UML (Unified Modeling Language)

4 Common Mechanism

Software Institute of
Nanjing University

Common Mechanisms

- The UML is made simpler by the presence of four common mechanisms that apply consistently throughout the language:
  - specifications,
  - adornments,
  - common divisions,
  - extensibility mechanisms.

1. Introduction

- This lesson explains the use of two of these common mechanisms : adornments and extensibility mechanisms.
  - Notes
  - Stereotypes
  - Tagged values
  - Constraints

1. Introduction (cont')

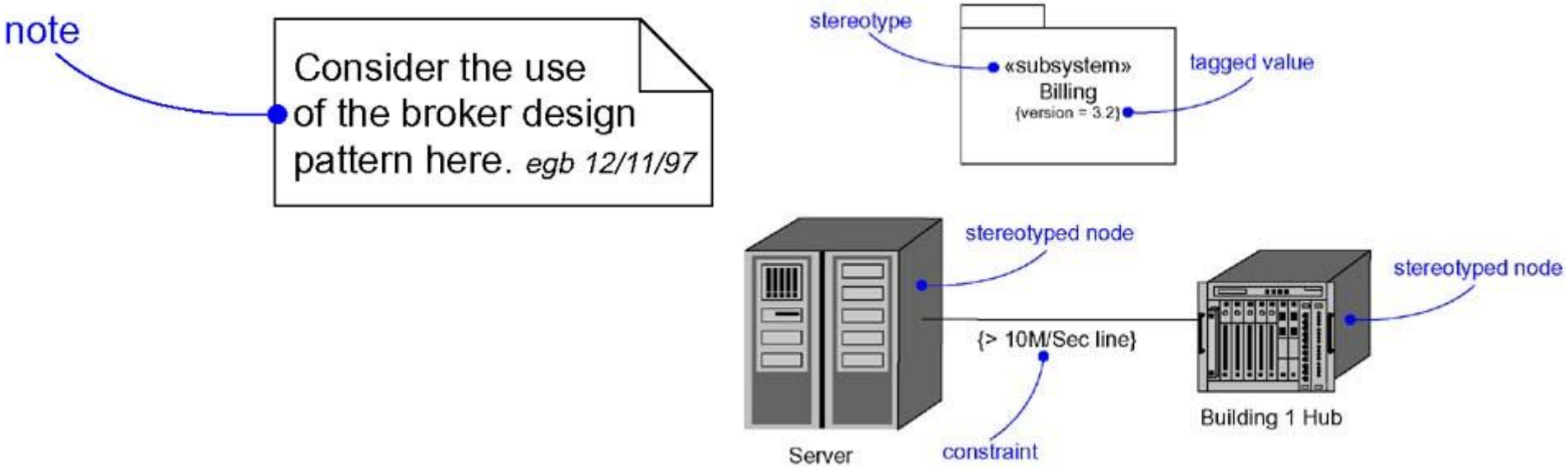
- Notes are the most important kind of adornment that stands alone.
  - A note is a graphical symbol for rendering constraints or comments attached to an element or a collection of elements.
  - Use notes to attach information to a model, such as requirements, observations, reviews, and explanations.

1. Introduction (cont')

- The UML's extensibility mechanisms permit you to extend the language in *controlled* ways. These mechanisms include:
  - stereotypes,
  - tagged values,
  - and constraints.

1. Introductions (cont')

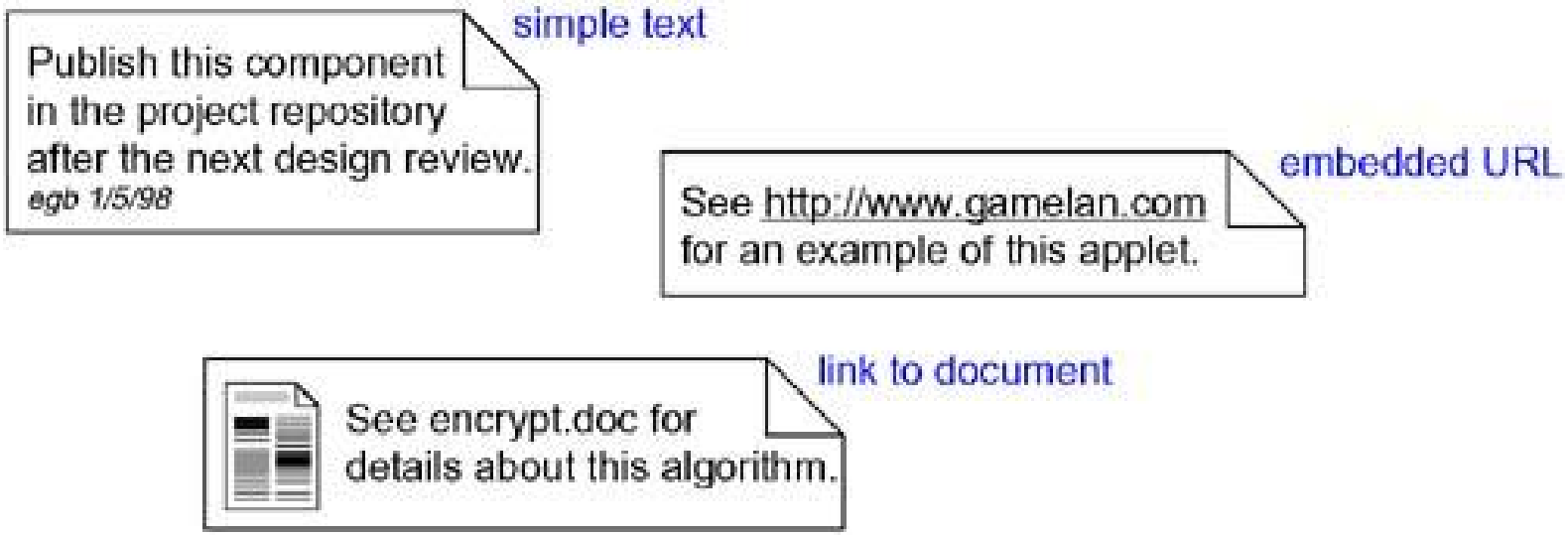
- The UML provides a textual representation for notes, stereotypes, tagged values, and constraints, as ...



2. Notes

- A *note* is a graphical symbol for rendering constraints or comments attached to an element or a collection of elements.
- Graphically, a note is rendered as a rectangle with a dog-eared corner, together with a textual or graphical comment.

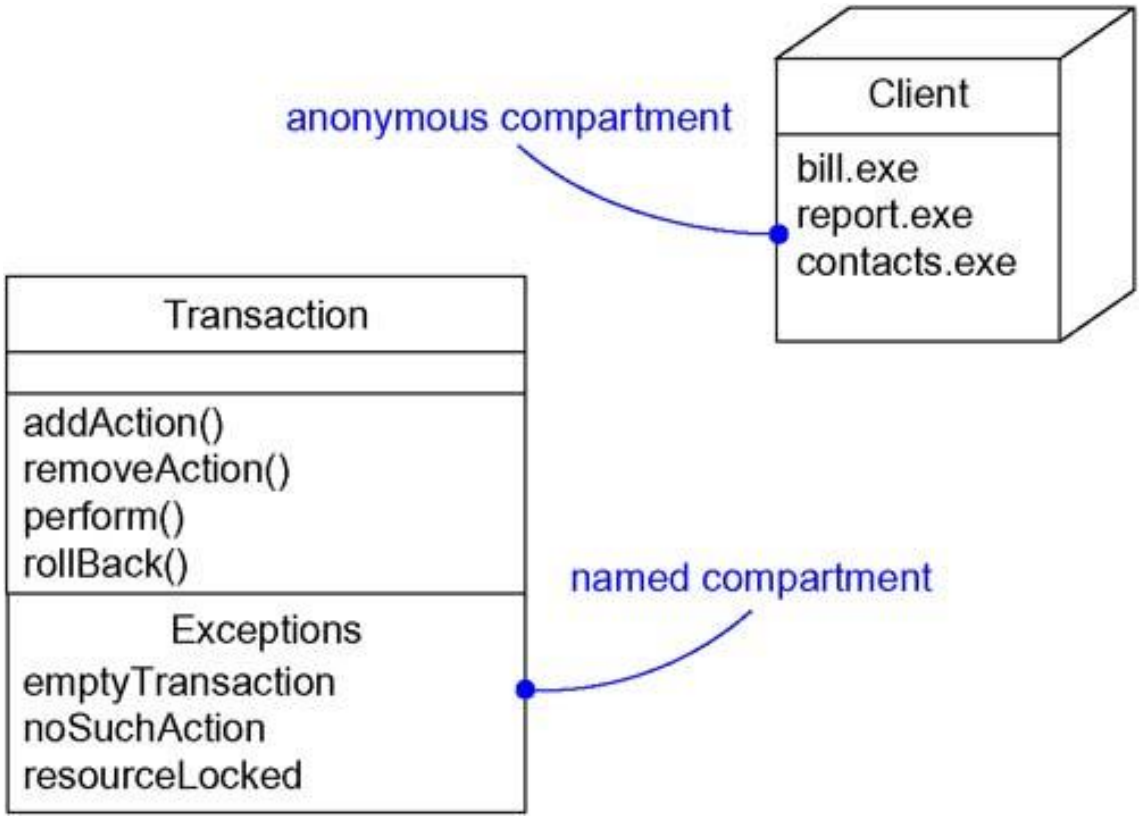
2. Notes (cont')


simple text
Publish this component in the project repository after the next design review.
egb 1/5/98
embedded URL
See http://www.gamelan.com for an example of this applet.
link to document
See encrypt.doc for details about this algorithm.

3. Other Adornments

- *Adornments* are textual or graphical items that are added to an element's basic notation and are used to visualize details from the element's specification.
- The general rule to follow is this:
  - Start with the basic notation for each element
  - and then add other adornments only as they are necessary to convey specific information that is important to your model.

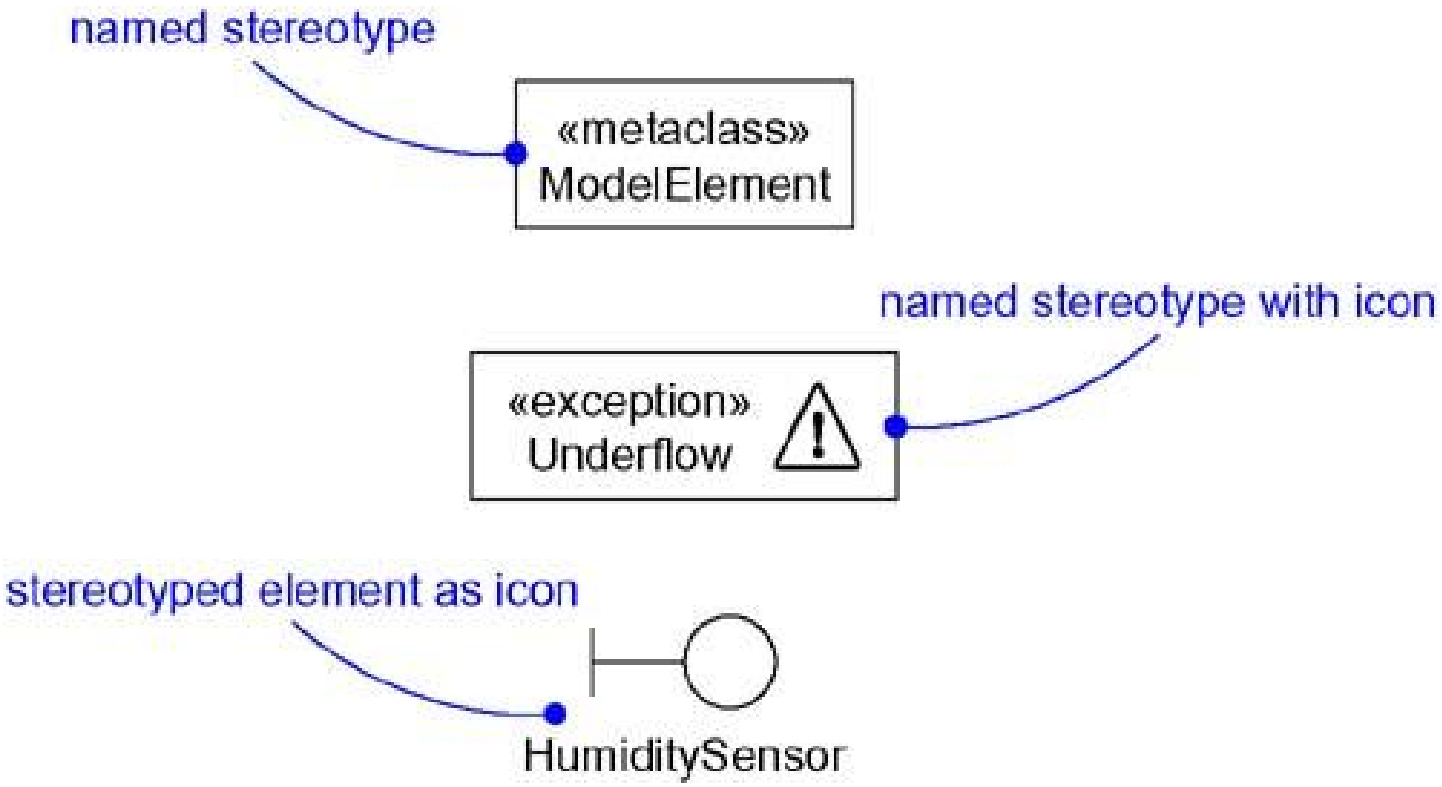
3. Other Adornment (cont')



3. Stereotypes

- A *stereotype* is an extension of the vocabulary of the UML, allowing you to create new kinds of building blocks similar to existing ones but specific to your problem.
- Graphically, a stereotype is rendered as a name enclosed by guillemets and placed above the name of another element.
  - As an option, the stereotyped element may be rendered by using a new icon associated with that stereotype.

3. Stereotypes



3.1 Standard Stereotype (Classes)

- *<<actor>>*
- *<<interface>>*
- *<<signal>>*
- *<<type>>*
- <<enumeration>>
- <<exception>>
- <<implementationClass>>
- <<process>>
- <<thread>>
- <<utility>>
- ...

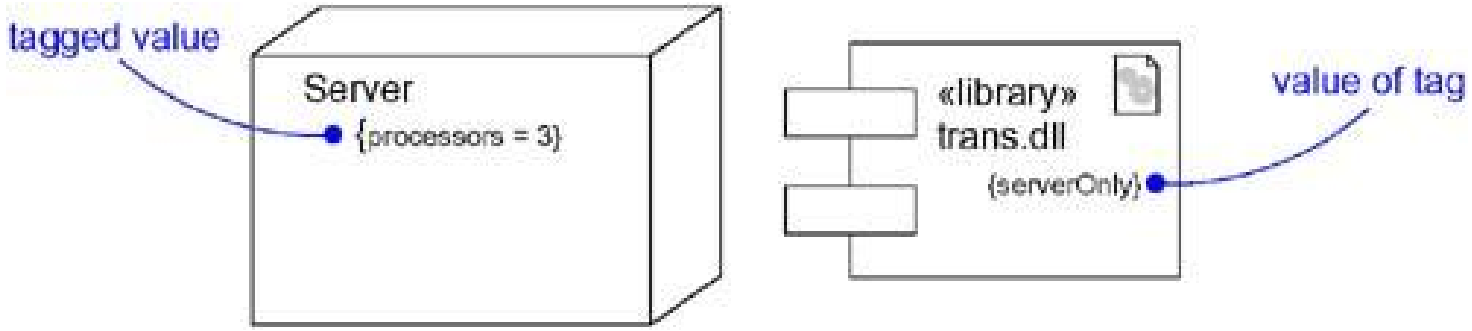
3.2 Standard Stereotype (Notes)

- <<requirement>>
- <<responsibility>>

4. Tagged Values

- A *tagged value* is an extension of the properties of a UML element, allowing you to create new information in that element's specification.
- Graphically, a tagged value is rendered as a string enclosed by brackets and placed below the name of another element.

4. Tagged Values (cont')



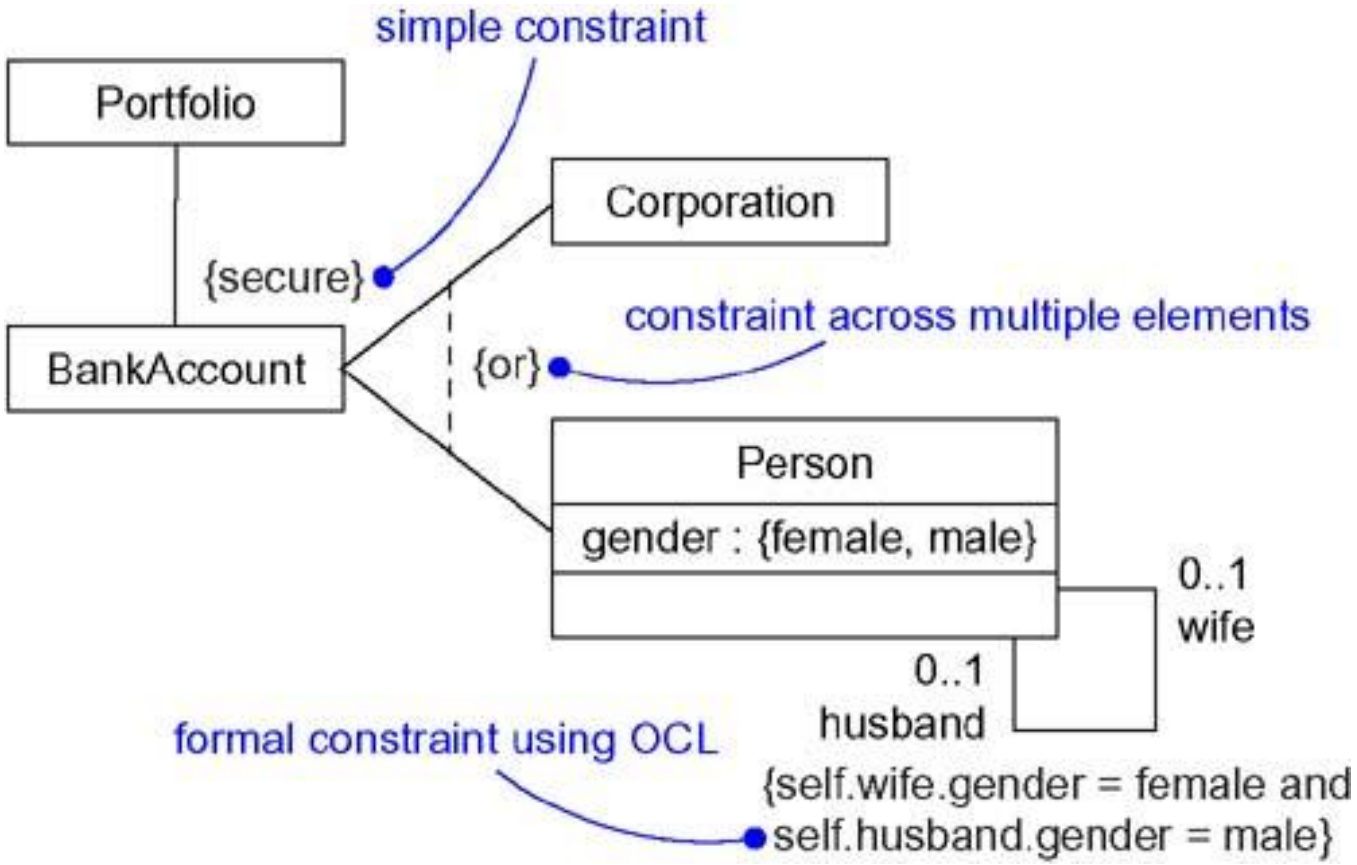
6.4 Standard Tagged Values

- {documentation=...}
- {location=...}
- {persistence=persistent/transient }
- {semantics=...}

5. Constraints

- A *constraint* is an extension of the semantics of a UML element, allowing you to add new rules or to modify existing ones.
- Graphically, a constraint is rendered as a string enclosed by brackets and placed near the associated element or connected to that element by dependency relationships.

5. Constraints (cont')



5.1 Standard Stereotype (constraint)

- Design by Contract
  - <<invariant>>
  - <<precondition>>
  - <<postcondition>>

5.2 Standard Constraints

- generalization
  - {complete}
  - {incomplete}
  - {disjoint}
  - {overlapping}
- association
  - {implicit}
  - {or}

6. Standard Elements

- The UML defines a number of standard stereotypes for classifiers, components, relationships, and other modeling elements.

7. Common Modeling Techniques

- Modeling Comments
- Modeling New Building Blocks
- Modeling New Properties
- Modeling New Semantics

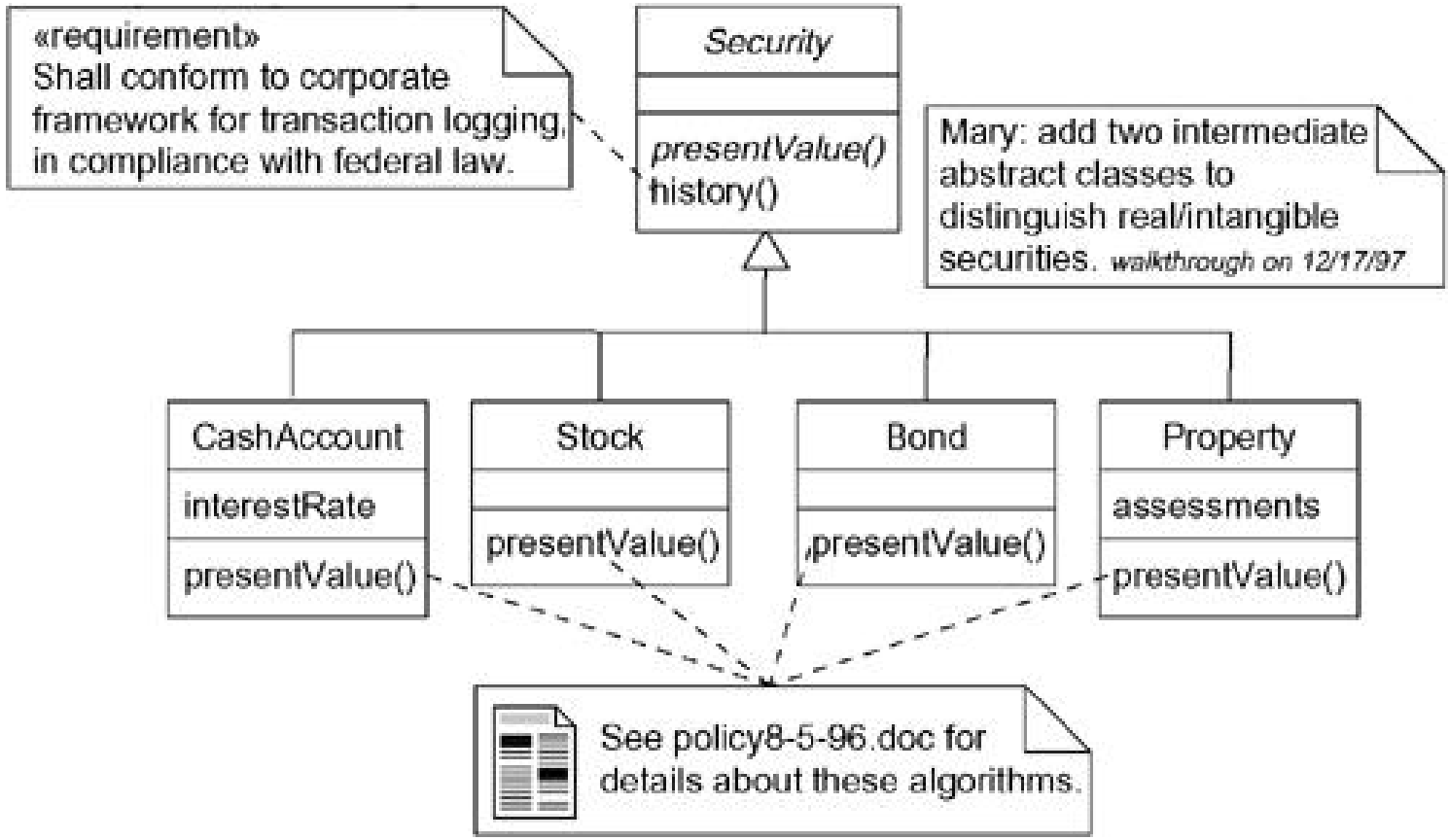
7.1 Modeling Comments

- Put your comment as text in a note and place it adjacent to the element to which it refers.
  - You can show a more explicit relationship by connecting a note to its elements using a dependency relationship.
- Remember that you can hide or make visible the elements of your model as you see fit.

7.1 Modeling Comments (cont')

- If your comment is lengthy or involves something richer than plain text, consider putting your comment in an external document and linking or embedding that document in a note attached to your model.
- As your model evolves, keep those comments that record significant decisions that cannot be inferred from the model itself, and—unless they are of historic interest—discard the others.

7.1 Modeling Comments (cont')



7.2 Modeling New Building Blocks

- Make sure there's not already a way to express what you want by using basic UML.
- If you're convinced there's no other way to express these semantics, identify the primitive thing in the UML that's most like what you want to model (for example, class, interface, component, node, association, and so on) and define a new stereotype for that thing.

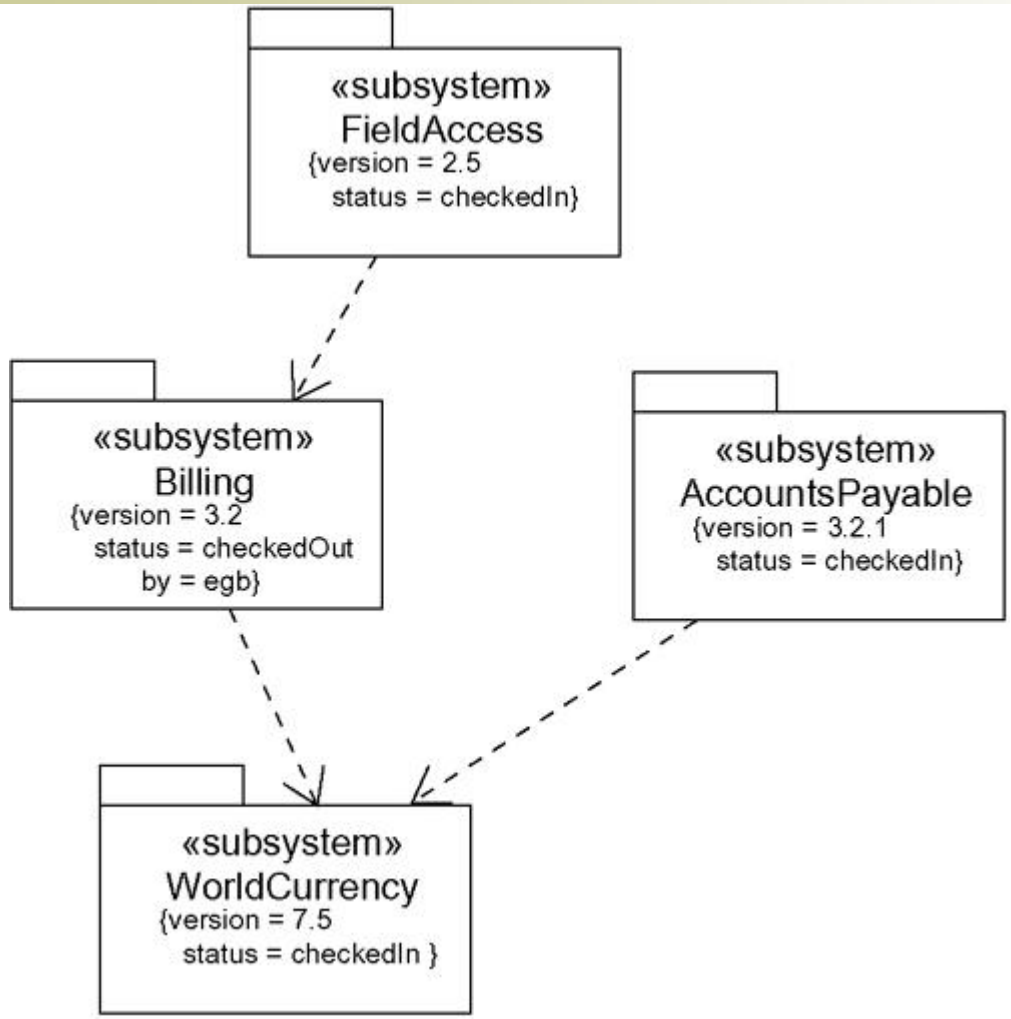
7.2 Modeling New Building Blocks (cont')

- Specify the common properties and semantics that go beyond the basic element being stereotyped by defining a set of tagged values and constraints for the stereotype.
- If you want these stereotype elements to have a distinctive visual cue, define a new icon for the stereotype.

7.3 Modeling New Properties

- First, make sure there's not already a way to express what you want by using basic UML. If you have a common modeling problem, chances are that there's already some standard tagged value that will do what you want.
- If you're convinced there's no other way to express these semantics, add this new property to an individual element or a stereotype. The rules of generalization apply—tagged values defined for one kind of element apply to its children.

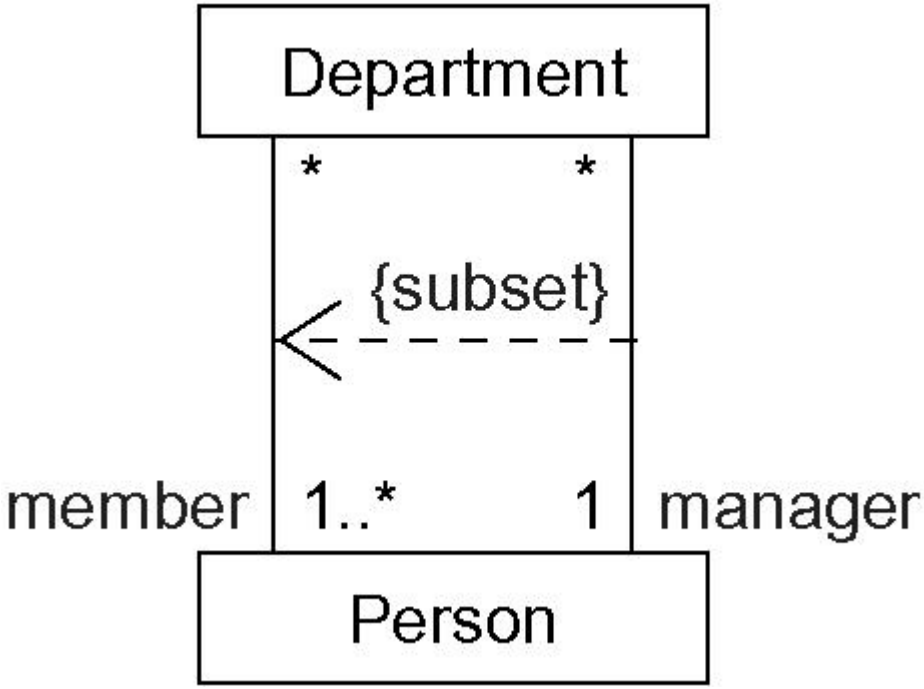
7.3 Modeling New Properties (cont')



7.4 Modeling New Semantics

- First, make sure there's not already a way to express what you want by using basic UML.
- If you're convinced there's no other way to express these semantics, write your new semantics as text in a constraint and place it adjacent to the element to which it refers. You can show a more explicit relationship by connecting a constraint to its elements using a dependency relationship.
- If you need to specify your semantics more precisely and formally, write your new semantics using OCL.

7.4 Modeling New Semantics (cont')



8. Hints and Tips

- When you adorn a model with notes,
  - Use notes only for those requirements, observations, reviews, and explanations that you can't express simply or meaningfully using existing features of the UML.
  - Use notes as a kind of electronic sticky note, to keep track of your work in progress.
- When you draw notes,
  - Don't clutter your models with large blocks of comments. Rather, if you really need a long comment, use notes as a placeholder to link to or embed a document that contains the full comment.

8. Hints and Tips (cont')

- When you extend a model with stereotypes, tagged values, or constraints,
  - Standardize on a small set of stereotypes, tagged values, and constraints to use on your project, and avoid letting individual developers create lots of new extensions.
  - Chose short, meaningful names for your stereotypes and tagged values.
  - Where precision can be relaxed, use free-form text for specifying constraints. If you need more rigor, use the OCL to write constraint expressions.

8. Hints and Tips (cont’)

- When you draw a stereotype, tagged value, or constraint,
  - Use graphical stereotypes sparingly. You can totally change the basic notation of the UML with stereotypes, but in so doing, you'll make it impossible for anyone else to understand your models.
  - Consider using simple color or shading for graphical stereotypes, as well as more complicated icons. Simple notations are generally the best, and even the most subtle visual cues can go a long way in communicating meaning.