CH3. Class Modeling

Michael Blaha, James Rumbaugh
Object-Oriented Modeling and Design with UML™

2006-04-12

Tae-Hoon Song
Contents

- Introduction
- Background
- Object and class concepts
- Link and association concepts
- Generalization and inheritance
- Navigation of class models
- Summary
Introduction

- Object-oriented modeling and design
  - Way of thinking about problems using models organized around real-world concepts
- Class modeling
  - Describe the static data of objects and their relationships to one another
    - Define each concept
    - Present the corresponding UML notation
Model

- Abstraction of something for purpose of understanding it before building it

Viewpoints

- **Class model**
  - Structure of objects in a system (class diagram)

- State model
  - Aspects of an object that change over time (state diagram)

- Interaction model
  - Objects in a system cooperate to achieve broader results (Use case/sequence/activity diagram)
Object
- Concept, abstraction, or thing with identity
- Proper nouns or specific reference in problem description

Class
- Group of objects with
  - Same properties (attributes)
  - Behavior (operations)
  - Kinds of relationships
  - Semantics
Class diagrams
- Graphic notation for modeling classes and their relationships

Object diagrams
- Graphic notation for individual objects and their relationships

Rookie
- TaeHoon: Rookie
- SeokJoong: Rookie

Class

Objects

Rookie
- Value: piece of data
- Attribute: value for each object
- Operation
  - Function or procedure that may be applied to or by objects in a class
  - All objects in a class share the same operations
### Summary of notation for class

<table>
<thead>
<tr>
<th>ClassName</th>
</tr>
</thead>
<tbody>
<tr>
<td>attributeName1 : dataType1 = defaultValue1</td>
</tr>
<tr>
<td>attributeName2 : dataType2 = defaultValue2</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>operationName1 ( argumentList1 ) : resultType1</td>
</tr>
<tr>
<td>operationName2 ( argumentList2 ) : resultType2</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>
Links and associations

- Link
  - Physical or conceptual connection among objects
  - Instance of an association

- Association
  - Group of links with common structure and common semantics
  - Set of potential links

※ Links and associations

: Verb in problem statements
- Links and associations (Cont’d)

Class diagram

(Association)

Object diagram

(Link)

Rookie
- name

Company
- name

Hoon:Rookie
- name="Hoon"

Joong:Rookie
- name="Joong"

Seok:Rookie
- name="Seok"

MS:Company
- name="MS"

SUN:Company
- name="SUN"
## Multiplicity

- Constrain the number of related objects
- Assign a multiplicity to an association end

### Class diagram

```
<table>
<thead>
<tr>
<th>Husband</th>
<th>HasWife</th>
<th>Wife</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td></td>
<td>name</td>
</tr>
</tbody>
</table>
```

```
HasWife

1 -> 1

(Multiplicity)
```

### Object diagram

```
<table>
<thead>
<tr>
<th>Hoon: Husband</th>
<th>Lyn: Wife</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=&quot;Hoon&quot;</td>
<td>name=&quot;Lyn&quot;</td>
</tr>
<tr>
<td>name=&quot;Seok&quot;</td>
<td>name=&quot;Hee&quot;</td>
</tr>
</tbody>
</table>
```

```
Hee: Wife

name="Hee"
```
Association end names

- Each end of an association can have a name
- Convenience for traversing associations

<table>
<thead>
<tr>
<th>Person</th>
<th>employee</th>
<th>employer</th>
</tr>
</thead>
<tbody>
<tr>
<td>WoksFor</td>
<td>0..1</td>
<td></td>
</tr>
</tbody>
</table>

(employee) - (Association end names)

| Taehoon  | MS       |
| Seokjoong | MS       |
| Youngseok | IBM      |
- **Ordering**
  - Indicate an ordered set of objects
    - Provide an explicit order
  - Permitted only for binary associations

```
<table>
<thead>
<tr>
<th>Screen</th>
<th>{ordered}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VisibleOn</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Window</td>
</tr>
</tbody>
</table>
```

Bags and Sequences

- Permit multiple links for a pair of objects
  - Bag
    - Collection of elements with *duplicates* allowed
  - Sequence
    - Ordered bag
- Permitted only for binary associations

Itinerary * {sequence} * Airport
- **Association classes**
  - Derive identity from instances of the constituent classes
  - Have attributes and operations
  - Participate in associations

![Diagram showing association classes and their attributes](image_url)
Qualified Associations

- Disambiguate the objects for a “many” association.

- Select among the target objects, reducing the effective multiplicity, from “many” to “one”.

![Diagram showing a qualified association between Bank and Account with multiplicity 1..1.](image)
Definition

- Generalization
  - Relationship between a class (the *superclass*) and one or more variations of the class (the *subclass*)

- Inheritance
  - Each subclass has the features of its superclass

Use of generalization

- Support for polymorphism
- Structure the description of objects
- Enable reuse of code
Overriding features

- Refines and replaces
  (Subclass feature → superclass feature)
  - To specify behavior that depends on the subclass
  - To tighten the specification of a feature
  - To improve performance
Generalization and inheritance (3/3)

Vehicle
- engine
- door
- wheel
- displacement
- move

Tank
- gun
- cannon
- fire
- move

Truck
- box
- bring

Crane
- jack
- lift

(Generalization)

(Overriding Feature)
Navigation

- Express the behavior of navigating among classes
  - Exercise a model and uncover hidden flaws and omissions
- Traverse the constructs in class models with OCL

Object Constraint Language (OCL)

- Attributes
  - Traverse from an object to an attribute value
    - *Vehicle.displacement*
Object Constraint Language (OCL) (Cont’d)

Operations
- Invoke an operation for an object
  - Tank. Fire
- Operate on entire collections
  - Vehicle. Tank. gun -> sum ()

Simple association
- Traverse an association to a target end
  - Person. employee

Qualified association
- Make a more precise traversal
  - Bank. accountNumber [060411]
Object Constraint Language (OCL) (Cont’d)

- Association class
  - Find the constituent objects or the multiple links of an association class

- Generalization
  - Traverse a generation hierarchy

- Filters
  - Filter the objects in a set
    - \( \text{Vehicle} . \text{displacement} \rightarrow \text{select ( amount > cc 2000 )} \)
Summary

- Class model
  - Static data structure of objects and their relationships to one another

- Concepts
  - Object: concept, abstraction, identified thing
  - Class: group of objects (attribute, behavior, relations)
  - Link and association: set of potential links
  - Generalization: relationship between the superclass and the subclasses
  - Navigation of class model: behavior of navigation among classes
The end
Thank you