

<<stereotype>>
AADLBusType

<<stereotype>>
AADLMemoryType

<<stereotype>>
AADLDeviceType

<<stereotype>>
AADLProcessorType

<<stereotype>>
AADLSystemType

<<stereotype>>
AADLDataType

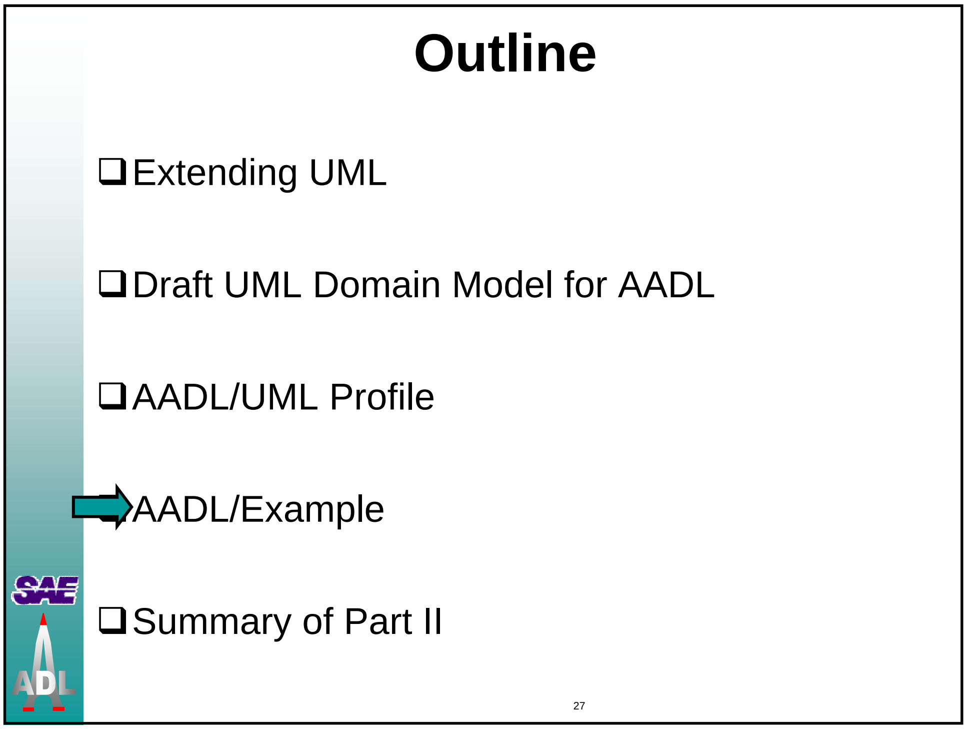
<<stereotype>>
AADLSubprogramType

<<stereotype>>
AADLThreadGroupType

<<stereotype>>
AADLProcessType

<<stereotype>>
AADLThreadType

26

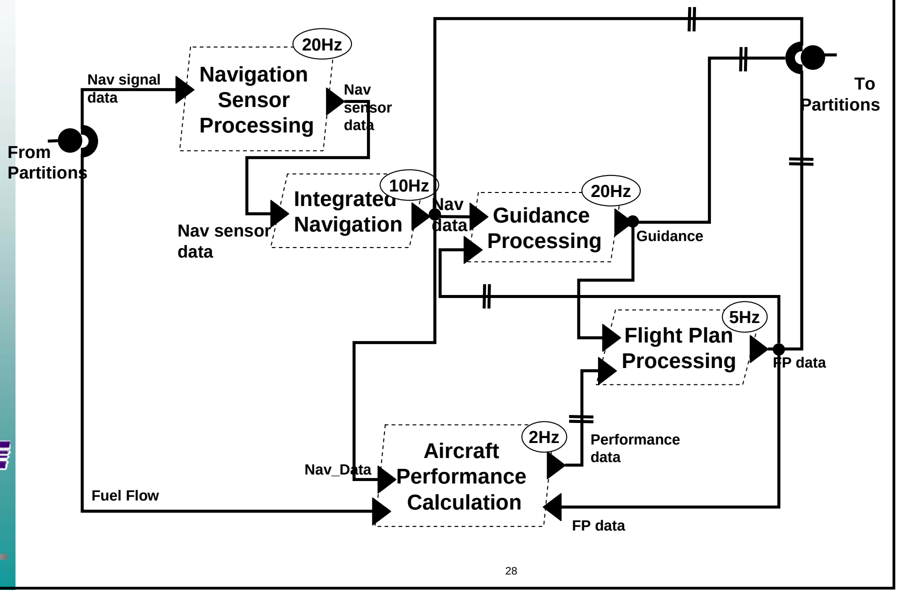


Outline

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

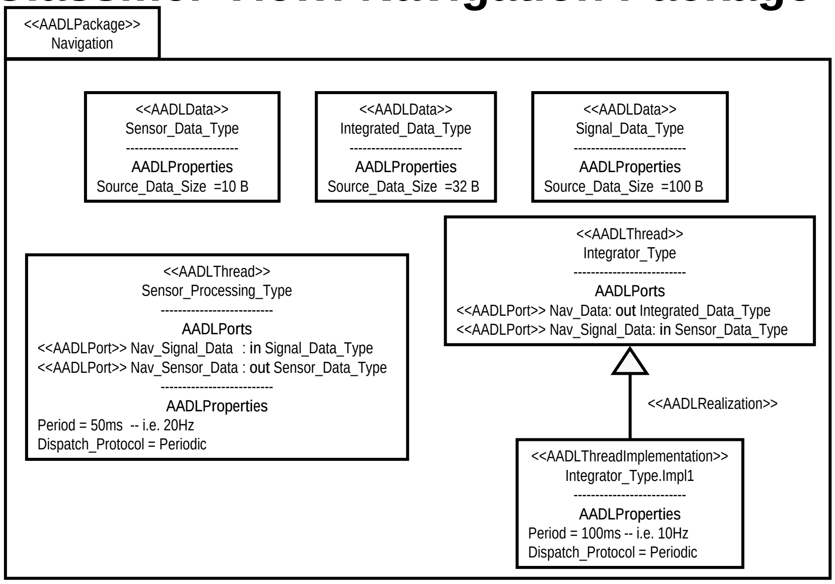
27

Flight Manager in AADL Graphical Notation



28

Flight Manager Example Using UML Profile
Classifier View: Navigation Package



<<AADLPackage>>
Navigation

<<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>>
Sensor_Processing_Type

AADLPorts
<<AADLPort>> Nav_Signal_Data : in Signal_Data_Type
<<AADLPort>> Nav_Sensor_Data : out Sensor_Data_Type

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

<<AADLThread>>
Integrator_Type

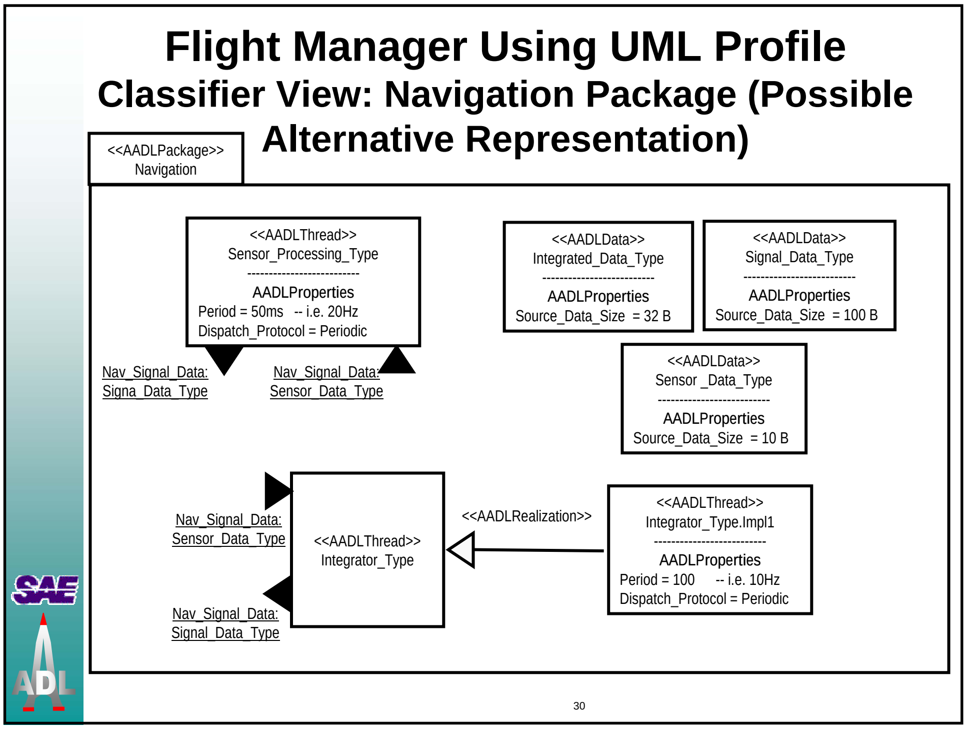
AADLPorts
<<AADLPort>> Nav_Data: out Integrated_Data_Type
<<AADLPort>> Nav_Signal_Data: in Sensor_Data_Type

<<AADLRealization>>

<<AADLThreadImplementation>>
Integrator_Type.Impl1

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

29



Flight Manager Using UML Profile

Classifier View: Navigation Package (Possible Alternative Representation)

<<AADLPackage>>
Navigation

<<AADLThread>>
Sensor_Processing_Type

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

Nav_Signal_Data: Signa_Data_Type

Nav_Signal_Data: Sensor_Data_Type

<<AADLData>>
Integrated_Data_Type

AADLProperties
Source_Data_Size = 32 B

<<AADLData>>
Signal_Data_Type

AADLProperties
Source_Data_Size = 100 B

<<AADLData>>
Sensor _Data_Type

AADLProperties
Source_Data_Size = 10 B

Nav_Signal_Data: Sensor_Data_Type

<<AADLThread>>
Integrator_Type

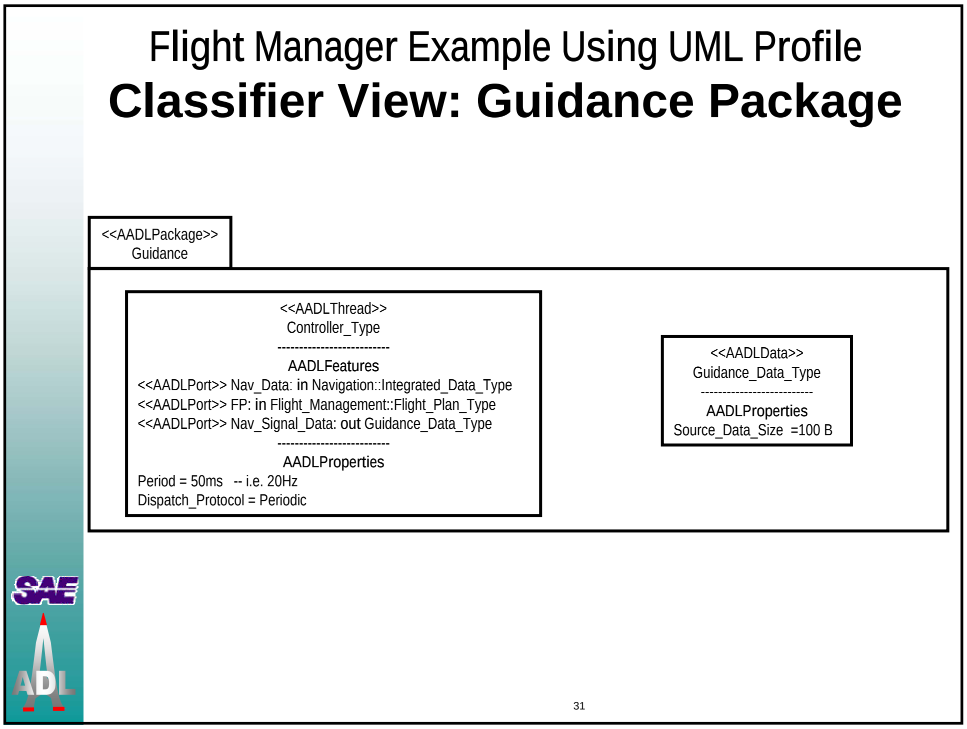
Nav_Signal_Data: Signal_Data_Type

<<AADLRealization>>

<<AADLThread>>
Integrator_Type.Impl1

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

30



Flight Manager Example Using UML Profile

Classifier View: Guidance Package

<<AADLPackage>>
Guidance

<<AADLThread>>
Controller_Type

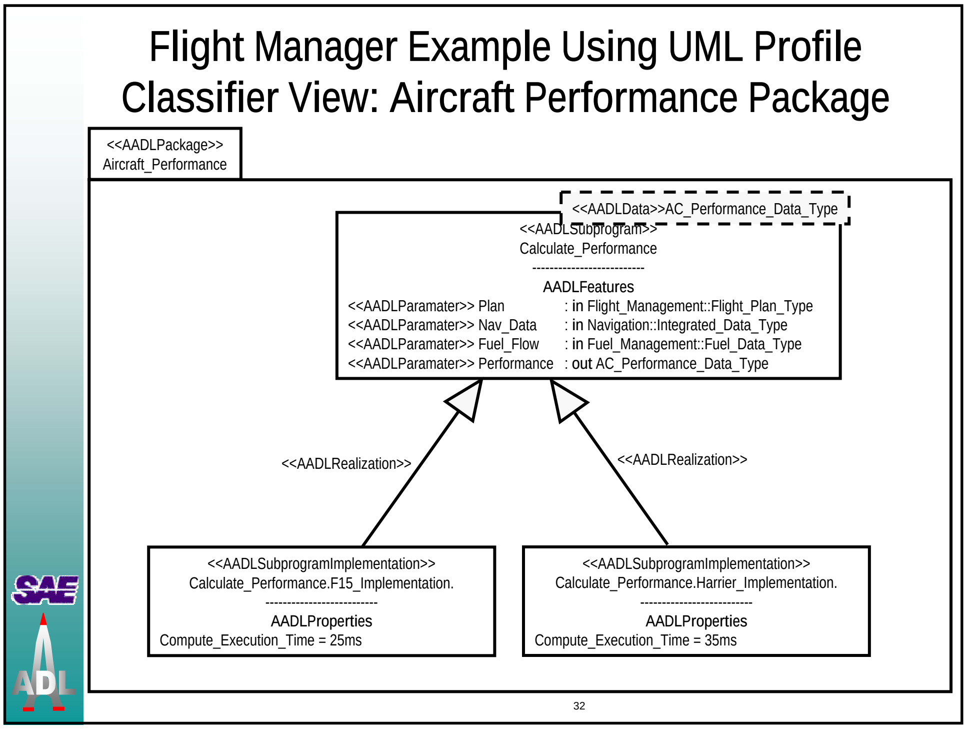
AADLFeatures
<<AADLPort>> Nav_Data: **in** Navigation::Integrated_Data_Type
<<AADLPort>> FP: **in** Flight_Management::Flight_Plan_Type
<<AADLPort>> Nav_Signal_Data: **out** Guidance_Data_Type

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

<<AADLData>>
Guidance_Data_Type

AADLProperties
Source_Data_Size =100 B

31



Flight Manager Example Using UML Profile Classifier View: Aircraft Performance Package

<<AADLPackage>>
Aircraft_Performance

<<AADLData>>AC_Performance_Data_Type

<<AADLSubprogram>>
Calculate_Performance

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

<<AADLRealization>>

<<AADLRealization>>

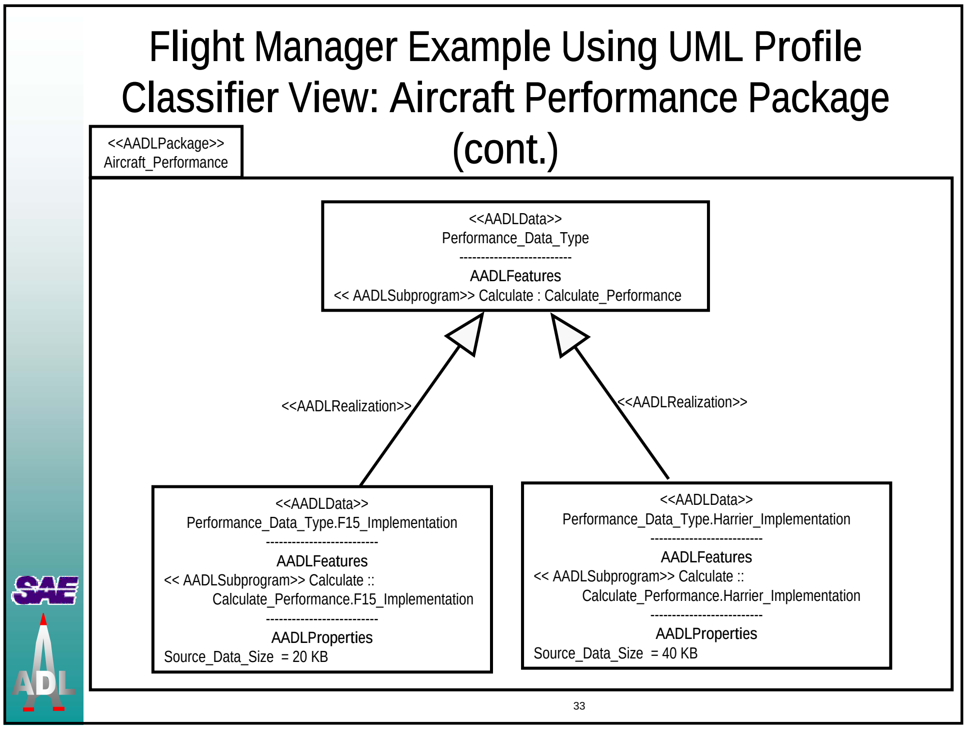
<<AADLSubprogramImplementation>>
Calculate_Performance.F15_Implementation.

AADLProperties
Compute_Execution_Time = 25ms

<<AADLSubprogramImplementation>>
Calculate_Performance.Harrier_Implementation.

AADLProperties
Compute_Execution_Time = 35ms

32



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

<<AADLPackage>>
Aircraft_Performance

<<AADLData>>
Performance_Data_Type

AADLFeatures
<< AADLSubprogram>> Calculate : Calculate_Performance

<<AADLRealization>>

<<AADLRealization>>

<<AADLData>>
Performance_Data_Type.F15_Implementation

AADLFeatures
<< AADLSubprogram>> Calculate ::
Calculate_Performance.F15_Implementation

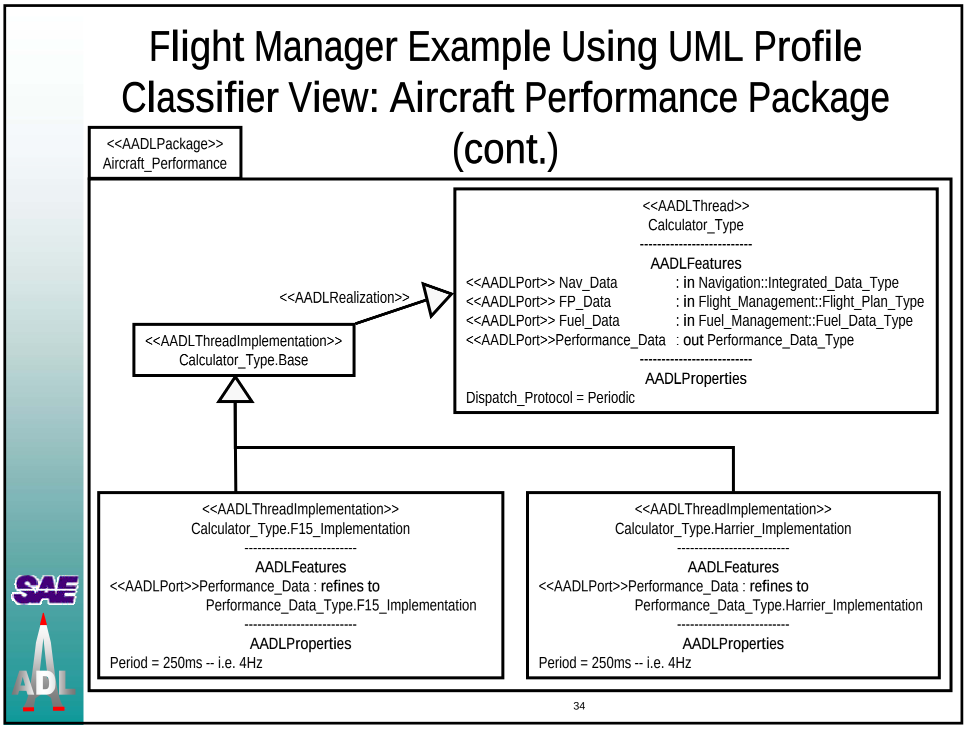
AADLProperties
Source_Data_Size = 20 KB

<<AADLData>>
Performance_Data_Type.Harrier_Implementation

AADLFeatures
<< AADLSubprogram>> Calculate ::
Calculate_Performance.Harrier_Implementation

AADLProperties
Source_Data_Size = 40 KB

33



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

<<AADLPackage>>
Aircraft_Performance

<<AADLThread>>
Calculator_Type

AADLFeatures
<<AADLPort>> Nav_Data : in Navigation::Integrated_Data_Type
<<AADLPort>> FP_Data : in Flight_Management::Flight_Plan_Type
<<AADLPort>> Fuel_Data : in Fuel_Management::Fuel_Data_Type
<<AADLPort>>Performance_Data : out Performance_Data_Type

AADLProperties
Dispatch_Protocol = Periodic

<<AADLRealization>>

<<AADLThreadImplementation>>
Calculator_Type.Base

<<AADLThreadImplementation>>
Calculator_Type.F15_Implementation

AADLFeatures
<<AADLPort>>Performance_Data : refines to
Performance_Data_Type.F15_Implementation

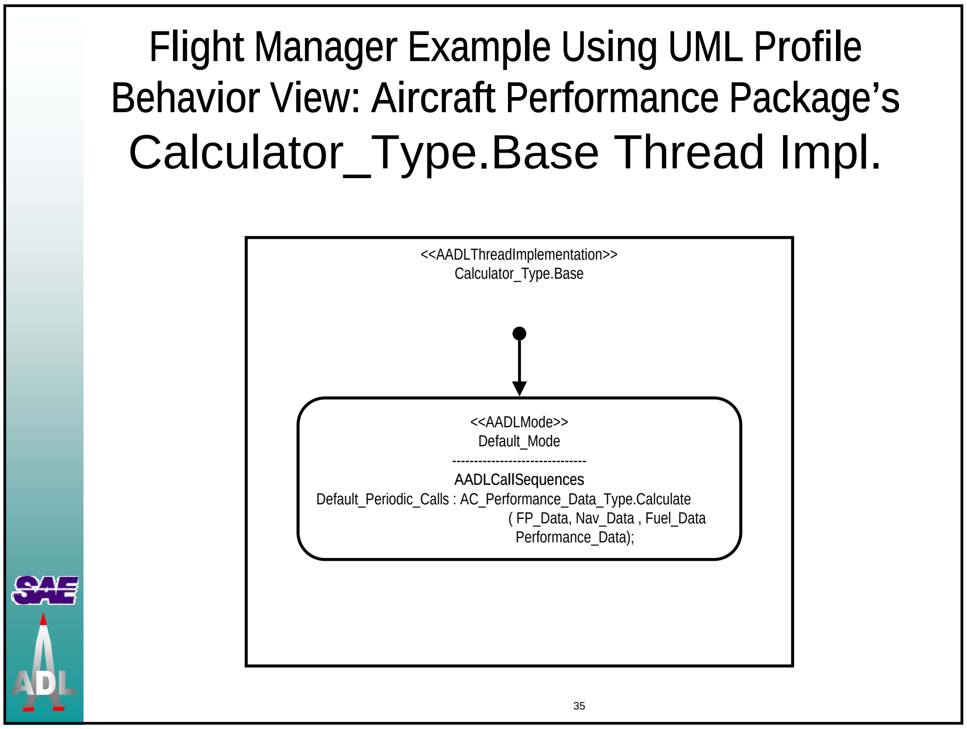
AADLProperties
Period = 250ms -- i.e. 4Hz

<<AADLThreadImplementation>>
Calculator_Type.Harrier_Implementation

AADLFeatures
<<AADLPort>>Performance_Data : refines to
Performance_Data_Type.Harrier_Implementation

AADLProperties
Period = 250ms -- i.e. 4Hz

34



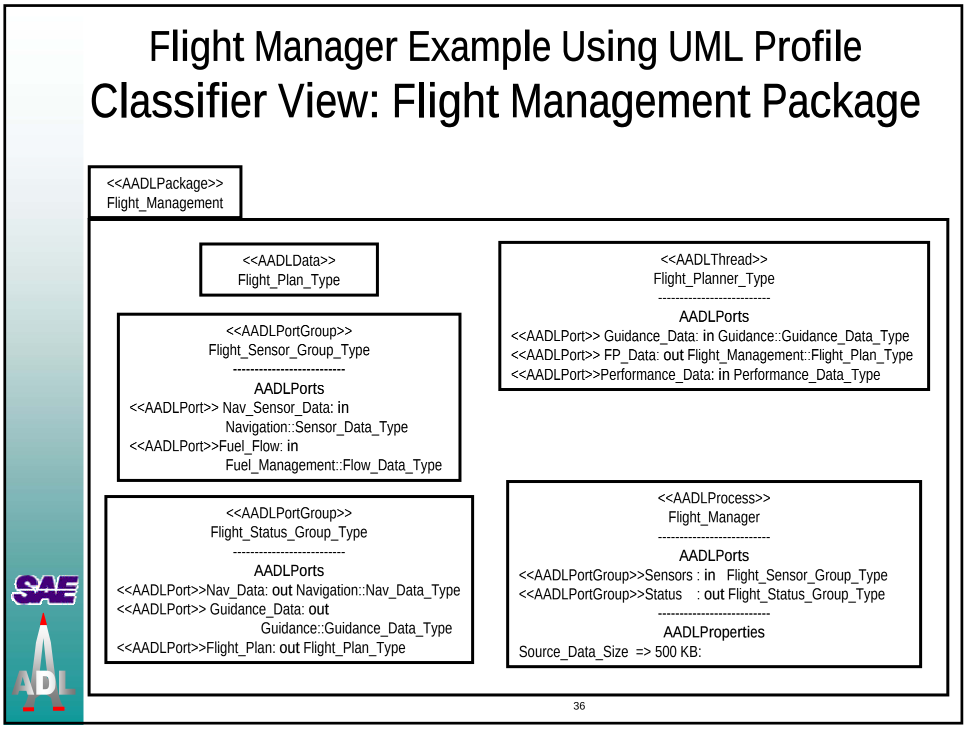
Flight Manager Example Using UML Profile Behavior View: Aircraft Performance Package's Calculator_Type.Base Thread Impl.

<<AADLThreadImplementation>>
Calculator_Type.Base

<<AADLMode>>
Default_Mode

AADLCallSequences
Default_Periodic_Calls : AC_Performance_Data_Type.Calculate
(FP_Data, Nav_Data , Fuel_Data
Performance_Data);

35



Flight Manager Example Using UML Profile
Classifier View: Flight Management Package

<<AADLPackage>>
Flight_Management

<<AADLData>>
Flight_Plan_Type

<<AADLPortGroup>>
Flight_Sensor_Group_Type

AADLPorts
<<AADLPort>> Nav_Sensor_Data: **in** Navigation::Sensor_Data_Type
<<AADLPort>>Fuel_Flow: **in** Fuel_Management::Flow_Data_Type

<<AADLThread>>
Flight_Planner_Type

AADLPorts
<<AADLPort>> Guidance_Data: **in** Guidance::Guidance_Data_Type
<<AADLPort>> FP_Data: **out** Flight_Management::Flight_Plan_Type
<<AADLPort>>Performance_Data: **in** Performance_Data_Type

<<AADLPortGroup>>
Flight_Status_Group_Type

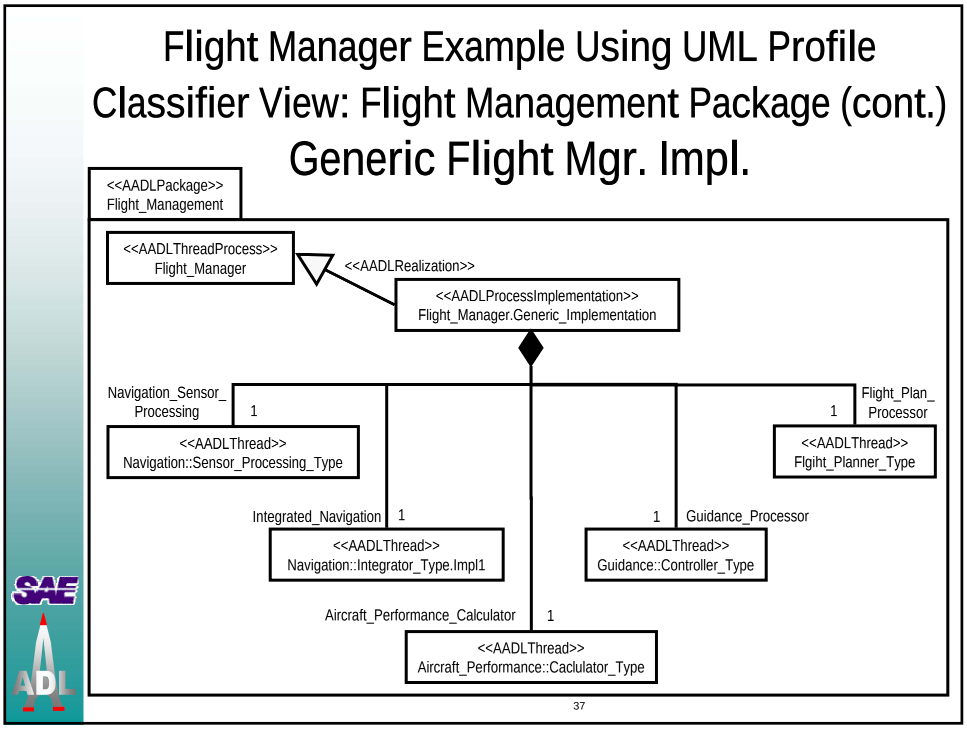
AADLPorts
<<AADLPort>>Nav_Data: **out** Navigation::Nav_Data_Type
<<AADLPort>> Guidance_Data: **out** Guidance::Guidance_Data_Type
<<AADLPort>>Flight_Plan: **out** Flight_Plan_Type

<<AADLProcess>>
Flight_Manager

AADLPorts
<<AADLPortGroup>>Sensors : **in** Flight_Sensor_Group_Type
<<AADLPortGroup>>Status : **out** Flight_Status_Group_Type

AADLProperties
Source_Data_Size => 500 KB:

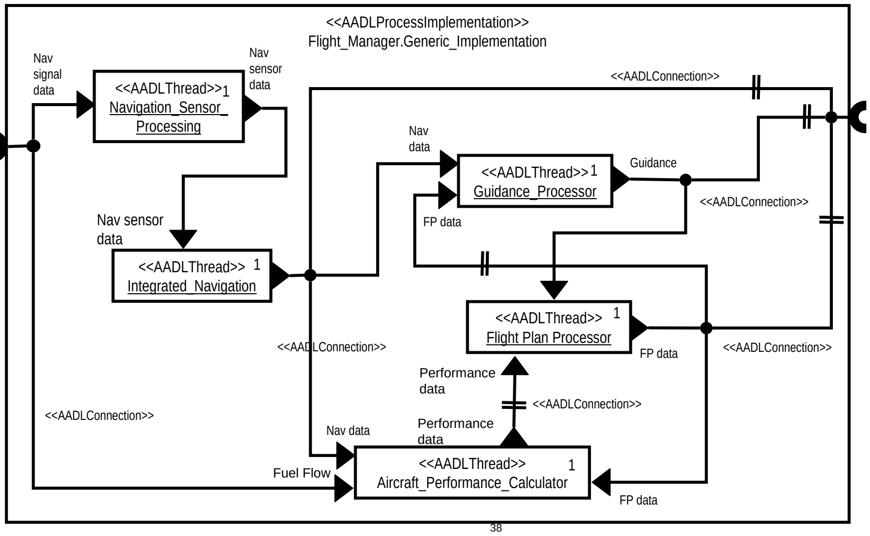
36



Flight Manager Example Using UML Profile
Classifier View: Flight Management Package (cont.)
Generic Flight Mgr. Impl.

37

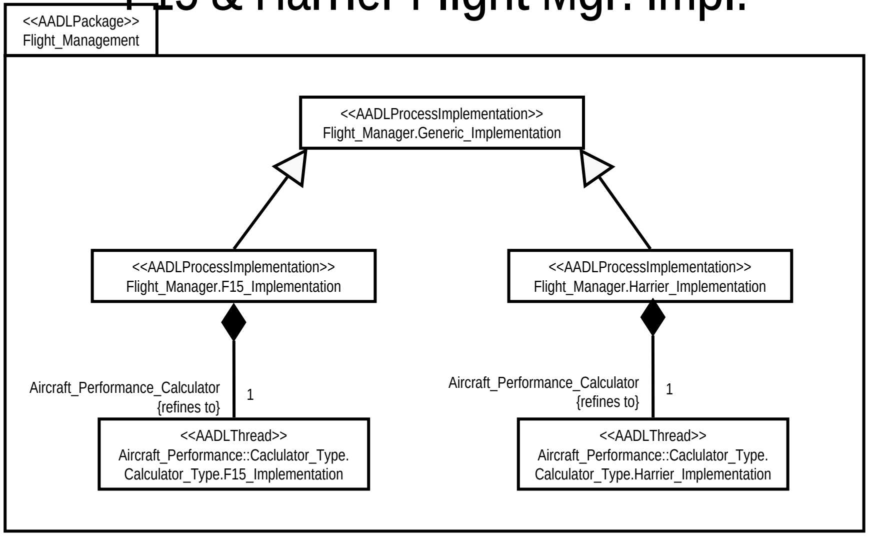
Flight Manager Example Using UML Profile Classifier View: Flight Manager Implementation (cont.) – Nesting & Association



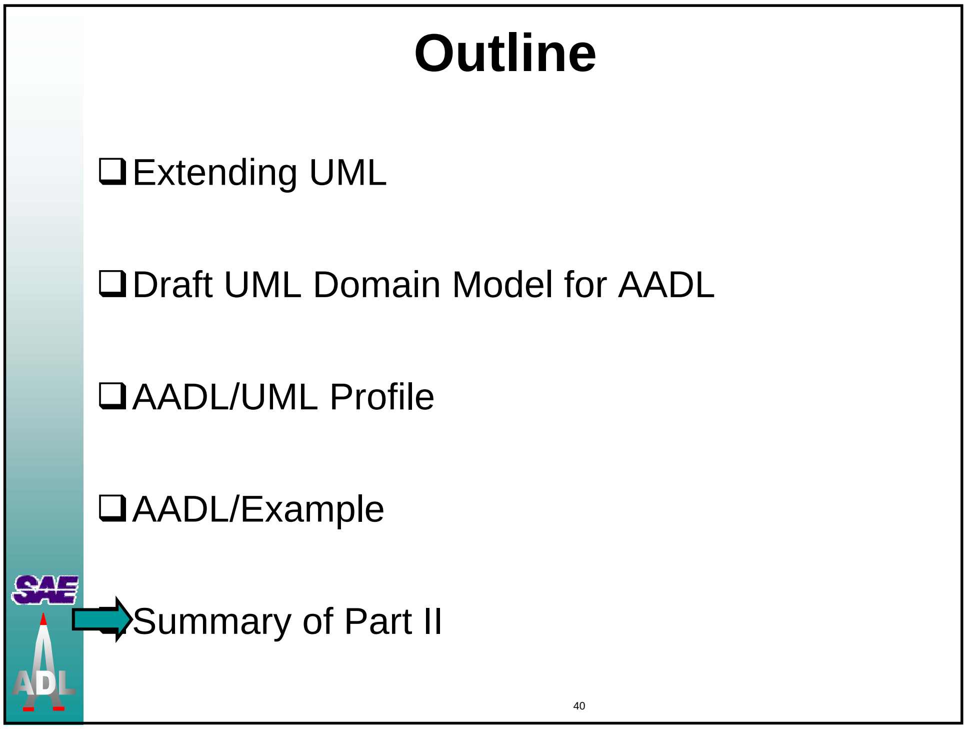
38

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

F15 & Harrier Flight Mgr. Impl.

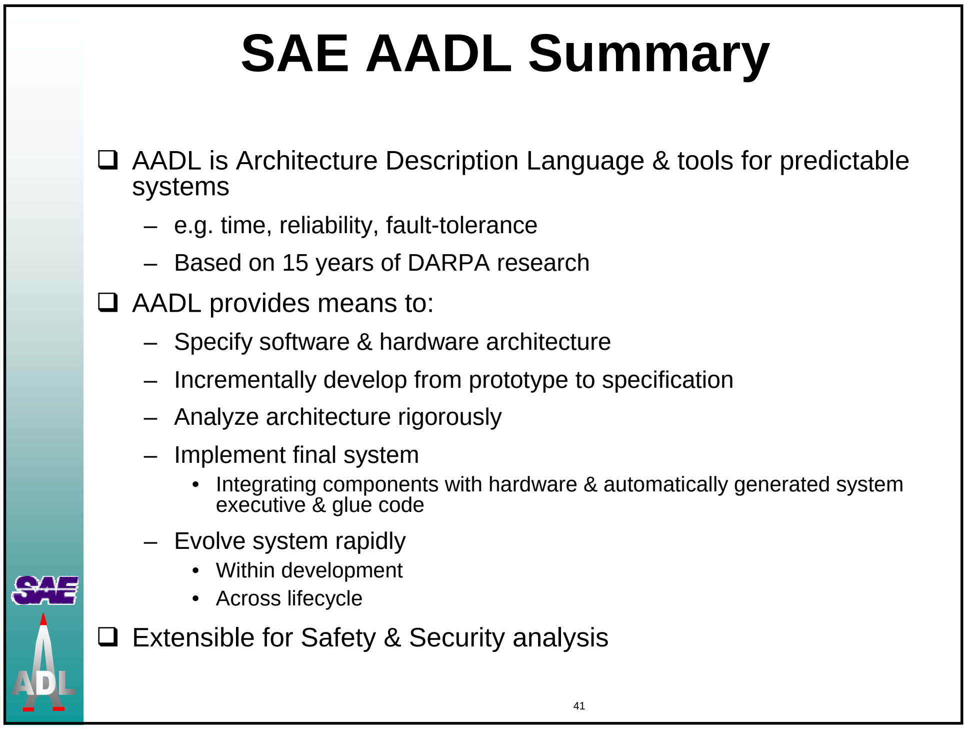

<<AADLPackage>>
Flight_Management
<<AADLProcessImplementation>>
Flight_Manager.Generic_Implementation
<<AADLProcessImplementation>>
Flight_Manager.F15_Implementation
<<AADLProcessImplementation>>
Flight_Manager.Harrier_Implementation
Aircraft_Performance_Calculator
{refines to}
1
Aircraft_Performance_Calculator
{refines to}
1
<<AADLThread>>
Aircraft_Performance::Caclulator_Type.
Calculator_Type.F15_Implementation
<<AADLThread>>
Aircraft_Performance::Caclulator_Type.
Calculator_Type.Harrier_Implementation

39



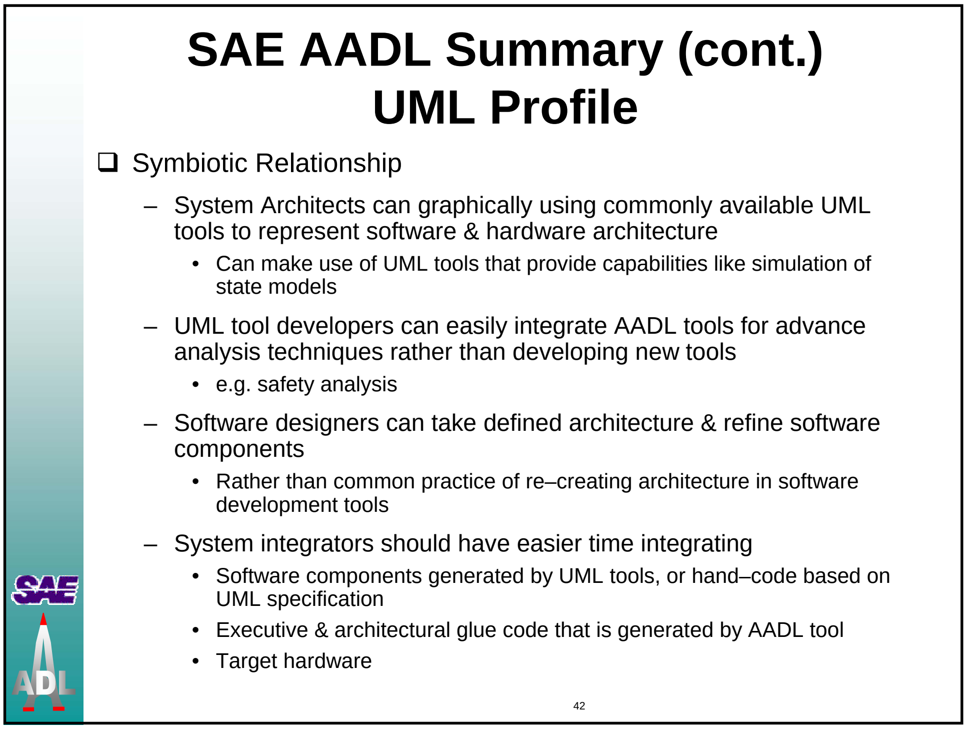
Outline

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



SAE AADL Summary

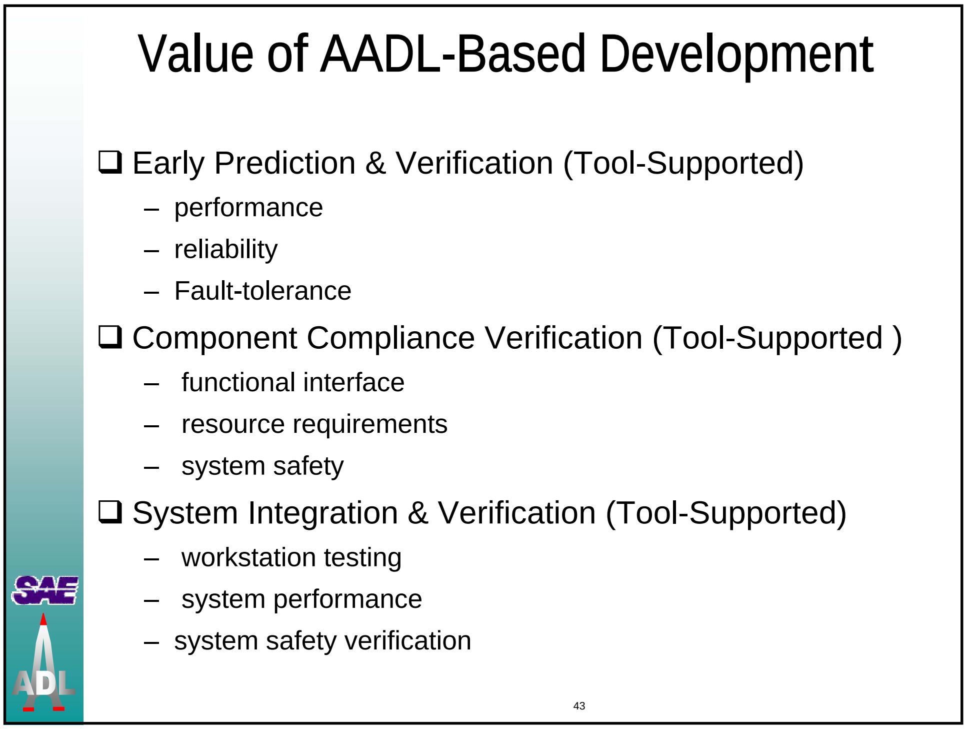
- □ AADL is Architecture Description Language & tools for predictable systems
  - – e.g. time, reliability, fault-tolerance
  - – Based on 15 years of DARPA research
- □ AADL provides means to:
  - – Specify software & hardware architecture
  - – Incrementally develop from prototype to specification
  - – 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

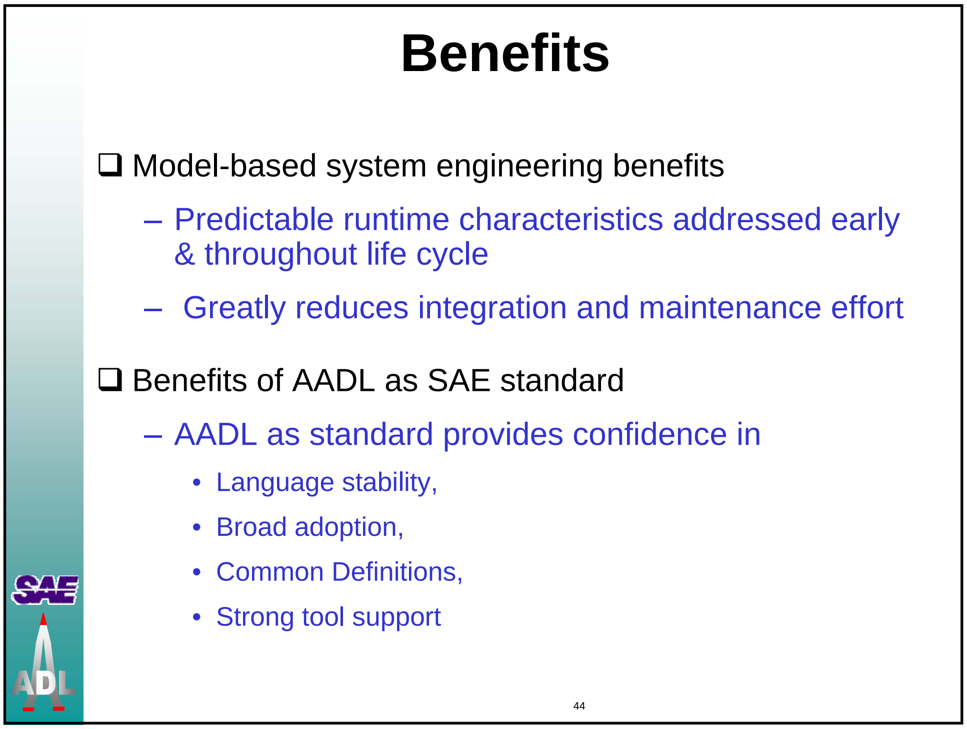
42



Value of AADL-Based Development

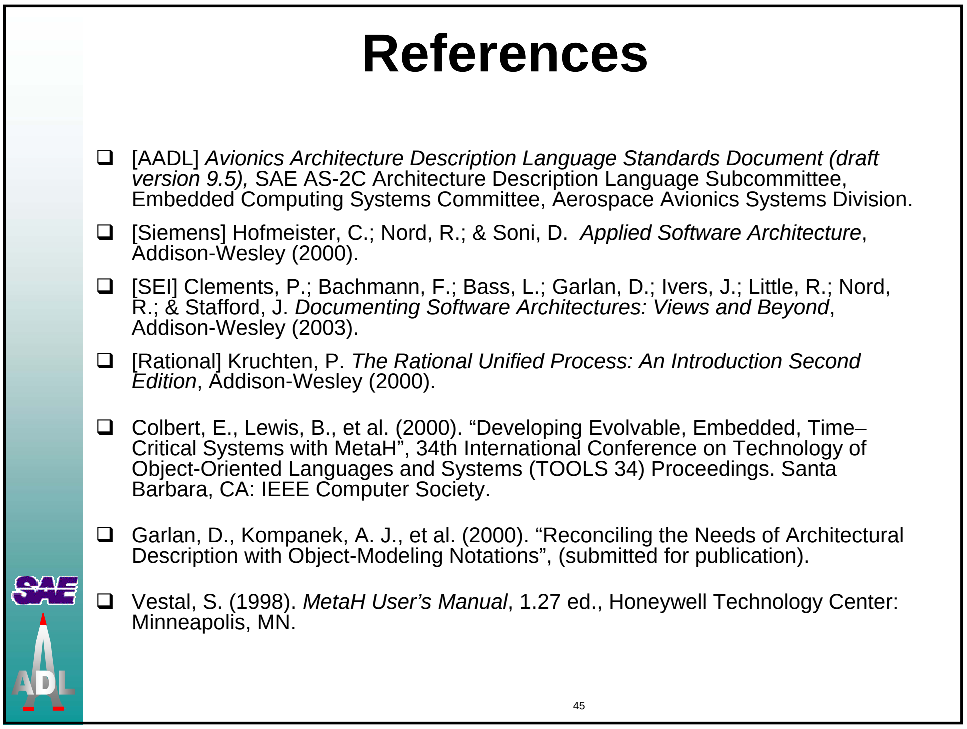
- ❑ Early Prediction & Verification (Tool-Supported)
  - – performance
  - – reliability
  - – Fault-tolerance
- ❑ Component Compliance Verification (Tool-Supported)
  - – functional interface
  - – resource requirements
  - – system safety
- ❑ System Integration & Verification (Tool-Supported)
  - – workstation testing
  - – system performance
  - – system safety verification

43



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

❑ [AADL] *Avionics Architecture Description Language Standards Document (draft version 9.5),* SAE AS-2C Architecture Description Language Subcommittee, Embedded Computing Systems Committee, Aerospace Avionics Systems Division.

❑ [Siemens] Hofmeister, C.; Nord, R.; & Soni, D. *Applied Software Architecture*, Addison-Wesley (2000).

❑ [SEI] Clements, P.; Bachmann, F.; Bass, L.; Garlan, D.; Ivers, J.; Little, R.; Nord, R.; & Stafford, J. *Documenting Software Architectures: Views and Beyond*, Addison-Wesley (2003).

❑ [Rational] Kruchten, P. *The Rational Unified Process: An Introduction Second Edition*, Addison-Wesley (2000).

❑ Colbert, E., Lewis, B., et al. (2000). "Developing Evolvable, Embedded, Time–Critical Systems with MetaH", 34th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS 34) Proceedings. Santa Barbara, CA: IEEE Computer Society.

❑ Garlan, D., Kompanek, A. J., et al. (2000). "Reconciling the Needs of Architectural Description with Object-Modeling Notations", (submitted for publication).

❑ Vestal, S. (1998). *MetaH User's Manual*, 1.27 ed., Honeywell Technology Center: Minneapolis, MN.