Object Modeling

Activities during Object Modeling
Main goal: Find the important abstractions
• 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
       • Goal: get the desired abstractions
       • Order of steps secondary, only a heuristic
     • What happens if we find the wrong abstractions?
       • We iterate and revise the model

Class Identification
• Approaches
  • Application domain approach
    • Ask application domain experts to identify relevant abstractions
  • Syntactic approach
    • Start with use cases
    • Analyze the text to identify the objects
    • Extract participating objects from flow of events
  • Design patterns approach
    • Use reusable design patterns
  • Component-based approach
    • Identify existing solution classes.

There are different types of Objects
• Entity Objects
  • Represent the persistent information tracked by the system (Application domain objects, also called “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 Modeling
To distinguish different object types in a model we can use the UML Stereotype mechanism

Naming Object Types in UML
• UML provides the stereotype mechanism to introduce new types of modeling elements
  • A stereotype is drawn as a name enclosed by angled double-quotes (“guillemets”) (<, >) and placed before the name of a UML element (class, method, attribute, ….)
  • Notation: <<String>>Name

Entity Objects
• Year
• Month
• Day

Control Object
• ChangeDate
• Button

Boundary Objects
• LCDDisplay

Entity Object
• <<Entity>>
• <<Entity>>
• <<Entity>>

Control Object
• <<Control>>
• <<Control>>

Boundary Object
• <<Boundary>>
• <<Boundary>>
• <<Boundary>>
UML is an Extensible Language

• Stereotypes allow you to extend the vocabulary of the UML so that you can create new model elements, derived from existing ones.

Examples:
  • Stereotypes can also be used to classify method behavior such as <<constructor>>, <<getter>> or <<setter>>.
  • To indicate the interface of a subsystem or system, one can use the stereotype <<interface>> (Lecture System Design).
  • Stereotypes can be represented with icons and graphics:
    • This can increase the readability of UML diagrams.

Icons for Stereotypes

• One can use icons to identify a stereotype.
  • When the stereotype is applied to a UML model element, the icon is displayed beside or above the name.

Object Types allow us to deal with Change

• Having three types of object leads to models that are more resilient to change.
  • The interface of a system changes more likely than the control.
  • The way the system is controlled changes more likely than entities in the application domain.

• Object types originated in Smalltalk:
  • Model, View, Controller (MVC)
  • Model <-> Entity Object
  • View <-> Boundary Object
  • Controller <-> Control Object

• Next topic: Finding objects.

Finding Participating Objects in Use Cases

• Pick a use case and look at flow of events.
• Do a textual analysis (noun-verb analysis):
  • Nouns are candidates for objects/classes.
  • Verbs are candidates for operations.
  • This is also called Abbott's Technique.

Example for using the Technique

Flow of Events:
• The customer enters the store to buy a toy.
• It has to be a toy that his daughter likes and it must cost less than $50.
• He tries a videogame, which uses a data glove and a head-mounted display. He likes it.
• An assistant helps him.
• The suitability of the game depends on the age of the child.
• His daughter is only 3 years old.
• The assistant recommends another type of toy, namely the boardgame "Monopoly".

Mapping parts of speech to model components (Abbott’s Technique)

<table>
<thead>
<tr>
<th>Example</th>
<th>Part of speech</th>
<th>UML model component</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Monopoly”</td>
<td>Proper noun</td>
<td>object</td>
</tr>
<tr>
<td>Toy</td>
<td>Improper noun</td>
<td>class</td>
</tr>
<tr>
<td>Buy, recommend</td>
<td>Doing verb</td>
<td>operation</td>
</tr>
<tr>
<td>is-a</td>
<td>being verb</td>
<td>inheritance</td>
</tr>
<tr>
<td>has an</td>
<td>having verb</td>
<td>aggregation</td>
</tr>
<tr>
<td>must be</td>
<td>modal verb</td>
<td>constraint</td>
</tr>
<tr>
<td>dangerous</td>
<td>adjective</td>
<td>attribute</td>
</tr>
<tr>
<td>enter</td>
<td>transitive verb</td>
<td>operation</td>
</tr>
<tr>
<td>depends on</td>
<td>intransitive verb</td>
<td>Constraint, class, association</td>
</tr>
</tbody>
</table>
The customer enters the store to buy a toy. It has to be a toy that his daughter likes and it must cost less than 50 Euro. He tries a videogame, which uses a data glove and a head-mounted display. He likes it.

An assistant helps him. The suitability of the game depends on the age of the child. His daughter is only 3 years old. The assistant recommends another type of toy, namely a boardgame. The customer buy the game and leaves the store.
Summary

- System modeling
  - Functional modeling + object modeling + dynamic modeling
- Functional modeling
  - From scenarios to use cases to objects
- Object modeling is the central activity
  - Class identification is a major activity of object modeling
  - Easy syntactic rules to find classes and objects
  - Abbot’s Technique
- Class diagrams are the “center of the universe” for the object-oriented developer
  - The end user focuses more on the functional model and usability.