The Unified Modeling Language (UML)

- Booch and Rumbaugh started working towards a unified modeling language (UML) in 1994 under the auspices of Rational Inc. They were later joined by Jacobson.
- UML only offers a notation, not a methodology for modeling (as various OOA techniques do).
- Combines Jacobson’s use cases with Booch and Rumbaugh concepts for object modeling, along with statecharts.
- UML has been adopted by the Object Management Group (OMG) as an (object) modeling standard. OMG UML 1.0 is the first version of this new modeling standard.

Where Do We Start? Use Cases

- Use cases describe how the system-to-be (or any artifact under design, for that matter!) from a user’s perspective.
- They answer the question: How will the artifact be used, once it is built?
- Used to show the functions to be supported.

Use Cases

- A use case is a function the new system needs to support.
- Each use case is a sequence of steps performed by an actor and the system through a dialogue.
- To find use case, examine each actor and her needs.

Use Case Diagrams

- Use case diagrams are created to capture the relationships between actors and use cases.
**Notation for Use Cases**

- **Actor**: Staff contact
- **Communication association**: System/Artifact boundary
- **Use case**: Change client contact

---

**Agate is an Advertising Company**

...which puts together advertising campaigns for client companies. Here is the breakdown of their staff:

<table>
<thead>
<tr>
<th>Department</th>
<th>Positions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Direction</strong></td>
<td>1 Campaign, 1 Creative, 1 Admin, 1 Finance</td>
</tr>
<tr>
<td><strong>Admin</strong></td>
<td>1 Office mgr, 3 Admin asst, 1 Creative Manager</td>
</tr>
<tr>
<td><strong>Campaigns Mgt</strong></td>
<td>2 Campaign managers, 3 Campaign marketers, 1 Editor in Chief</td>
</tr>
<tr>
<td><strong>Edition</strong></td>
<td>1 Filing clerk, 2 Editors, 4 Copy writers</td>
</tr>
<tr>
<td><strong>Graphic</strong></td>
<td>6 Graphic designers, 2 Photographers</td>
</tr>
<tr>
<td><strong>IT</strong></td>
<td>1 IT manager, 1 Network admin, 1 System admin</td>
</tr>
<tr>
<td><strong>Accounting</strong></td>
<td>1 Accountant manager, 2 Accounts clerks</td>
</tr>
<tr>
<td><strong>Production</strong></td>
<td>5 Copy writers, 2 Photographers</td>
</tr>
<tr>
<td><strong>Accounts</strong></td>
<td>1 IT manager, 1 Media librarian, 1 Knowledge worker</td>
</tr>
<tr>
<td><strong>Document</strong></td>
<td>1 Media librarian, 1 Resource librarian</td>
</tr>
<tr>
<td><strong>System Admin</strong></td>
<td>1 Analyst, 2 Purchasing assistants</td>
</tr>
<tr>
<td><strong>Sys Admin</strong></td>
<td>1 Analyst, 2 Purchasing assistants</td>
</tr>
</tbody>
</table>

---

**Agate Case Study**

- **Add new staff member**
- **Add new staff grade**
- **Calculate staff bonuses**
- **Change grade for staff member**
- **Change rate for staff grade**

---

**Finding Actors**

- Actors can be identified by answering the following:
  - Who will be a primary user of the artifact?
  - Who will be supported?
  - Who will maintain, administrate the artifact?
  - What hardware does the system need?
  - Which other systems does it interact with?
  - Who or what has an interest in the results that the artifact produces?
- **Tip**: don't consider only the users who directly use the artifact, but also others who need services from the artifact!

---

**Finding Use Cases**

For each actor, ask the following questions:

- Which functions does the actor require from the artifact? What does the actor need to do?
- Does the actor need to read, create, destroy, modify, or store some kinds of information in the artifact?
- Does the actor have to be notified about events in the artifact? Or, does the actor need to notify the artifact about something? What do those events require in terms of artifact functionality?
- Could the actor's daily work be simplified or made more efficient through new functions provided by the artifact?
Documenting Use Cases

- For each use case, prepare a "flow of events" document, written from an actor's point of view.
- The document details what the system must provide to the actor when the use case is executed.
- Typical contents
  - How the use case starts and ends;
  - Normal flow of events;
  - Alternate flow of events;
  - Exceptional flow of events;

Use Cases for a Meeting Scheduling System

Initiator

ValidateUser

Participant

Constraints

Schedule

ScheduleMtg

GenerateSchedule

Edit

Constraints

Use Cases for a Car

Driver

GasAttendant

Mechanic

Drive

FixCar

TurnOnEngine

CheckOil

FillUp

FixCarOnTheRoad

FixCar

Additional Readings