Unified Modeling Language (UML)

An Overview
• UML is a modeling *notation*
    • now in 2.5.1 (December 2017)
  ▪ mature
  ▪ based on
    • notations previously used in the software engineering OOA&D – Bracha .. ,
      Booch (Ada at Rational), Rumbaugh OMT (at GE), Jacobson (use cases at
      Ericsson)
  ▪ suitable for Object-oriented
  ▪ design
  ▪ implementation
UML Diagrams

- Defines a number of diagrams
UML Diagrams – Class Diagram

ClassA

- name: String
- shape: Rectangle
  + size: Integer [0..1]
- area: Integer {readOnly}
- height: Integer = 5
- width: Integer

ClassB

- id {redefines name}
- shape: Square
- height = 7
- width
UML Diagrams – Component Diagram

UML Diagrams

• Defines a number of diagrams

Figure from: OMG, “Unified Modeling Language: Superstructure, Version 2.1.1”
UML Diagrams – Sequence Diagram

UML Diagrams – Communication Diagram

UML Diagrams – State Machine Diagram

Figures from: OMG, "Unified Modeling Language: Superstructure, Version 2.1.1"
UML Diagrams – Use Case Diagram

• CoCoME trading system...
Unified Modeling Language (UML)

Class Diagrams
Class Diagrams

• A class diagram shows
  ▪ classes
  ▪ relations among classes
  ▪ generalization
  ▪ associations (with multiplicities, names)
    • special case: aggregation and composition

• It’s definitely good to be able to read class diagrams!
**Class Diagrams – Class**

- Shows a class with attributes with explicitly marked visibility

<table>
<thead>
<tr>
<th>Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ size: Area = (100, 100)</td>
</tr>
<tr>
<td># visibility: Boolean = true</td>
</tr>
<tr>
<td>+ defaultSize: Rectangle</td>
</tr>
<tr>
<td>- xWin: XWindow</td>
</tr>
<tr>
<td>display()</td>
</tr>
<tr>
<td>hide()</td>
</tr>
<tr>
<td>- attachX(xWin: XWindow)</td>
</tr>
</tbody>
</table>
Class Diagrams – Notation

• Class compartments
  ▪ top
    • name and annotations
    • stereotypes, superclass,...
  ▪ attributes
  ▪ operations
  ▪ additional compartments
    • added by extensions, e.g.
    • EJB finder/business/activation compartments...

<table>
<thead>
<tr>
<th>Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ size: Area = (100, 100)</td>
</tr>
<tr>
<td># visibility: Boolean = true</td>
</tr>
<tr>
<td>+ defaultSize: Rectangle</td>
</tr>
<tr>
<td>- xWin: XWindow</td>
</tr>
<tr>
<td>display()</td>
</tr>
<tr>
<td>hide()</td>
</tr>
<tr>
<td>- attachX(xWin: XWindow)</td>
</tr>
</tbody>
</table>
Class Diagrams – Notation

- class name: bold
- abstract class (or method): italics
- class scope (aka static): underlined (instance-scope otherwise)
- visibility (attributes, operations)
  - + public visibility
  - # protected visibility
  - - private visibility
  - ~ package visibility

```
Window
+ size: Area = (100, 100)
# visibility: Boolean = true
+ defaultSize: Rectangle
- xWin: XWindow

display()
hide()
- attachX(xWin: XWindow)
```
Class Diagrams – Example

Figure from: Larman, C., “Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process”
Associations

• Association
  - Semantic relationship
  - At least two ends
    • May be navigable
  - May have assigned aggregation type
    • Shared
    • Composite
Profiles & stereotypes

• UML can be extended by defining so called profiles and stereotypes
• This allows assigning particular roles and associating additional attributes to existing UML blocks
Meta-models

Modeling models...
Models and meta-models

• Model is what we specify
  ▪ e.g. data model of an application modeled in UML

• But what the language, which we use for modeling?
  ▪ The language itself can be again described by a model
  ▪ This model is called meta-model

• Meta-modeling constructs
  ▪ Classes
  ▪ Associations
  ▪ DataTypes
  ▪ Packages
  ▪ Constraints

XML metamodel
Modeling hierarchy

M3 – MOF definition

M2 – metamodel

generated

M1 – model

Repository of

M0 – application

MOF model

IDL metamodel

IDL interfaces

CORBA Objects

UML metamodel

UML diagrams

Objects
Modeling hierarchy

Hard-wired Meta-metamodel

MetaModel ("RecordTypes",
MetaClass ("Record",
[ MetaAttr ("name", String),
  MetaAttr ("fields", List <"Field">) ]
MetaClass ("Field", ...)

Record ("StockQuote",
[ Field ("company", String)
  Field ("price", FixedPoint) ])

StockQuote ("Sunbeam Harvesters", 98.77)
StockQuote ("Ace Taxi Cab Ltd", 12.32)

...
How many meta layers?

• The minimal number of layers is two

• Examples
  ▪ 2 layers
    • generic reflective systems - Class/Objecs
  ▪ 3 layers
    • relational database systems - SysTable/Table/Row
  ▪ 4 layers
    • UML, MOF specification - MOF/UML/User Model/User Object
      ▪ MOF is a UML-like language for meta-modeling (i.e. only core constructs compared to UML)
Representing MOF

• MOF has no own graphical representation
  ▪ Uses UML
  ▪ Relies on the fact that UML and MOF have a lot of similarities

• Brain exercise:
  ▪ UML is M2-model
    • Thus, it is an instance of MOF
  ▪ UML is used to represent MOF models
  ▪ MOF is modeled in MOF
    • Thus, MOF is formalized by UML
The CMOF package reuses the abstract syntax defined in the InfrastructureLibrary for UML, MOF.

Figure from: OMG, "Meta Object Facility (MOF) Core Specification, Version 2.0"