

# Overview of the UML Profile for the Architecture Design & Analysis Language

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



**Abs(S/W)**  
Absolute Software Co., Inc.

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



## 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



2

## Outline

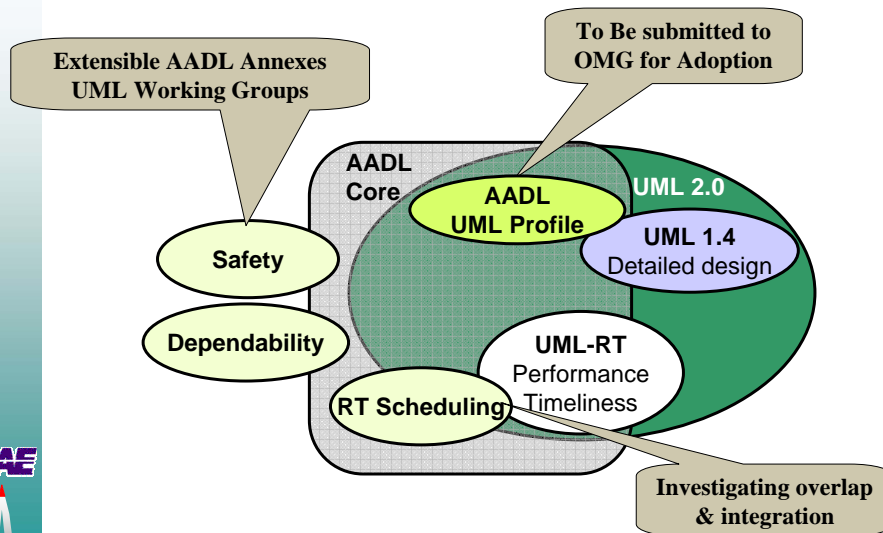
### ➡ Extending UML

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



3

## AADL/UML Relationship



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

4

## 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

## UML Profile

- ❑ Specializes UML for specific domain or process
- ❑ Predefined package of
  - Stereotypes
  - Tagged values
  - Constraints
  - Stereotype-specific icons
- ❑ Does not extend UML by adding any new basic concepts
- ❑ Provides conventions for applying & specializing standard UML to particular environment or domain
- ❑ (As defined in UML 1.5)



6

## Stereotypes

- ❑ Classify model elements at object-model level
  - Instances of stereotyped element behave as if they were instances of new metamodel classes whose form is based on existing "base" metaclasses
- ❑ Augment UML classification mechanism based on built-in UML metamodel class hierarchy
- ❑ Adds "virtual" UML metaclasses with new
  - Semantics
  - Meta-attributes
  - Property lists
  - Constraints
  - Graphical representation



7

## Stereotypes (Cont.)

- ❑ Names of new stereotypes must not clash with
  - Names of predefined metamodel elements
  - Names of other stereotypes
- ❑ A model element can be marked by 1 stereotype
  - Also called "classified by" or "stereotyped"
  - Stereotype can be constructed as specialization of other stereotypes
  - Receives features & semantics defined for stereotype
- ❑ Intent is that tools & repositories be able to manipulate stereotyped element
  - Same as ordinary element for most editing & storage purposes
  - Differentiating it for certain semantic operations, such as well-formedness checking, code generation, or report writing



8

## Property Lists & Tagged Values

- Any modeling element may have “arbitrary” information attached in form of property list
- Property List consists of tag-value pairs
  - Tag is user-definable unique name string for property
  - Value is string
    - “Arbitrary” from UML’s perspective
    - May be constrained by definer
    - May be meaningful to tools
- Stereotype may require specific
  - Set of tags
    - “pseudo-attributes”
  - Optional default values
    - “constraints”



9

## Constraints


- Semantic condition or restriction
  - Boolean expression associated with model element(s)
    - Must be true for the model to be well formed
  - Assertion not an executable statement
  - Certain constraints are predefined in UML
- 3 forms
  - Invariant
  - Precondition
  - Postcondition
- May be expressed in UML’s Object-Constraint Language (OCL)
- May be associated with specific stereotype to define semantics



10


## 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.



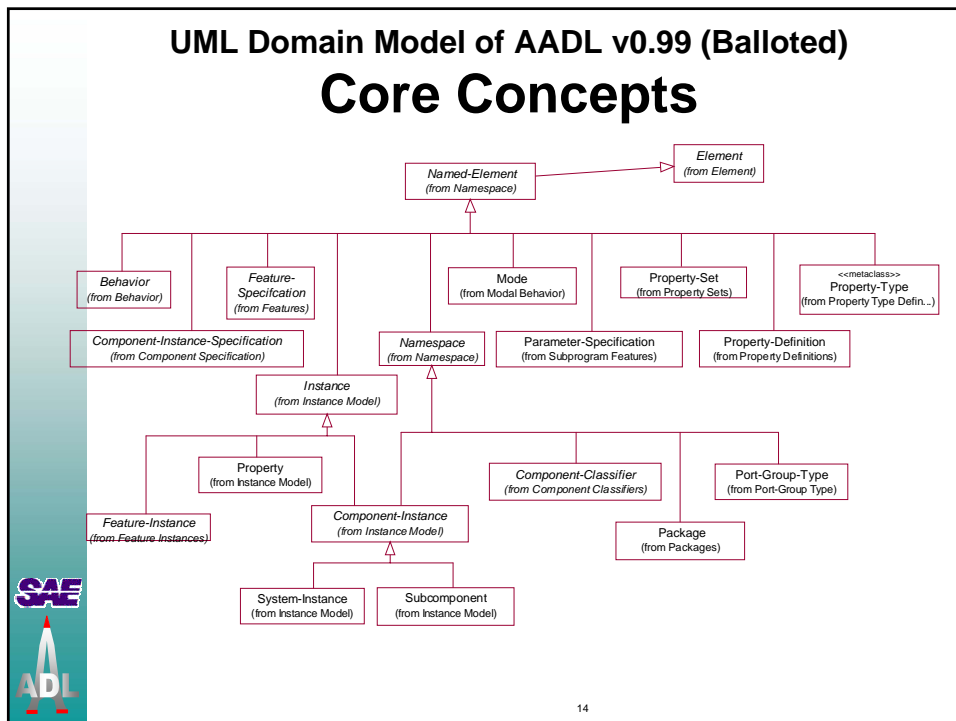
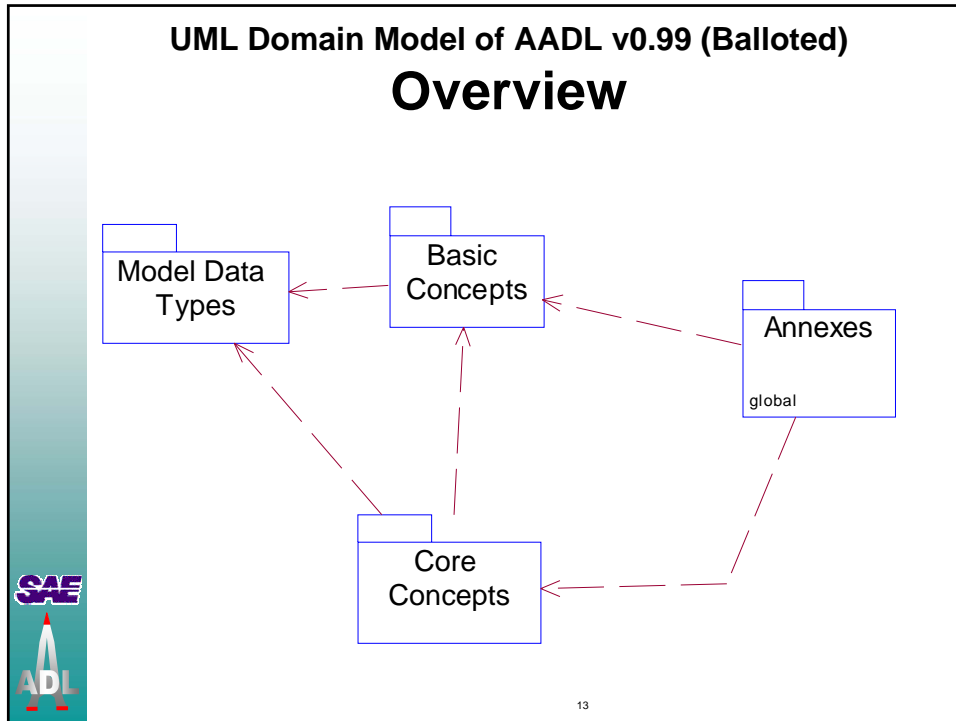
11

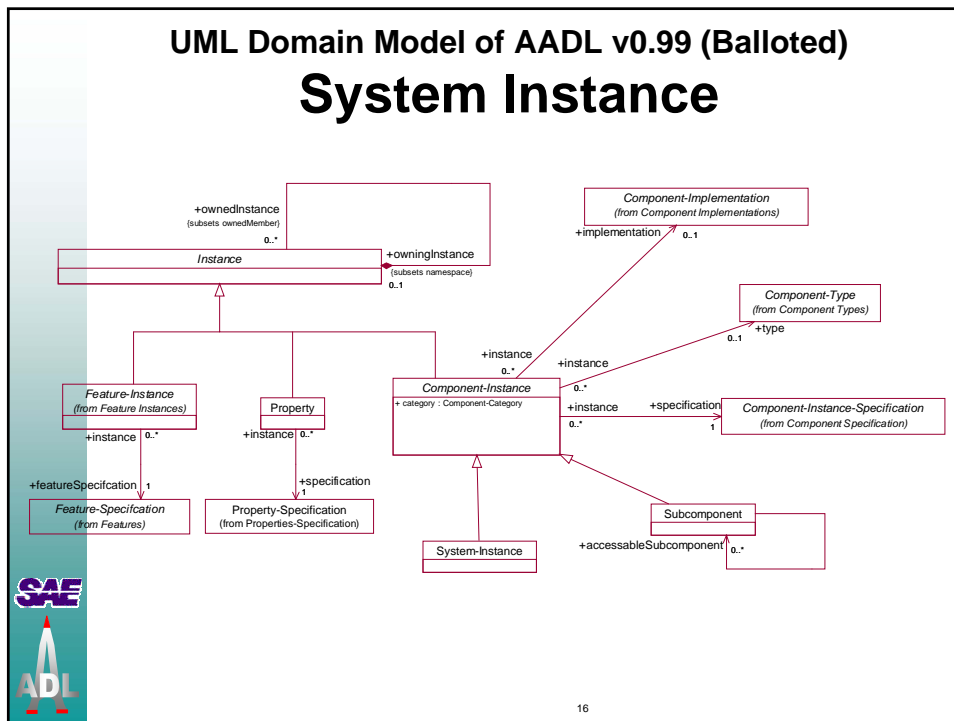
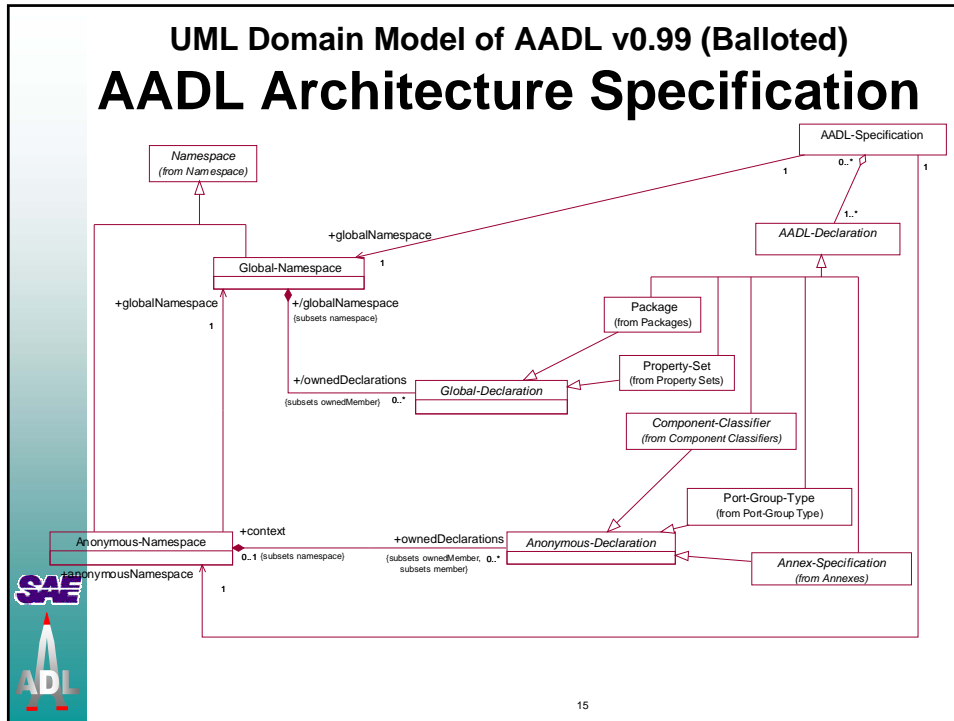
## Outline

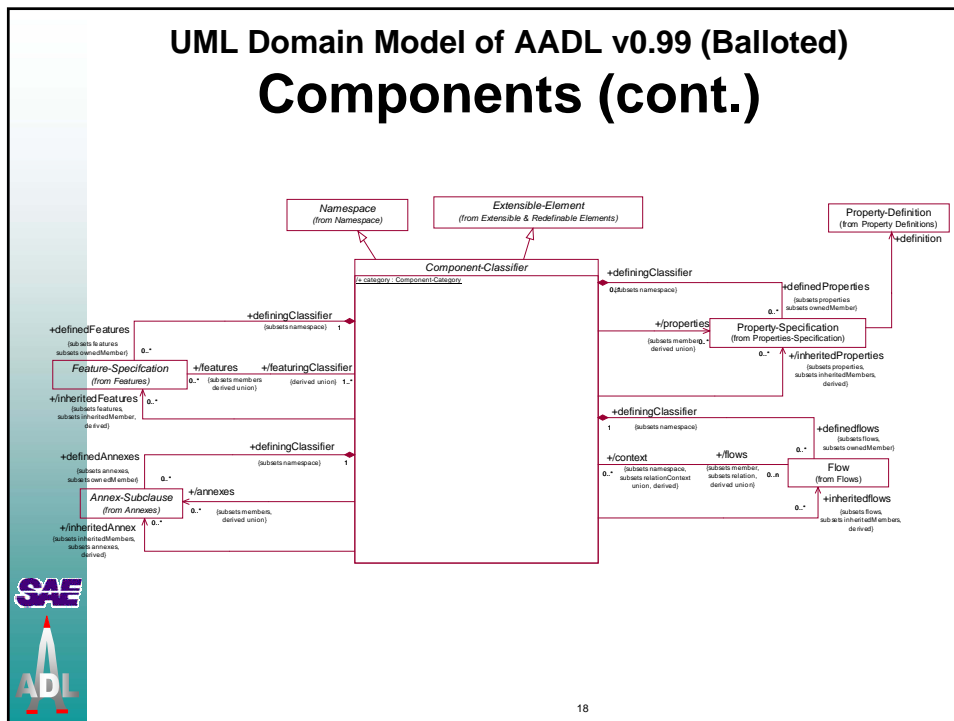
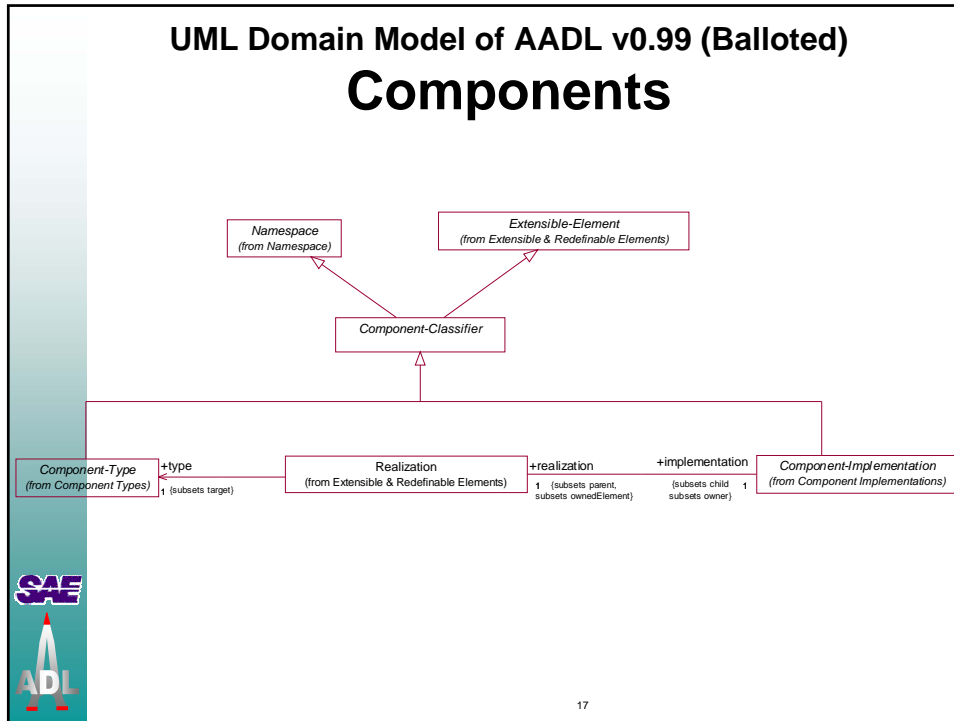
- Extending UML
-  Draft UML Domain Model for AADL
- AADL/UML Profile
- AADL/Example
- Summary of Part II

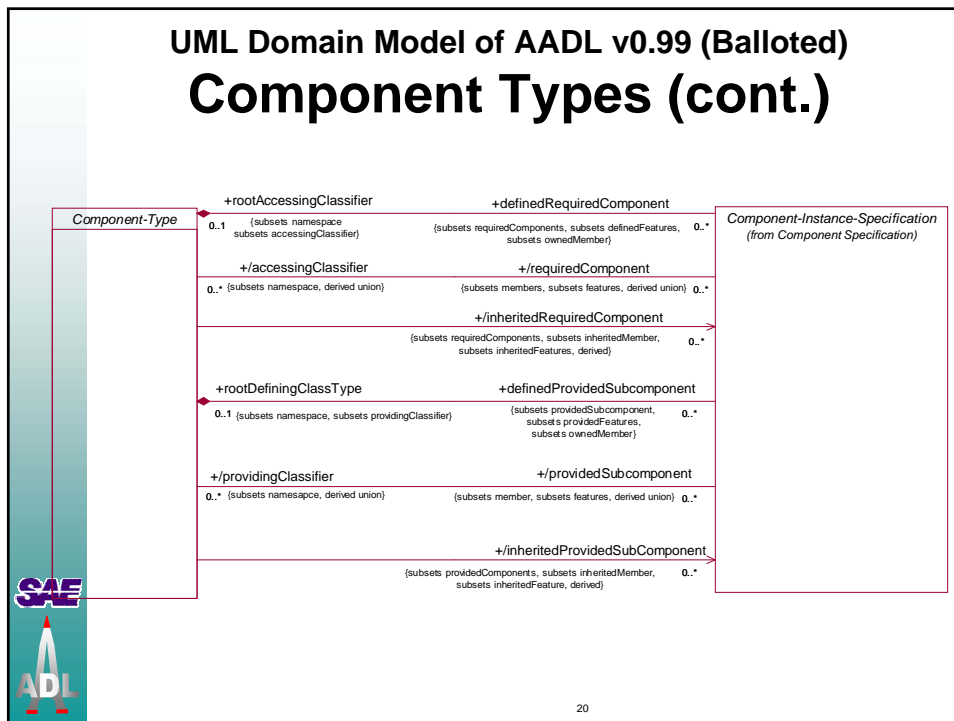
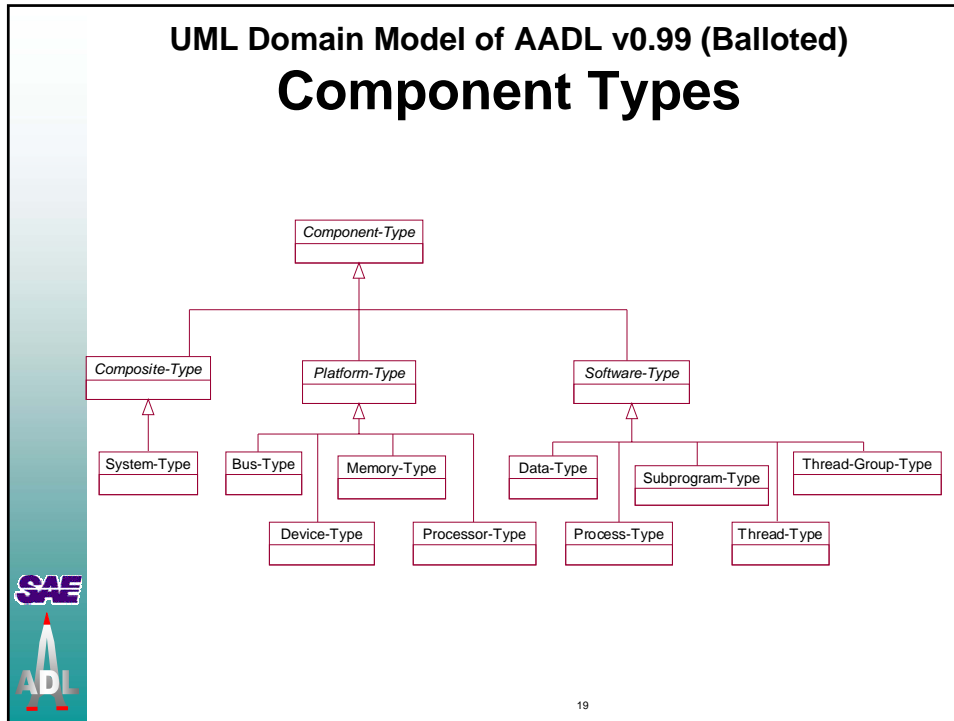


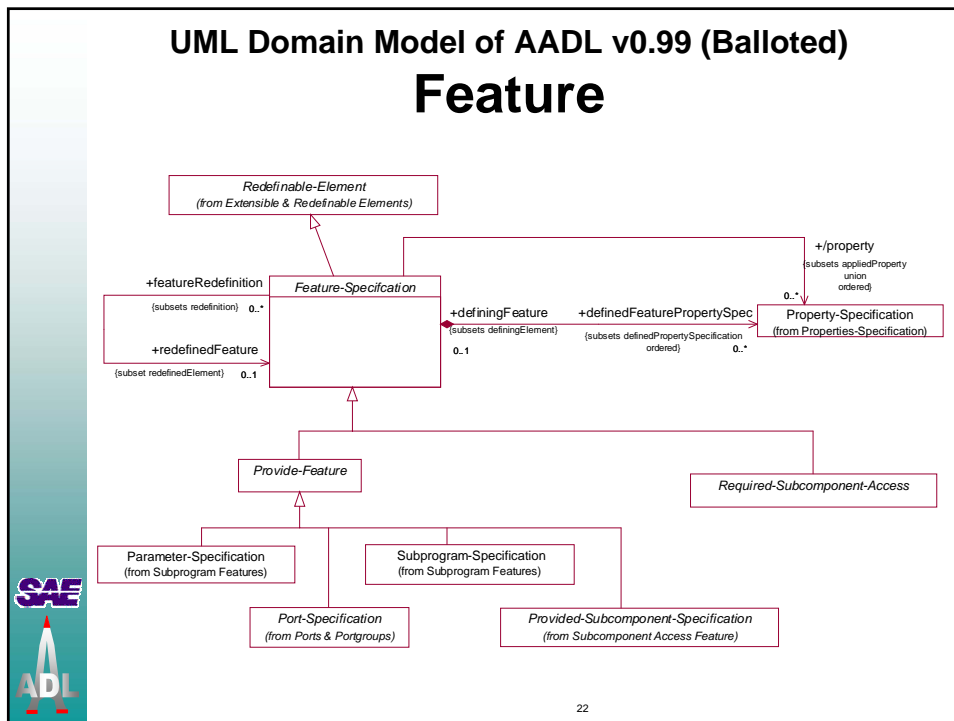
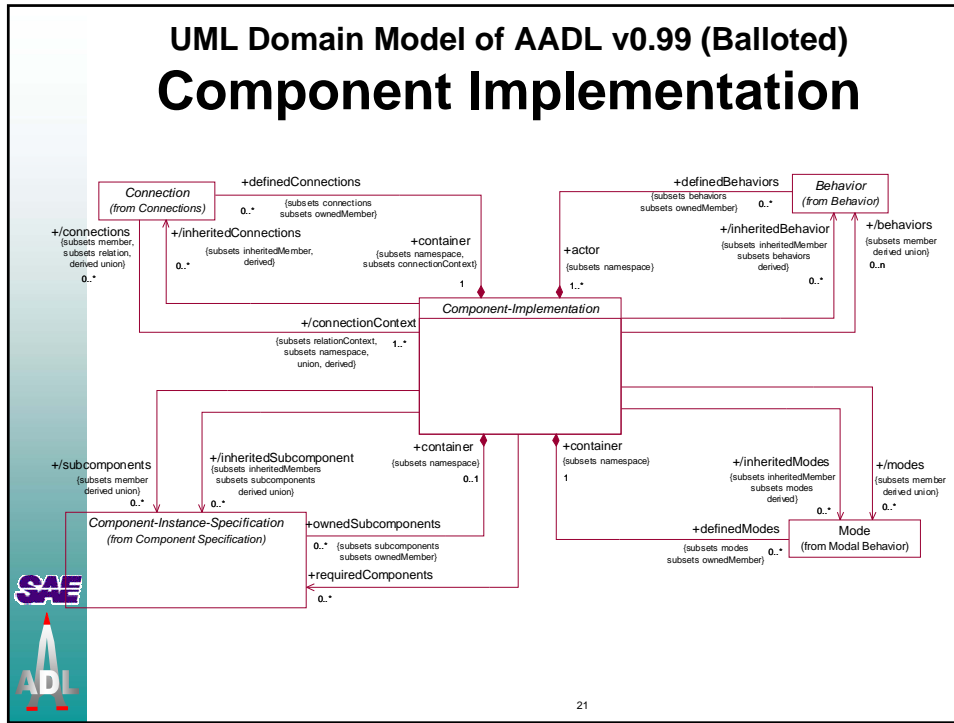
12

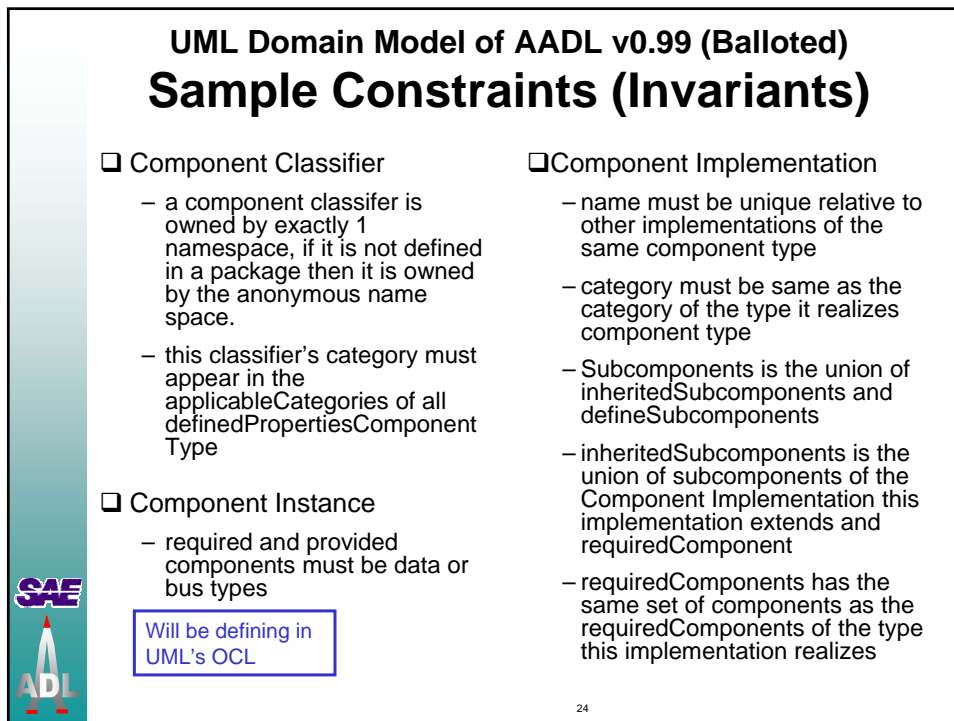
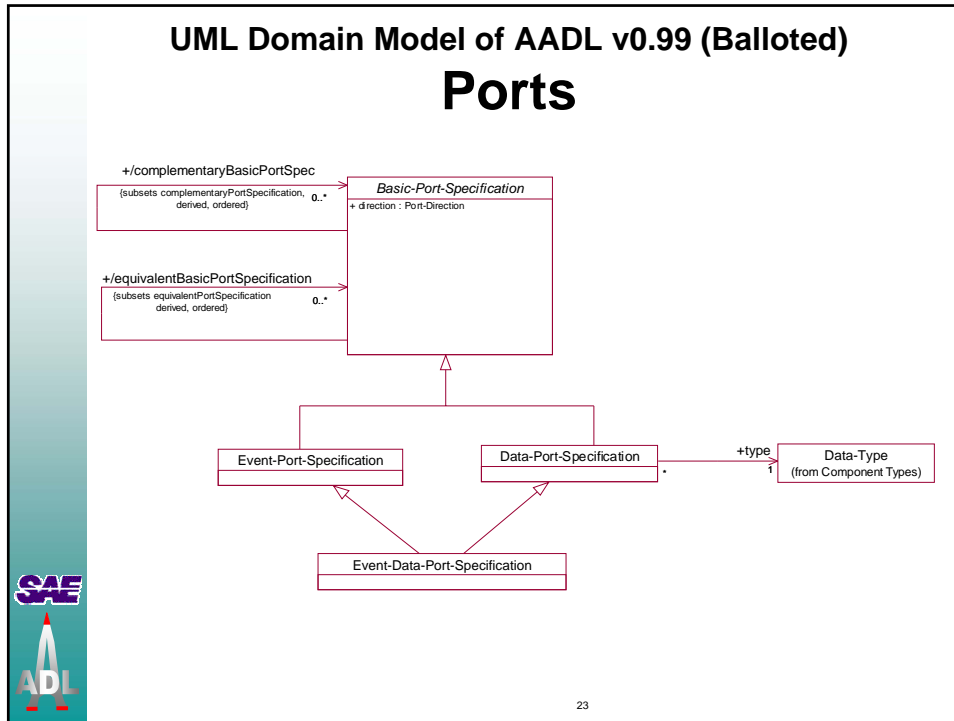












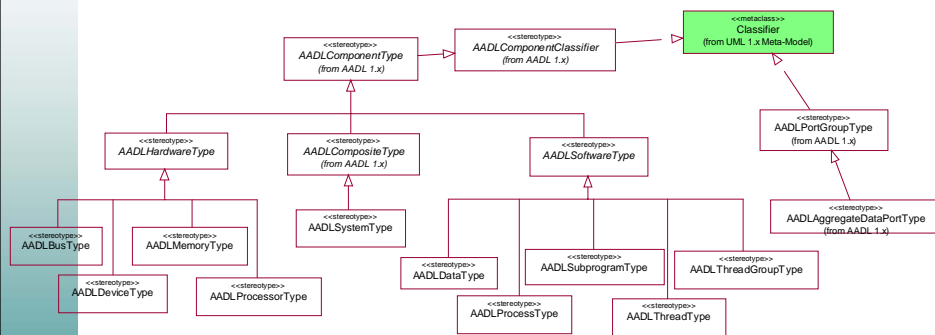
## Outline

- ❑ 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



26

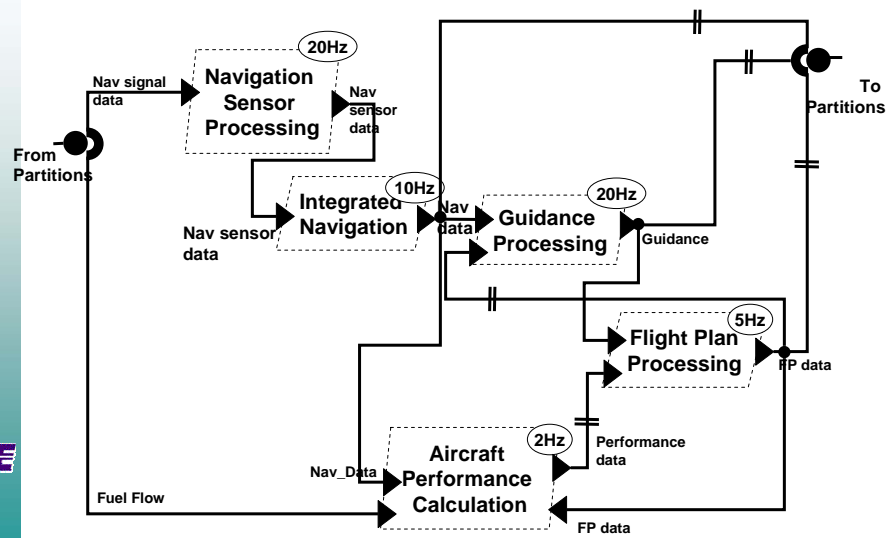
## Outline

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

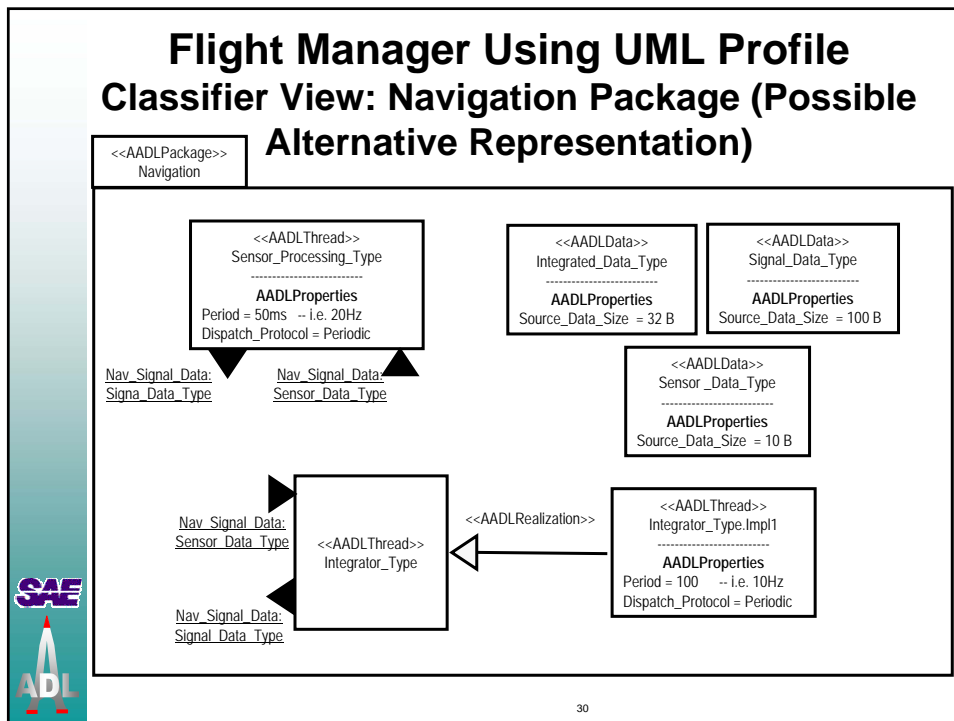
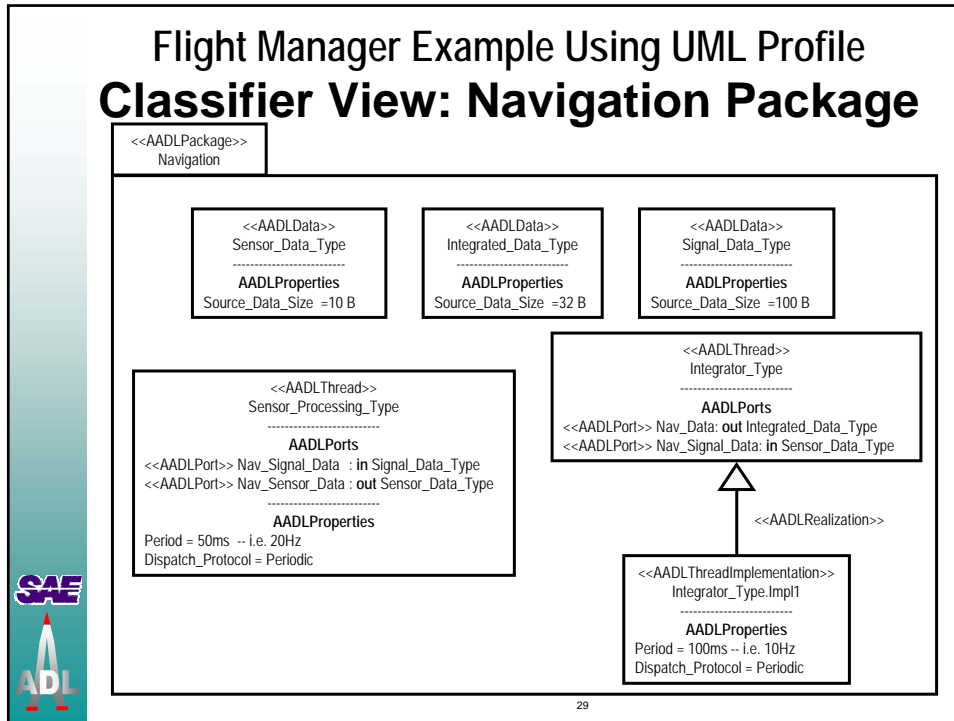


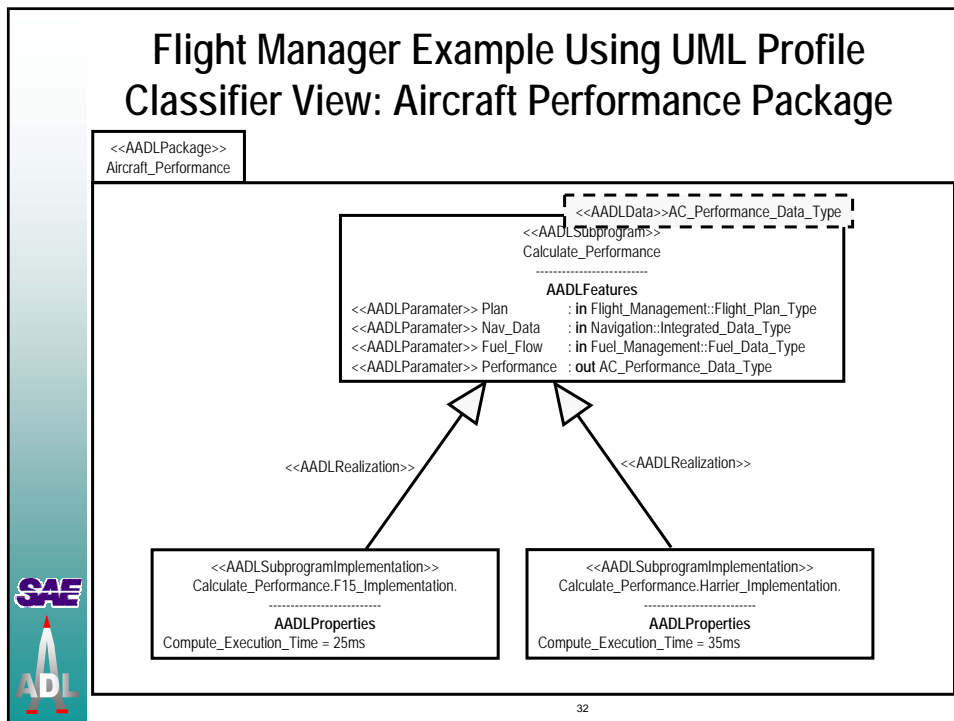
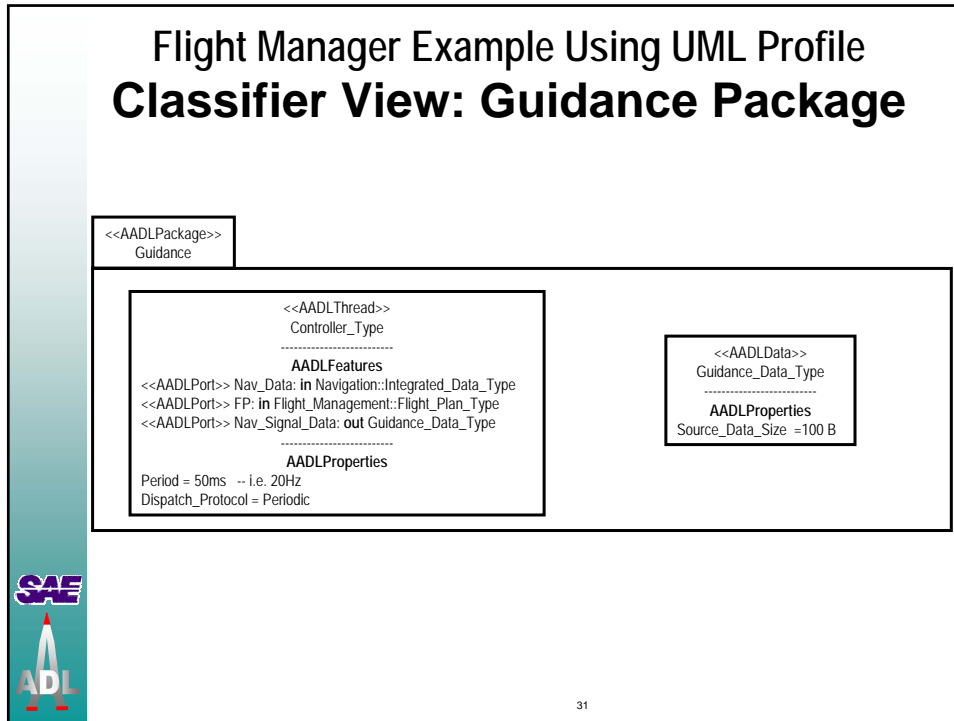
27

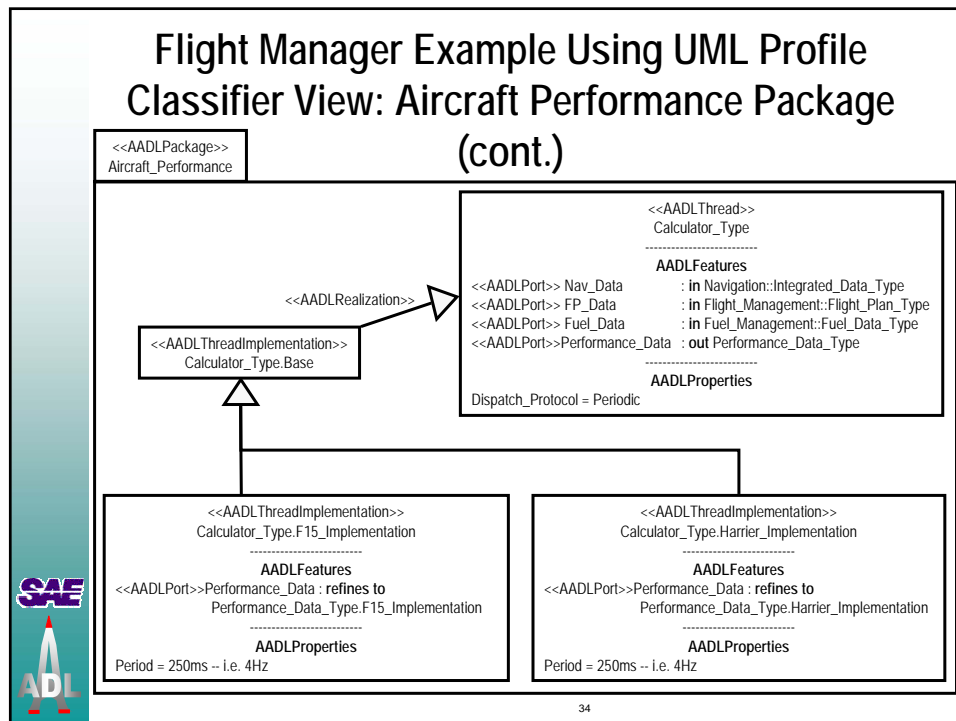
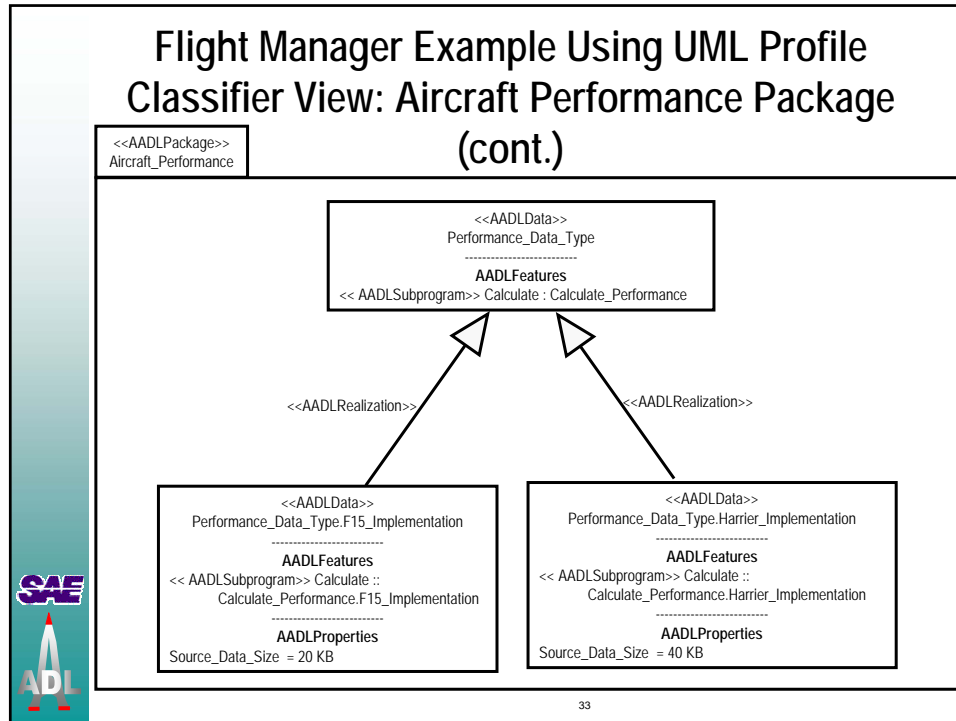
## Flight Manager in AADL Graphical Notation



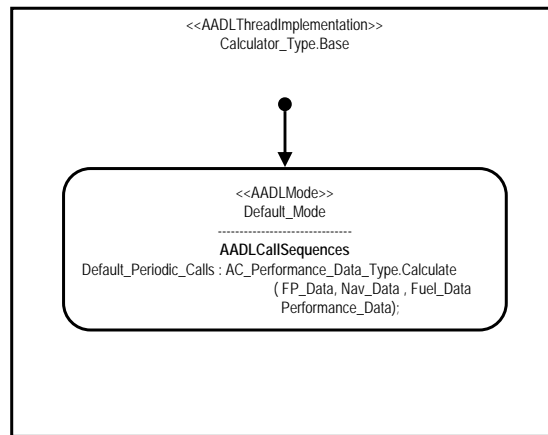
28





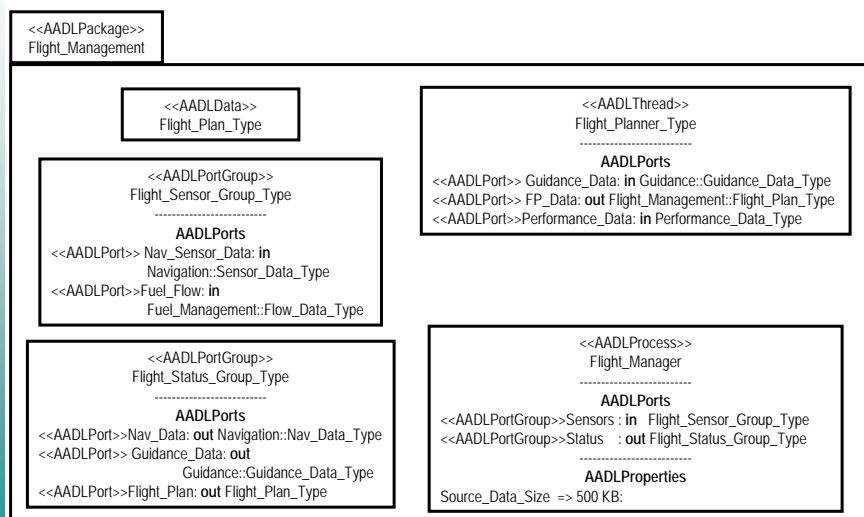


## Flight Manager Example Using UML Profile Behavior View: Aircraft Performance Package's Calculator\_Type.Base Thread Impl.

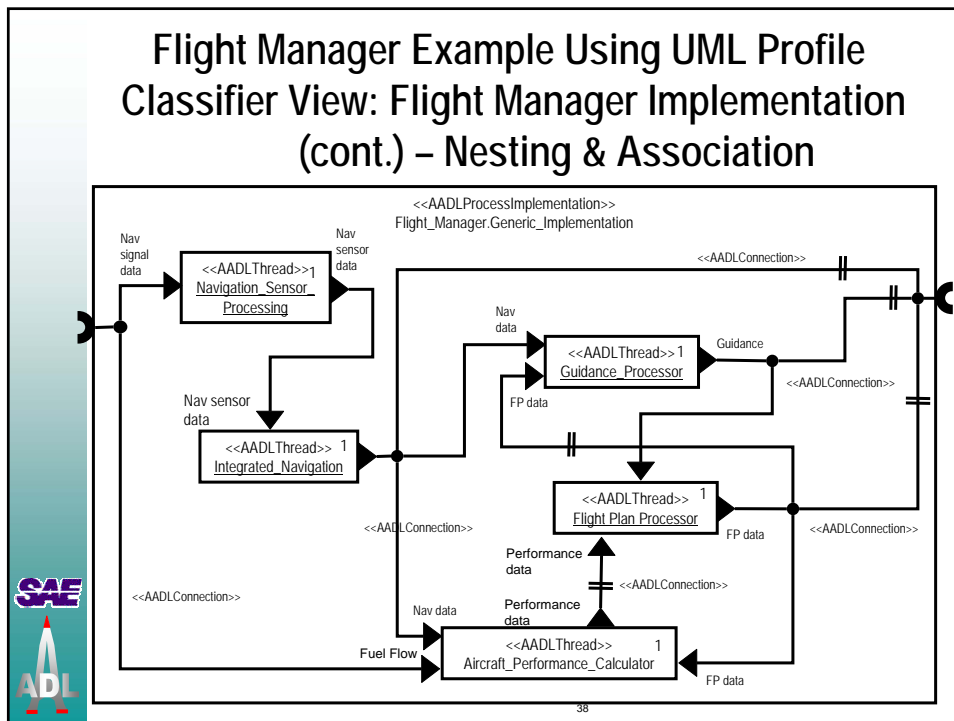
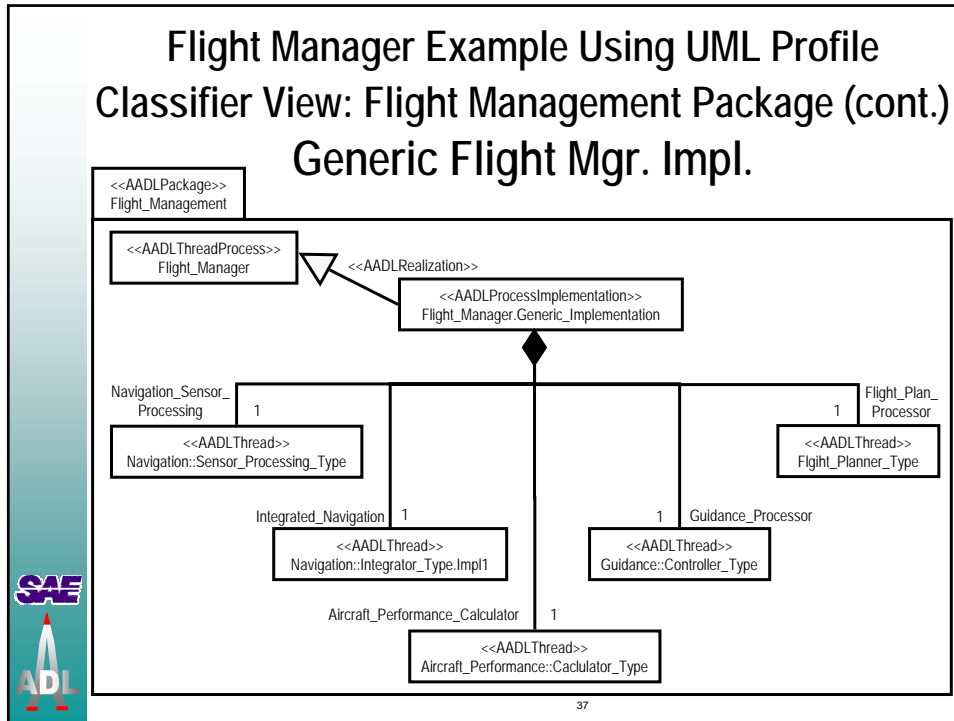


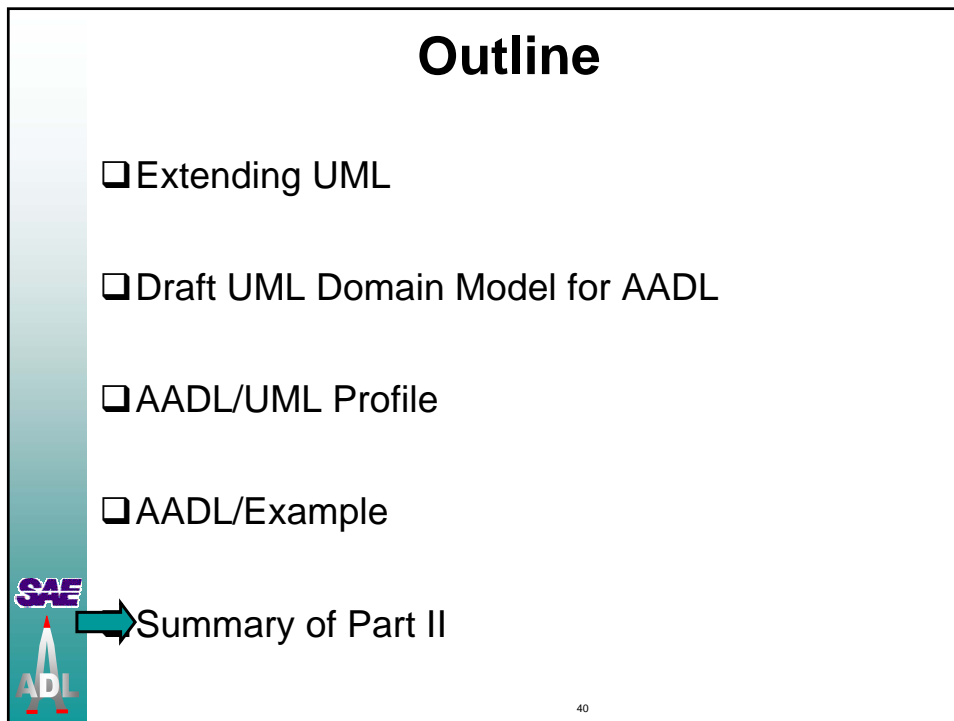
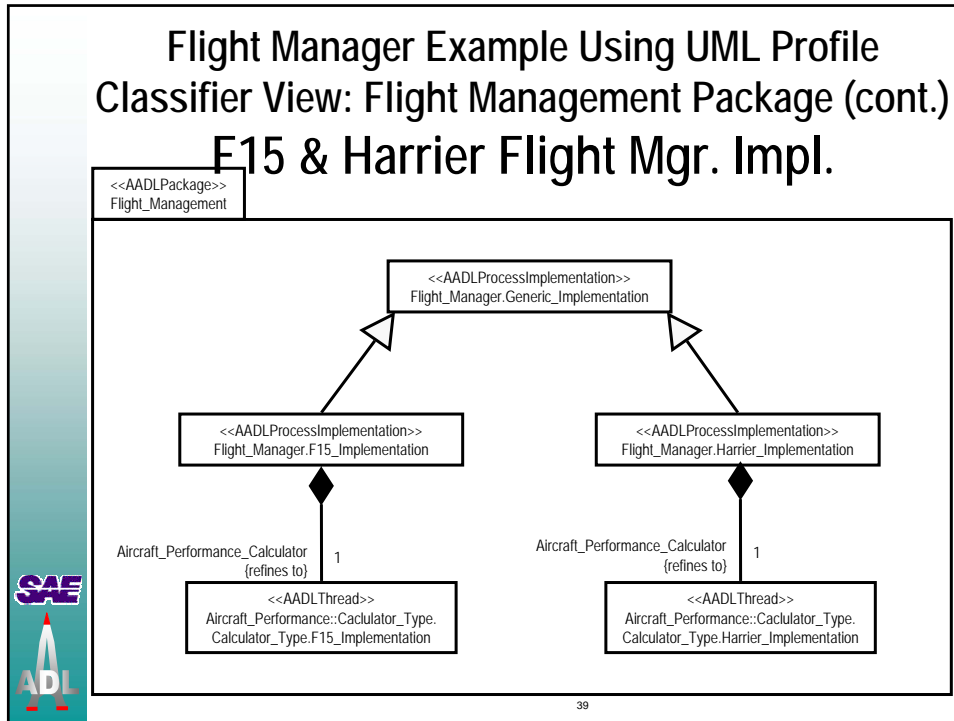
35

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



36





## 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



41

## 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.



42