Ingegneria del Software

Corso di Laurea in Informatica per il Management

O-O modeling and UML

Davide Rossi
Dipartimento di Informatica
Università di Bologna

Modeling

- A model is an (abstract) representation of reality.
- Models are used to capture relevant (from the point of view of the modeler) properties allowing more focused reasoning.
- Models are also useful to share knowledge.
- Creating models is a natural process.

Modeling languages

- Models are expressed with languages.
- Languages are systems of signs for encoding and decoding information.
- Modeling languages define the entities composing the modeled subject, their properties and their relationships.

O-O principles

- O-O is a paradigm that shift the focus of analysis and design from algorithms and data to objects, intended as autonomous entities with a state and a behavior.
- The main O-O principles are: abstraction, encapsulation, inheritance and polymorphism.

O-O principles

- **Abstraction**: focus on essential characteristics (w.r.t. the perspective of the viewer).
- **Encapsulation**: hide the details (your status).
- **Inheritance**: behavior and state can be specialized.
- **Polymorphism**: behavior depends on who you are.

Inheritance

- Reuse of an existing object (prototype-based) or an existing class (class-based).
- In some object models class-based inheritance implies sub-typing:
  sub-class *is a* base-class.
- In some object models sub classes can override (*specialize*) part of the behavior of the base class.

Polymorphism

- In languages supporting class-based inheritance as a sub-typing mechanism, an object instance of a sub-class can be *used* whenever a base-class type is required (sub-type polymorphism).
- In Java, for example, object references can be polymorphic.
- When a behavior is activated the outcome depends on the type of the object, not on the type of the reference (dynamic dispatching).

Object oriented modeling with UML

- UML is a modeling language
- UML assumes on object-oriented approach for both analysis and design.
- Analysis focuses on the problem domain.
- Design focuses on the solution domain.
- UML supports both aspects (but it is agnostic with respect to how).

Brief history of UML

The adoption of object-oriented analysis and design methods in the 80s lead to the development of several modeling languages.

Between 1989 and 1994 the number of O-O methods increased from 10 to more than 50.

Prominent methods included Booch's method, Jacobson's OOSE (Object-Oriented Software Engineering) and Rumbaugh's OMT (Object-Modeling Technique).

Brief history of UML

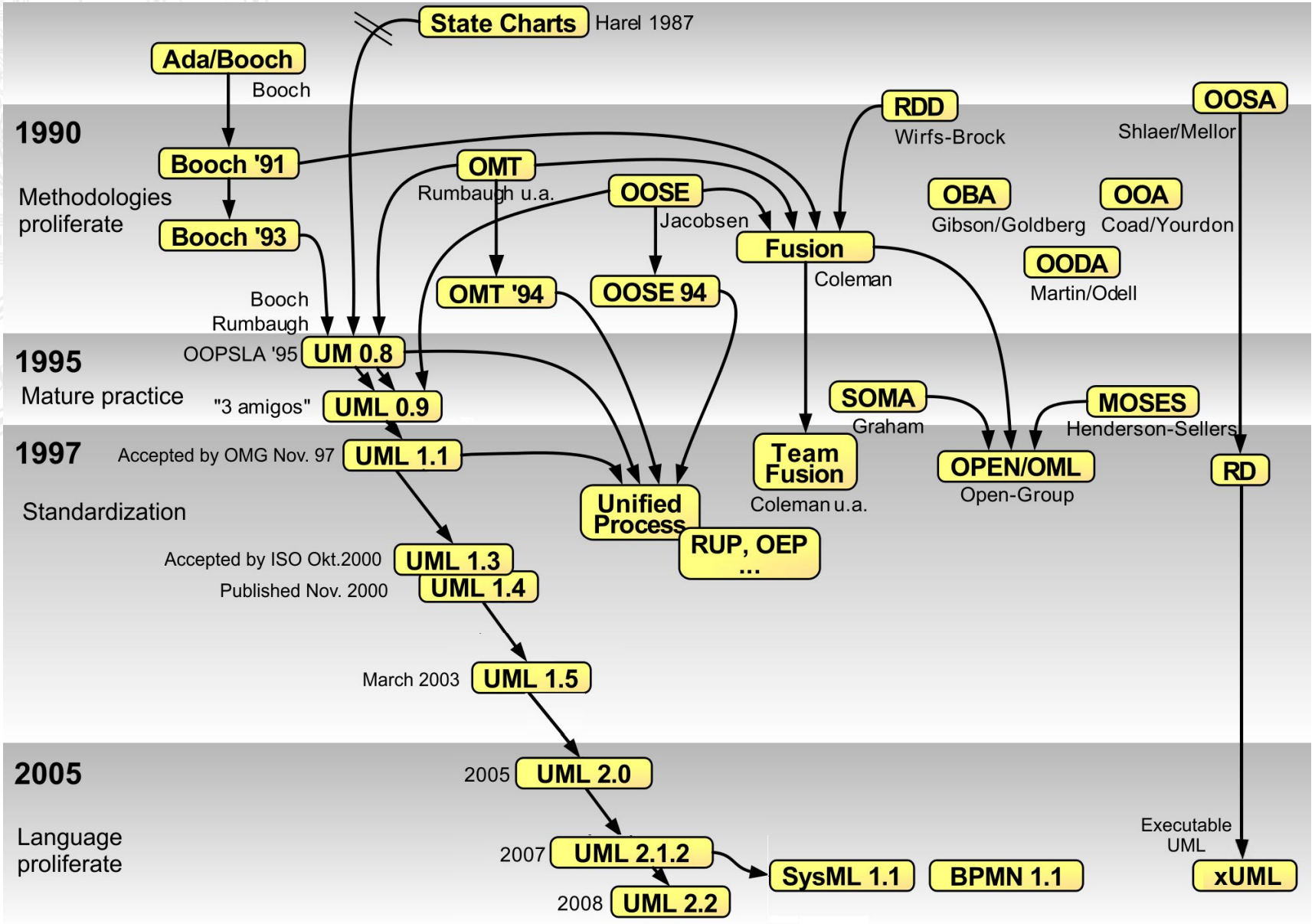
In 1994 Rumbaugh joins Booch at Rational (coming from General Electric) and started working on a unification of Booch and OMT methods. In 1995 Jacobson joined rational too (coming from Objectory).

UML (Unified Modeling Language) 0.9 was released in June 1996.

UML 1.0 was offered for standardization to the OMG in January 1997.

UML 2.0 was adopted by the OMG in early 2005.

Modeling languages history



[wikipedia]

UML

- UML is a modeling language for software-intensive systems.
- It is a graphical, semi-formal language that is used to specify, visualize, construct and document software artifacts.
- Artifacts are the products of a software development process.

What UML is not

- UML is not a software development process.
- UML in independent from the project domain, from the development process, from specific programming languages and specific development tools.

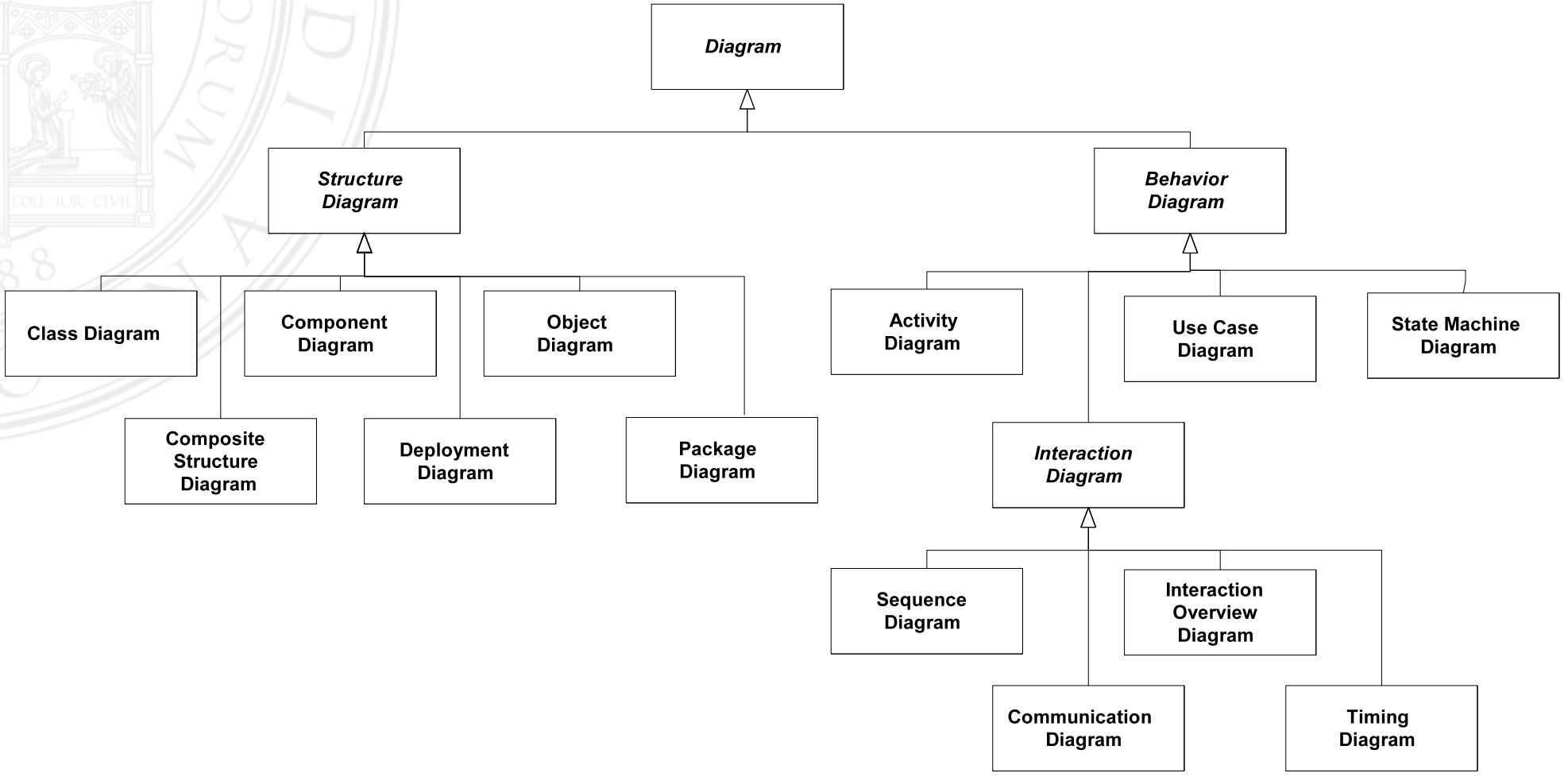
Syntax and semantics in UML

- UML is a language, not just a graphical notation; it has syntactic rules and semantic rules.
- Syntactic rules define how to create valid diagrams.
- Semantic rules define how to create meaningful diagrams.

MOF

- UML is defined on top of an OMG modeling standard called MOF (Meta-Object Facility).
- MOF is structured in 4 levels: M0, M1, M2, M3.
- MOF-based languages (such as UML) can be serialized as defined by the XMI (XML Metadata Interchange) standard.

The 13 UML diagrams

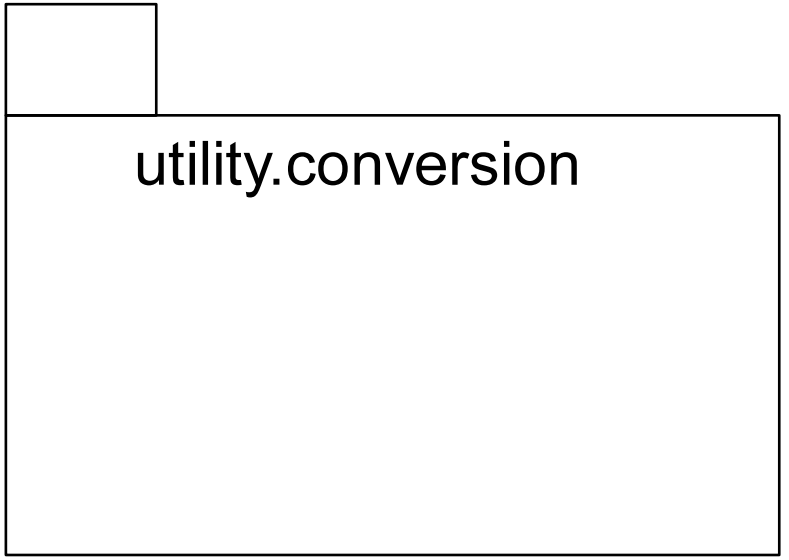


UML primitives

- A UML model includes:
  - Classifiers (sets of things);
  - Events (sets of occurrences);
  - Behaviors (sets of executions).
- Allowed element distinguish among different diagram types.

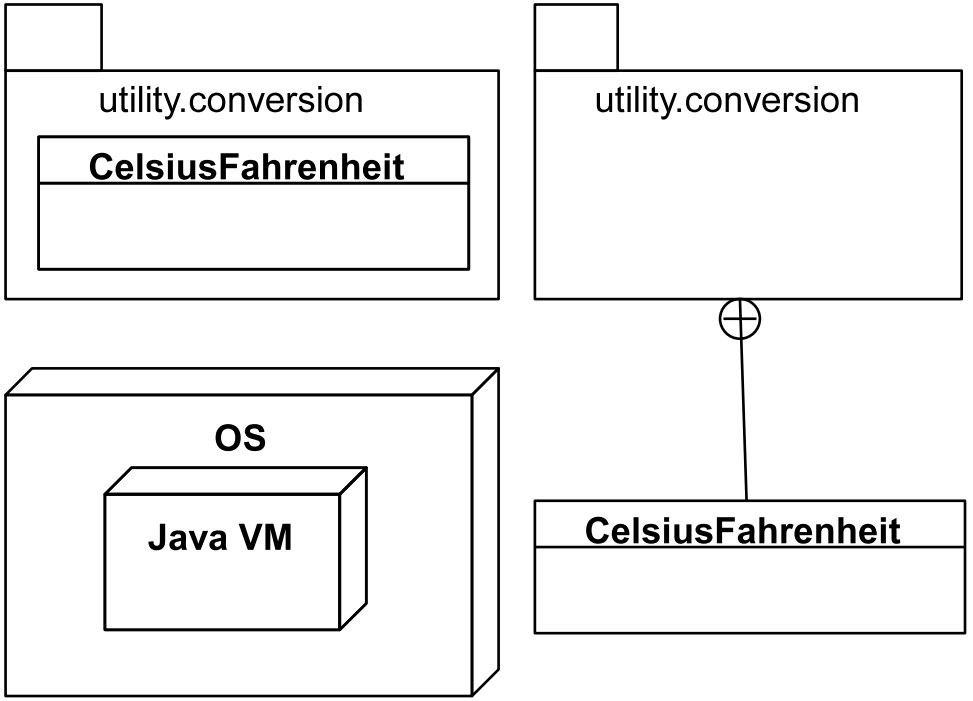
UML grouping entities

- Packages are used to group elements and give them a namespace.



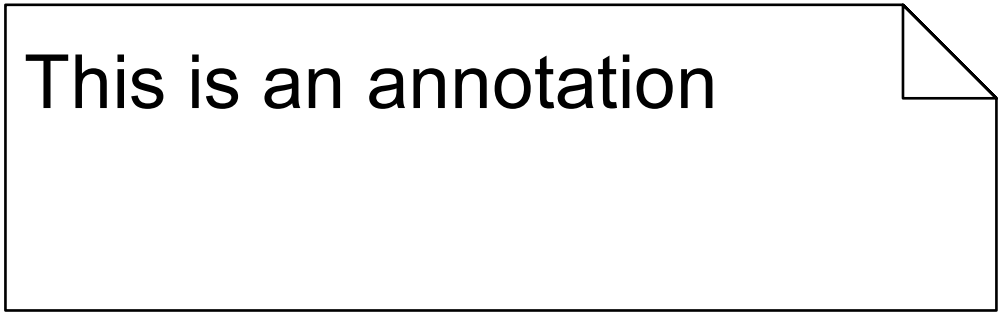
UML containment

- In UML several elements can contain other elements; creating a hierarchical structure. These relations can be depicted graphically.



UML information entities

- Annotation: improves the readability of a diagram; it has no effect on the model.



UML relationships

- Relationships correlate two or more elements in a model. They are represented as lines and can have names.
- 4 basic types of relationships:
  - Association
  - Generalization
  - Dependency
  - Realization

Stereotypes

- A stereotype is rendered as a name enclosed by guillemets « » or << >>.
- They allow designers to extend the vocabulary of UML in order to create new model elements, derived from existing ones, but that have specific properties.

OCL

- OCL (Object Constraint Language) is an OGM specification.
- It is a declarative language for describing rules that apply to Unified Modeling Language.
- It provides constraint and object query expressions.

OCL constraints

- inv: invariant
- pre: precondition
- post: postcondition
- body: a query in a context
- init: initial value in a context
- derive: define a derived attribute in a context

OCL in UML diagrams