


Supplementary:

Using and Extending UML

CSE870: Advanced Software Engineering: Extending and Using UML (Cheng)




Extensibility Mechanisms

- Stereotypes
- Tagged Values
- Constraints

{add runs in O(1) time}

<code><<container>></code>
ActionQueue <code>{version = 3.2}</code>
add(a: Action) remove(n: Integer)

CSE870: Advanced Software Engineering: Extending and Using UML (Cheng)




Stereotypes

- Extends the UML vocabulary
- Create new building blocks derived from existing ones,
 - specific to problem domain
 - Example: Sensors for embedded systems,

<code><<sensor>></code> thermometer	<code><<sensor>></code> Voltage meter
temp	volt
read	read

CSE870: Advanced Software Engineering: Extending and Using UML (Cheng)




Tagged Values

- Extends properties of UML building block
- Create new information in that element's spec
- Example: {version, author values for Software Class}

<pre> <<sensor>> Voltage meter {reader=J. Smith} volt read </pre>	<pre> EventQueue {version =3.2 Author = bhc} add() remove() flush() </pre>
---	--

CSE870: Advanced Software Engineering: Extending and Using UML (Cheng)




Constraint

- Extends the semantics of a UML building block
- Add new rules or modify existing ones
 - Example: {ordered constraint on an add operation for a Queue}

<pre> <<sensor>> Voltage meter {reader=J. Smith} volt read </pre>	{read before thermostat check}
---	--------------------------------

CSE870: Advanced Software Engineering: Extending and Using UML (Cheng)



Interface ○

- Interface: collection of operations that specify a service of a class or component
 - Externally visible behavior of that element
 - Set of operation specs (signatures)
 - Not a set of operation implementations

CSE870: Advanced Software Engineering: Extending and Using UML (Cheng)

Interface Example

- Interface specifies desired behavior of an abstraction independent of implementation
- Specifies contract
- May also create stereotype to further define behavior
 - No attributes
 - Only operations
 - (visibility, concurrency, stereotypes, etc.)
- Naming convention:
 - I-Simple name OR
 - Prefixed by source package

CSE870: Advanced Software Engineering: Extending and Using UML (Cheng)


Dependency Relationship

- Definition:**
 - some set of model elements requires presence of another set of model elements
 - for semantic completeness or correctness
- Examples:**
 - `<<realize>>`: between type and class
 - `<<trace>>`: analysis class and design class
 - `<<create>>`: source class creates a target class instance
 - `<<access>>`: permission of one pkg to access public elements of another
 - `<<bind>>`: reln between parameterized model elt with formal parameter list and that model elt bound to actual parameters. (e.g., templates in C++)

CSE870: Advanced Software Engineering: Extending and Using UML (Cheng)

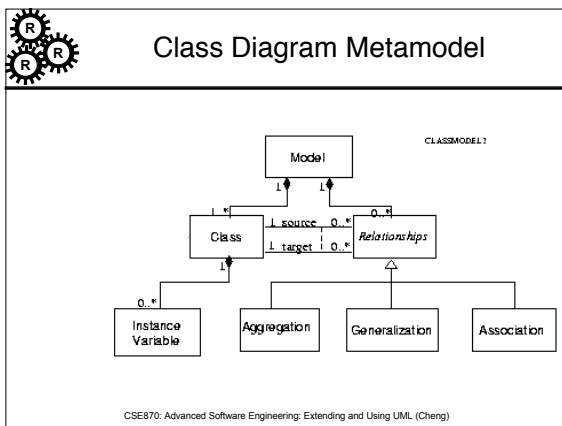
Dependency Example


CSE870: Advanced Software Engineering: Extending and Using UML (Cheng)



Metamodels

CSE870: Advanced Software Engineering: Extending and Using UML (Cheng)





UML Profile

- Official way to define customized UML for specific purpose/domain.
- Use UML extension elements (stereotypes and tags) to define language/modeling elements.
 - Elements, Attributes, Methods, Links, Link Ends, etc.

CSE870: Advanced Software Engineering: Extending and Using UML (Cheng)
