

Information modeling basics

Ferd van Odenhoven

Fontys Hogeschool voor Techniek en Logistiek

May 27, 2015

Modeling 1

- Websites:
 - General public: <http://mod1.fontysvenlo.org>
 - Student Information: <https://portal.fontys.nl/instituten/tbm/students/sebi/S2/MOD1/default.aspx>
- Lecturer: F.vanOdenhoven@fontys.nl
- Book: Martin Fowler: UML Distilled, 3rd ed.
- Week 1-7: hand in work; will be discussed in class and or during practical hours.
- Practical from week 8 on: work on a case, making a design and partly test-driven realizing in Java.
- Tooling:
 - In MOD1: Umlet, just for drawing diagrams
 - Download at <http://umlet.com/>

Object orientation



Software structure:

- Functional structure
- Data oriented structure
- *Object oriented*: function and data

An object is self-contained in a system.

The structure of the system consists of communication between connected objects.

Most important aspects/goals of an object:

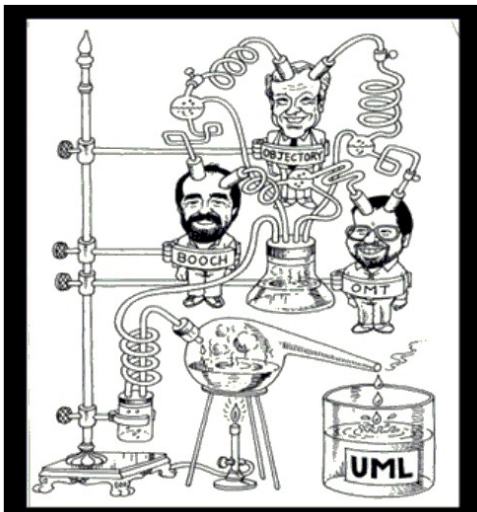
- **Encapsulation**
- **Dedicated responsibilities**

Unified Modeling Language

UML was borne in the nineties through the association of the “3 Amigos”:

- OMT (James Rumbaugh)
- Booch methode (Grady Booch)
- OOSE (Ivar Jacobsen)
- UML was the proposition of Rational to the OMG. This proposition has been supplemented with elements from other propositions (Object Constraint Language).
- **UML is a graphical modeling language, not a development method.**
- The development method we partly use is an agile development that uses a test driven development technique.
- Internet: <http://www.agilemodeling.com/>

'Tres Amigos' of the UML



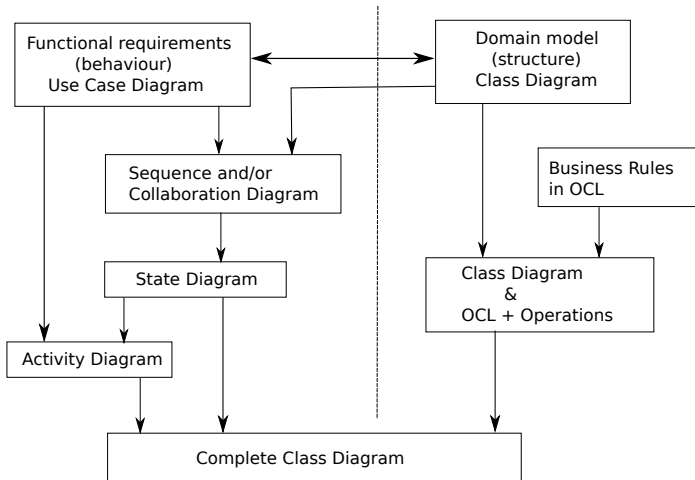
Unified Modeling Language & Process

Overview of the diagram types

- Software Requirements:
 - Use Case diagram, Class Diagram, Activity Diagram
- Static model:
 - Class Diagram , Object diagram
- Dynamic model:
 - Sequence Diagram, Collaboration Diagram, Activity Diagram
- Architecture:
 - Package Diagram, Component Diagram
- Implementation:
 - Component Diagram, Deployment Diagram

A process overview / Coherence of the diagrams

Unified Modeling Language



Project management ⇒ SEN1 / BUMA

Project management and object oriented development: iterative and incremental with timeboxes.

- Whole process is divided into increments
- every increment will be implemented in a time box.
- In every time box with milestones “traditional” phases can be looped: conceptual phase, analysis phase, design phase, implementation phase, test phase and deployment phase.
- Repeat the process in each increment: iterations

This is just an example, there are many variants. Popular is *Scrum* which is an agile development model (easy to adjust the requirements).

Three Modeling approaches for using UML

- As a sketch, typically in the analysis phase
 - selectivity: important issues
 - informal and dynamic
 - focus on communication
- As a blueprint, typically in the design phase
 - sufficiently complete
 - detailed up to interfaces
 - aimed at realisation
 - tools needed

(round trip(code↔diagrams), tripless(code ≡ repository))
- As a programming language, implementation oriented
 - compile to executable code
 - sophisticated tools needed
 - Model Driven Architecture e.g.
openArchitectureWare.org (colloquium 3 december 2008)
now moved to Eclipse Modeling Project

Modeling approach

How do we analyse the domain and which is our starting diagram?

Possibilities are:

- A domain expert exists: interview!
- A preliminary textual description exists.

Textual analysis

- Search for classes: Every noun belonging to a subject in the domain is a candidate class.
- Which relations exist between classes (essential objects in the domain description)?
- Classes make use of other classes: verbs, methods!
- Relations are based upon facts that can be stated.

Modeling: perspectives

- Conceptual perspective
 - domain study and or analysis
 - building a vocabulary
- Software perspective
 - implementation oriented
 - elements of a software system
- Communication perspective
 - to describe some of the requirements (stakeholder)
 - documentation of the design results (team)

An example: Library

A textual description as a starting point.

- The public library Venlo has four locations: Belfeld, Tegelen, Blerick en Venlo. To borrow a book, one must be member of the library. Membership is valid for one year, starts at the month of subscription and one obtains a member card.
- With a subscription every member can make use of the central library, as well as the locations at Blerick, Belfeld, Tegelen. Books, audio cassettes, magazines, video tapes, dvd disks, cd-roms and daisy-rom can be borrowed. Per lending period a certain amount is to be paid, depending on the kind of media borrowed.
- The lending fee is free for members younger than 18. Members aged 18 and above pay 20 cents per book. The lending period for a book is three weeks.

An example: Library

- Extending a lending period can be done twice and per extension the costs are the lending sum for one period.
- Extension of the lending period is **not** possible if a reservation has been made for the specific media item. If a book has been brought back after the lending period has expired, a fine of 10 cents per day is charged. This is also the case if media were lent for half the usual price.
- Lending media or extending the lending period is not possible if the members's debt at the library is larger than 25 €.

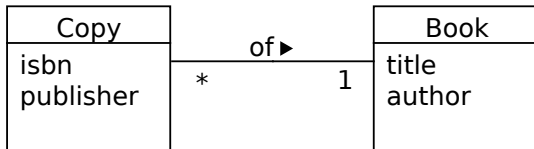
Domain analysis.

- Domain model, diagram type: Class diagram
 - Which are the domain objects?
 - Relations between the domain objects.
- Activity Diagrams
 - Specification of business processes.
 - Contain the use cases.
- Use cases
 - Specification of user interaction.
 - Specification of system actions.
- Data model dictionary
 - Specification of each object with its data and actions, independent of system.
 - No relations between objects.

Domain analysis: books and copies.

- The author 'Joseph O'Connor' has written a book with the title 'Star of the Sea'.
- The library possesses 5 copies of that book.

The first information point was not obtained from the library! We do need this trivial fact for our domain analysis. Book and copy are different objects.



Use Case example

UC-1	
Name:	<i>Lend a book</i>
Actor:	<i>Librarian</i>
Description:	<i>member of the library borrows a book</i>
Pre-condition:	<i>member has valid membership card on him(her)</i>
Scenario:	<ol style="list-style-type: none"><i>1. Actor takes book information code.</i><i>2. Actor takes info data membership card</i><i>3. Actor feeds book code and member code into system</i><i>4. System respond message: "OK"</i><i>5. System adds lending fee to debt stored in account properties of member.</i><i>6. Ticket is printed with: book title, book code, date of transaction, expiry date, lending fee.</i><i>7. System respond message: "Action successfully completed"</i>
Result:	<i>Member has borrowed the book successfully from the Library.</i>
Exceptions:	<ol style="list-style-type: none"><i>4. Member is not allowed to take the book with him because his debt at the library is too high. System respond message: "Debt too high". Use case ends here.</i>

Use Case example

UC-1	
Name:	<i>Lend a book</i>
Actor:	<i>Librarian</i>
Description:	<i>member of the library borrows a book</i>
Pre-condition:	<i>member has valid membership card on him(her)</i>
Scenario:	<ol style="list-style-type: none"><i>1. Actor takes book information code.</i><i>2. Actor takes info data membership card</i><i>3. Actor feeds book code and member code into system</i><i>4. System respond message: "OK"</i><i>5. System adds lending fee to debt stored in account properties of member.</i><i>6. Ticket is printed with: book title, book code, date of transaction, expiry date, lending fee.</i><i>7. System respond message: "Action successfully completed"</i>
Result:	<i>Member has borrowed the book successfully from the Library.</i>
Exceptions:	<ol style="list-style-type: none"><i>4. Member is not allowed to take the book with him because his debt at the library is too high. System respond message: "Debt too high". Use case ends here.</i>

Data model dictionary

- In a larger project it is good practice to write a *data model dictionary*. In it you define the precise names of the objects or entities that make up the domain.
- All attributes are added and possible actions on these objects too.
- This dictionary can be used for: an object model for the *gui-design*, the data model for a *database* and the *domain model* for the class diagram.
- It will greatly improve the *meetings* with the domain expert.

Model dictionary example

Data types Rental and Book:

- Rental

- *attributes*

- date
- book
- member

- *actions*

- get book
- change date
- prolong rental
- add book

- Book

- *attributes*

- author
- title
- publisher
- isbn

- *actions*

- get title
- get author
- get ISBN
- get publisher

Where to start?

- One can start with either:
 - use cases or
 - a domain model (=conceptual class diagram) or
 - with activity diagrams
- After some time you need them all.
- In the chapters to come we shall take a look at the details of both use cases and class diagrams.
- Top down (activity diagrams first) or bottom up (use cases first).
- Later on we shall use sequence diagrams to display the result.