

- and the inheritance (*is a*): describes a hierarchy between classes

Now with the Class diagram finished, you can lean back: if you have a good UML Modelling tool you simply click on the 'Generate Code' button and it will create stubs for all the classes with methods and attributes in your favorite programming language. By the way, don't expect your manager to understand class diagrams.

UML Models and Diagrams

The Unified Modeling Language is a standardized general-purpose modeling language and nowadays is managed as a de facto industry standard by the Object Management Group (OMG).^[1] UML includes a set of graphic notation techniques to create visual models of software-intensive systems.^[2]

History

UML was invented by the *Three Amigos*: James Rumbaugh, Grady Booch and Ivar Jacobson. After Rational Software Corporation hired James Rumbaugh from General Electric in 1994, the company became the source for the two most popular object-oriented modeling approaches of the day: Rumbaugh's Object-modeling technique (OMT), which was better for object-oriented analysis (OOA), and Grady Booch's Booch method, which was better for object-oriented design (OOD). They were soon assisted in their efforts by Ivar Jacobson, the creator of the object-oriented software engineering (OOSE) method. Jacobson joined Rational in 1995, after his company, Objectory AB,^[3] was acquired by Rational.

Definition

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software-intensive system under development.^[4] UML offers a standard way to visualize a system's architectural blueprints, including elements such as activities, actors, business processes, database schemas, components, programming language statements, and reusable software components.^[5]

UML combines techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies.^[6]

Models and Diagrams

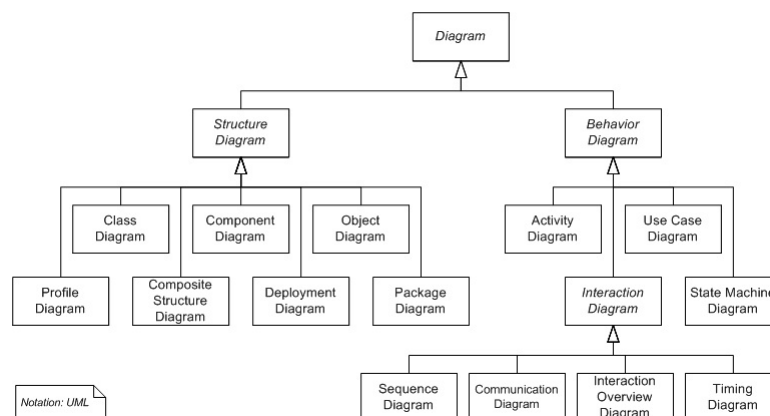
It is important to distinguish between the UML model and the set of diagrams of a system. A diagram is a partial graphic representation of a system's model. The model also contains documentation that drive the model elements and diagrams.

UML diagrams represent two different views of a system model ^[7]:

- 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 Overview

In UML 2.2 there are 14 types of diagrams divided into two categories.^[8] Seven diagram types represent structural information, and the other seven represent general types of behavior, including four that represent different aspects of interactions. These diagrams can be categorized hierarchically as shown in the following diagram:



Structure Diagrams

Structure diagrams emphasize the things that must be present in the system being modeled. Since structure