

UNIFIED
MODELING
LANGUAGE



What is UML?

UML is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system.

It can be used with all processes, throughout the software development life cycle, and across different implementation technologies.

How can UML help a project?

Unified Modeling Language (UML) combines techniques from data modeling (entity relationship diagrams), business modeling (workflows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies.^[1]

How can UML help a project?

The Unified Modeling Language (UML) offers a standard way to visualize a system's architectural blueprints, including elements such as:

- Activities
- Actors
- Business processes
- Database schemas
- (logical) components
- Programming language statements
- Reusable software components

UML and technologies

UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language.^[2] UML aims to be a standard modeling language which can model concurrent and distributed systems.

UML models may be automatically transformed to other representations (e.g. Java) by means of QVT-like transformation languages. UML is extensible, with two mechanisms for customization: profiles and stereotypes.

The UML specification

There are four parts to the UML 2.x specification:

- The **Superstructure** that defines the notation and semantics for diagrams and their model elements
- The **Infrastructure** that defines the core metamodel on which the Superstructure is based
- The **Object Constraint Language (OCL)** for defining rules for model elements
- The **UML Diagram Interchange** that defines how UML 2 diagram layouts are exchanged

Models

It is important to distinguish between the UML model and the set of diagrams of a system. A diagram is a partial graphic representation of a system's model. The model also contains documentation that drives the model elements and diagrams (such as written use cases).

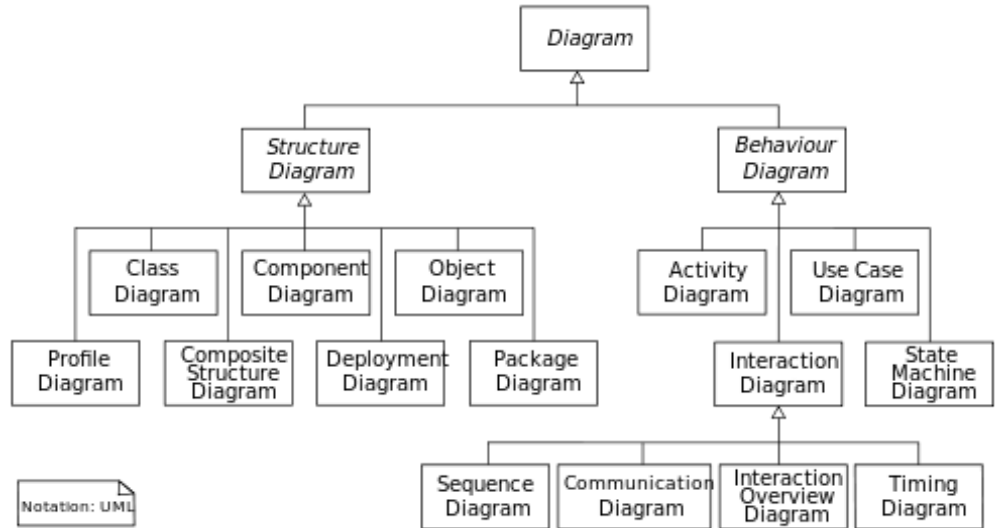
Models

UML diagrams represent two different views of a system model:[3]

- Static (or structural) view: emphasizes the static structure of the system using objects, attributes, operations and relationships. The structural view includes class diagrams and composite structure diagrams.
- Dynamic (or behavioral) view: emphasizes the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects. This view includes sequence diagrams, activity diagrams and state machine diagrams.

Diagrams

UML 2.2 has 14 types of diagrams divided into two categories. Seven diagram types represent structural information, and the other seven represent general types of behavior, including four that represent different aspects of interactions.



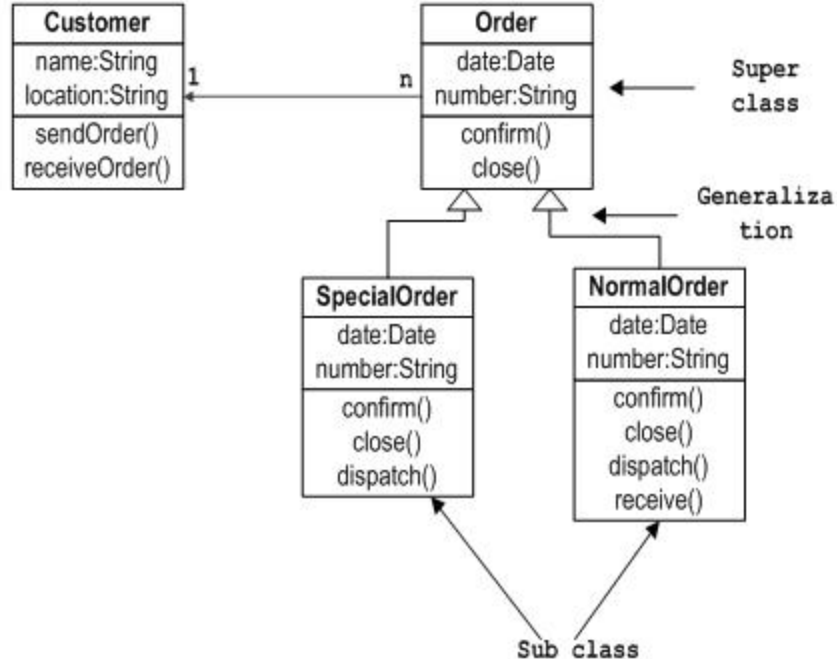
Structure diagrams

Structure diagrams emphasize the things that must be present in the system being modeled. Since structure diagrams represent the structure, they are used extensively in documenting the software architecture of software systems.

- Class diagram
- Component diagram
- Composite structure diagram
- Deployment diagram
- Object diagram
- Package diagram
- Profile diagram

Structure diagrams

Class diagram



Behavior diagrams

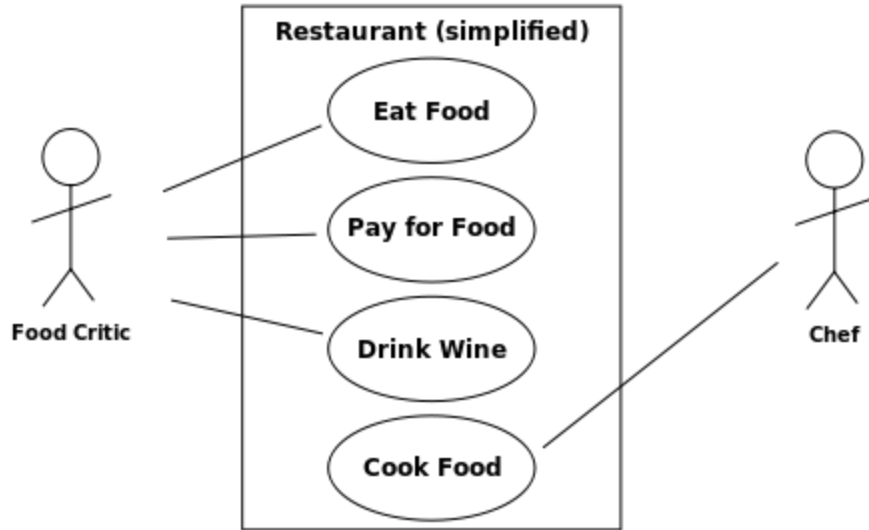
Behavior diagrams emphasize what must happen in the system being modeled. Since behavior diagrams illustrate the behavior of a system, they are used extensively to describe the functionality of software systems.

- Activity diagram
- UML state machine diagram
- Use Case Diagram
- Communication diagram*
- Interaction overview diagram*
- Sequence diagram*
- Timing diagrams*

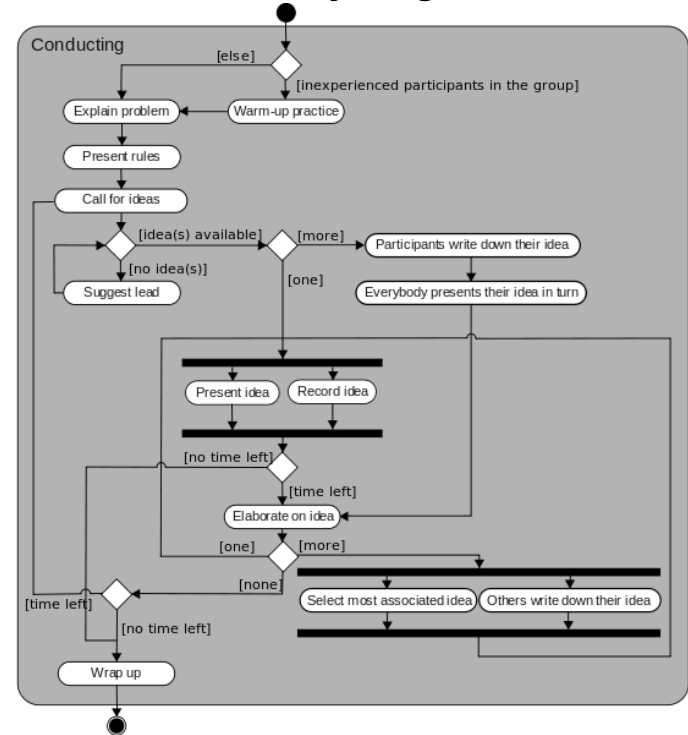
*Interaction diagrams, a subset of behavior diagrams

Behavior diagrams

Use case diagram



Activity diagram



References

- 1 Satish Mishra (1997). "Visual Modeling & Unified Modeling Language (UML) : Introduction to UML". Rational Software Corporation. Accessed 9 November 2008.
- 2 "OMG Unified Modeling Language (OMG UML), Superstructure. Version 2.4.1". Object Management Group. Retrieved 2013-03-28.
- 3 Jon Holt Institution of Electrical Engineers (2004). UML for Systems Engineering: Watching the Wheels IET, 2004, ISBN 0-86341-354-4. p.58