Software Lifecycle Activities

From Use Cases to Objects

Definition: Object Modeling
Main goal: Find the important abstractions
What happens if we find the wrong abstractions?
• Iterate and correct the model
Steps during object modeling:
• 1. Class identification (based on the fundamental assumption that we can find abstractions)
• 2. Find the attributes
• 3. Find the methods
• 4. Find the associations between classes (Order of steps is secondary, only a heuristic)

Do UML associations have direction?
An association between two classes is by default a bi-directional mapping.
• Class A can access class B and class B can access class A
• Both classes play the agent role.

Aggregation
• Models "part of" hierarchy
• Useful for modeling the breakdown of a product into its component parts (sometimes called bill of materials (BOM) by manufacturers)
• UML notation: Like an association but with a small diamond indicating the assembly end of the relationship.
November 10, 2016
CS410 – Software Engineering
Lecture #18: UML II

Inheritance
- Models "kind of" hierarchy
- Powerful notation for sharing similarities among classes while preserving their differences
- UML Notation: An arrow with a triangle

Aggregation vs. Inheritance
Both associations describe trees (hierarchies).
- Aggregation tree describes “part-of” relationships (also called and-relationship)
- Inheritance tree describes “kind-of” relationships (also called or-relationship)

Aggregation relates instances (involves two or more different objects)
Inheritance relates classes (a way to structure the description of a single object)

Other Associations
"Uses":
- A subsystem uses another subsystem (System Design)
"Contains":
- Sometimes called “spatial aggregation"
- ... contains ...
- Example: A UML package contains another UML package

Parent/child relationship:
- ... is father of ...
- ... is mother of ...

Seniority:
- ... is older than ...
- ... is more experienced than ...

Object Types
Entity Objects:
- Represent the persistent information tracked by the system (Application domain objects, "Business objects")

Boundary Objects:
- Represent the interaction between the user and the system

Control Objects:
- Represent the control tasks performed by the system

Example: 2BWatch Objects
- UML provides several mechanisms to extend the language
- UML provides the stereotype mechanism to present new modeling elements
Roles

- A role name is the name that uniquely identifies one end of an association.
- A role name is written next to the association line near the class that plays the role.

When do you use role names?
- Necessary for associations between two objects of the same class
- Also useful to distinguish between two associations between the same pair of classes

When do you not use role names?
- If there is only a single association between a pair of distinct classes, the names of the classes serve as good role names

Example of Role

Problem Statement:
A person assumes the role of repairer with respect to another person, who assumes the role of inspector with respect to the first person.

<table>
<thead>
<tr>
<th>Person</th>
<th>Create Workorders</th>
<th>Person</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>inspector</td>
</tr>
<tr>
<td></td>
<td></td>
<td>repairperson</td>
</tr>
</tbody>
</table>

Roles in Associations

Client Role:
- An object that can operate upon other objects but that is never operated upon by other objects.

Server Role:
- An object that never operates upon other objects. It is only operated upon by other objects.

Agent Role:
- An object that can both operate upon other objects and be operated upon by other objects. An agent is usually created to do some work on behalf of an actor or another agent.

Qualification

The qualifier improves the information about the multiplicity of the association between the classes. It is used for reducing 1-to-many multiplicity to 1-1 multiplicity.

Example: Without qualification, a directory has many files. A file belongs to only one directory.

<table>
<thead>
<tr>
<th>Directory</th>
<th>File</th>
<th>filename</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1*</td>
</tr>
</tbody>
</table>

With qualification: A directory has many files, each with a unique name.

<table>
<thead>
<tr>
<th>Directory</th>
<th>File</th>
<th>filename</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0..1</td>
</tr>
</tbody>
</table>