

Unified Modeling Language

Unified Modeling Language (UML) is a standardized (ISO/IEC 19501:2005), general-purpose modeling language in the field of software engineering. The Unified Modeling Language includes a set of graphic notation techniques to create visual models of object-oriented software-intensive systems.

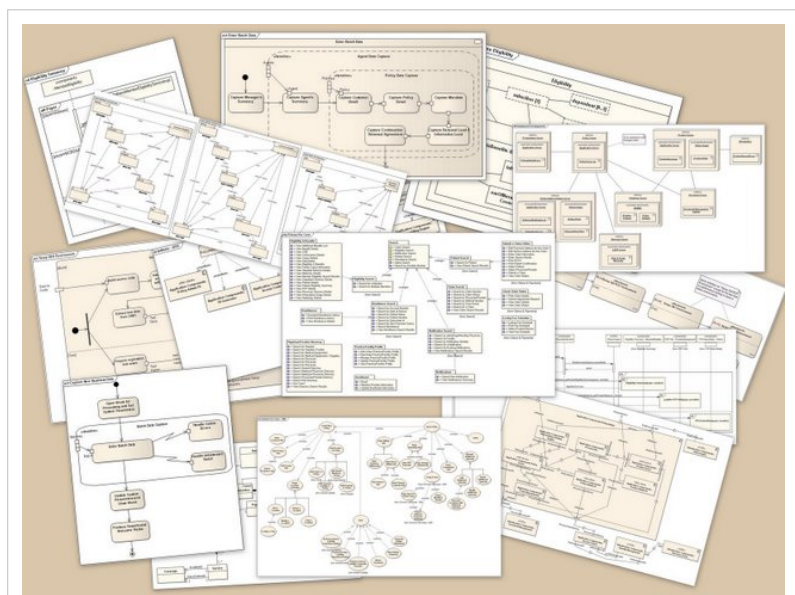
The Unified Modeling Language was developed by Grady Booch, Ivar Jacobson and James Rumbaugh at Rational Software in the 1990s.^[1] It was adopted by the Object Management Group (OMG) in 1997, and has been managed by this organization ever since. In 2000 the Unified Modeling Language was accepted by the International Organization for Standardization (ISO) as industry standard for modeling software-intensive systems. The current version of the UML is 2.4.1 published by the OMG in August 2011.

Overview

Unified Modeling Language (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.^[2]

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.^[3]



A collage of UML diagrams.

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. 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.

History

UML has been evolving since the second half of the 1990s and has its roots in the object-oriented methods developed in the late 1980s and early 1990s. The timeline (see image) shows the highlight of the history of object-oriented modeling methods and notation.

Before UML 1.x

After Rational Software Corporation hired James Rumbaugh from General Electric in 1994, the company became the source for two of the most popular object-oriented modeling approaches of the day.^[4] Rumbaugh's

Object-modeling technique (OMT) and Grady Booch's method known as 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,^[5] was acquired by Rational. The three methodologists were collectively referred to as the *Three Amigos*.

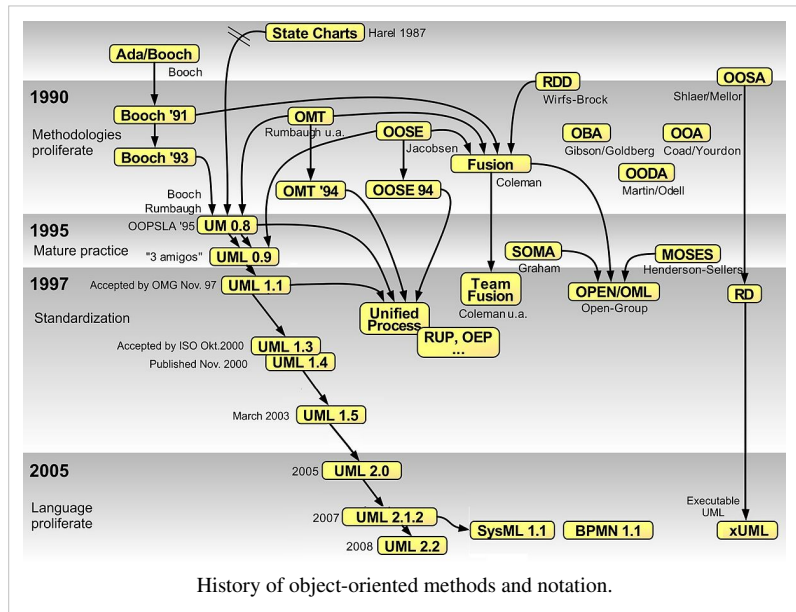
In 1996, Rational concluded that the abundance of modeling languages was slowing the adoption of object technology,^[citation needed] so repositioning the work on a unified method, they tasked the "Three Amigos" with the development of a non-proprietary Unified Modeling Language. Representatives of competing object technology companies were consulted during OOPSLA '96,^[citation needed] they chose *boxes* for representing classes rather than the *cloud* symbols that were used in Booch's notation.

Under the technical leadership of the "Three Amigos", an international consortium called the UML Partners was organized in 1996 to complete the *Unified Modeling Language (UML)* specification, and propose it as a response to the OMG RFP. The UML Partners' UML 1.0 specification draft was proposed to the OMG in January 1997. During the same month the UML Partners formed a Semantics Task Force, chaired by Cris Kobryn and administered by Ed Eykholt, to finalize the semantics of the specification and integrate it with other standardization efforts. The result of this work, UML 1.1, was submitted to the OMG in August 1997 and adopted by the OMG in November 1997.

UML 1.x

As a modeling notation, the influence of the OMT notation dominates (e. g., using rectangles for classes and objects). Though the Booch "cloud" notation was dropped, the Booch capability to specify lower-level design detail was embraced. The use case notation from Objectory and the component notation from Booch were integrated with the rest of the notation, but the semantic integration was relatively weak in UML 1.1, and was not really fixed until the UML 2.0 major revision.^[citation needed]

Concepts from many other OO methods were also loosely integrated with UML with the intent that UML would support all OO methods. Many others also contributed, with their approaches flavouring the many models of the day, including: Tony Wasserman and Peter Pircher with the "Object-Oriented Structured Design (OOSD)" notation (not a method), Ray Buhr's "Systems Design with Ada", Archie Bowen's use case and timing analysis, Paul Ward's data analysis and David Harel's "Statecharts"; as the group tried to ensure broad coverage in the real-time systems domain. As a result, UML is useful in a variety of engineering problems, from single process, single user



applications to concurrent, distributed systems, making UML rich but also large.

The Unified Modeling Language is an international standard:

ISO/IEC 19501:2005 Information technology – Open Distributed Processing – Unified Modeling Language (UML) Version 1.4.2

UML 2.x

UML has matured significantly since UML 1.1. Several minor revisions (UML 1.3, 1.4, and 1.5) fixed shortcomings and bugs with the first version of UML, followed by the UML 2.0 major revision that was adopted by the OMG in 2005.

Although UML 2.1 was never released as a formal specification, versions 2.1.1 and 2.1.2 appeared in 2007, followed by UML 2.2 in February 2009. UML 2.3 was formally released in May 2010. UML 2.4.1 was formally released in August 2011. UML 2.5 was released in October 2012 as an "In process" version and has yet to become formally released.

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

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

The current versions of these standards follow: UML Superstructure version 2.4.1, UML Infrastructure version 2.4.1, OCL version 2.3.1, and UML Diagram Interchange version 1.0.

Although many UML tools support some of the new features of UML 2.x, the OMG provides no test suite to objectively test compliance with its specifications.

Topics

Software development methods

UML is not a development method by itself;^[6] however, it was designed to be compatible with the leading object-oriented software development methods of its time (for example OMT, Booch method, Objectory). Since UML has evolved, some of these methods have been recast to take advantage of the new notations (for example OMT), and new methods have been created based on UML, such as IBM Rational Unified Process (RUP). Others include Abstraction Method and Dynamic Systems Development Method.

Modeling

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).

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

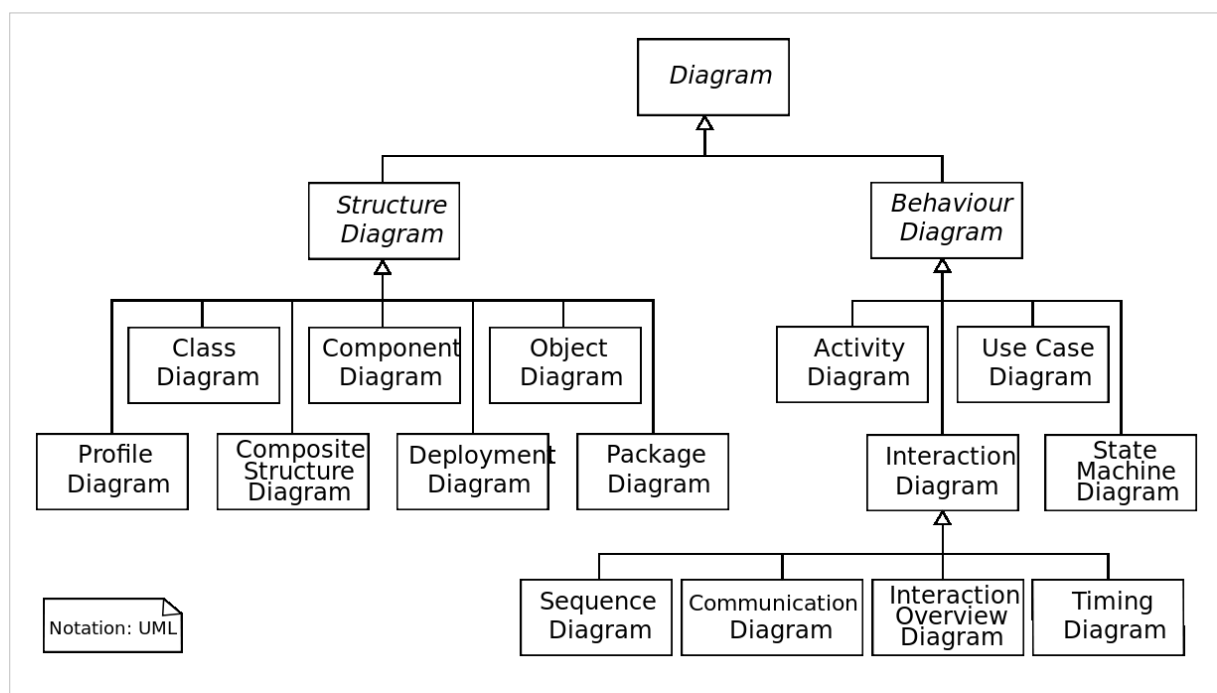
- Static (or *structural*) view: emphasizes the static structure of the system using objects, attributes, operations and relationships. The structural view includes class diagrams and composite structure diagrams.
- Dynamic (or *behavioral*) view: emphasizes the dynamic behavior of the system by showing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

UML models can be exchanged among UML tools by using the XML Metadata Interchange (XMI) interchange format.

Diagrams overview

UML diagrams	
Structural UML diagrams	
<ul style="list-style-type: none"> • Class diagram • Component diagram • Composite structure diagram • Deployment diagram • Object diagram • Package diagram • Profile diagram 	
Behavioral UML diagrams	
<ul style="list-style-type: none"> • Activity diagram • Communication diagram • Interaction overview diagram • Sequence diagram • State diagram • Timing diagram • Use case diagram 	
•	v
•	t
•	e [8]

UML 2.2 has 14 types of diagrams divided into two categories.^[9] 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 class diagram:



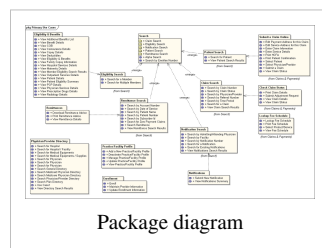
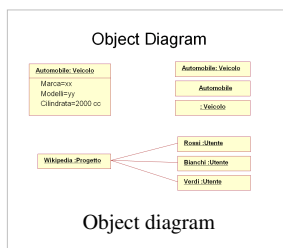
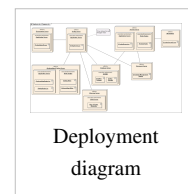
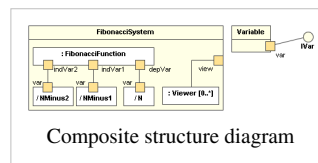
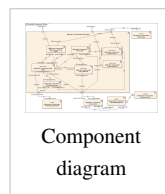
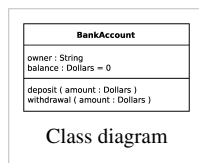
UML does not restrict UML element types to a certain diagram type. In general, every UML element may appear on almost all types of diagrams; this flexibility has been partially restricted in UML 2.0. UML profiles may define additional diagram types or extend existing diagrams with additional notations.

In keeping with the tradition of engineering drawings,^[citation needed] a comment or note explaining usage, constraint, or intent is allowed in a UML diagram.

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.

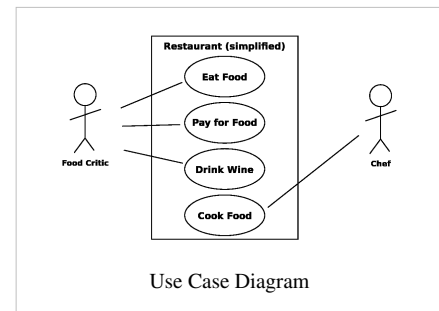
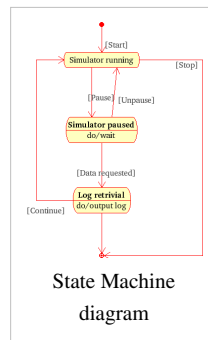
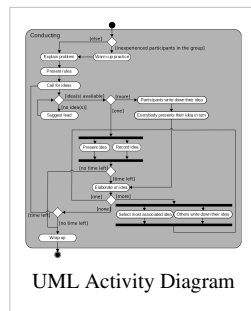
- Class diagram: describes the structure of a system by showing the system's classes, their attributes, and the relationships among the classes.
- Component diagram: describes how a software system is split up into components and shows the dependencies among these components.
- Composite structure diagram: describes the internal structure of a class and the collaborations that this structure makes possible.
- Deployment diagram: describes the hardware used in system implementations and the execution environments and artifacts deployed on the hardware.
- Object diagram: shows a complete or partial view of the structure of an example modeled system at a specific time.
- Package diagram: describes how a system is split up into logical groupings by showing the dependencies among these groupings.
- Profile diagram: operates at the metamodel level to show stereotypes as classes with the <<stereotype>> stereotype, and profiles as packages with the <<profile>> stereotype. The extension relation (solid line with closed, filled arrowhead) indicates what metamodel element a given stereotype is extending.



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.

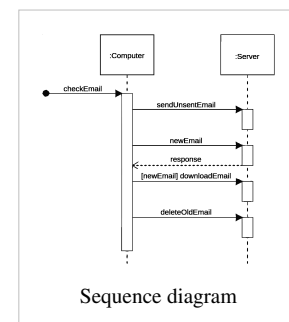
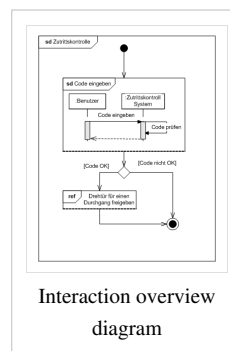
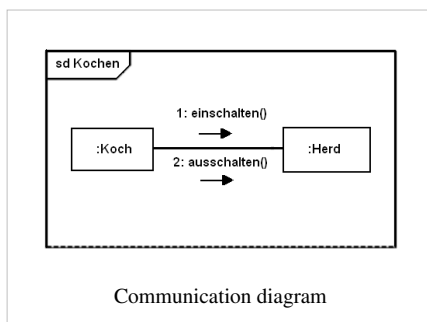
- Activity diagram: describes the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.
- UML state machine diagram: describes the states and state transitions of the system.
- Use Case Diagram: describes the functionality provided by a system in terms of actors, their goals represented as use cases, and any dependencies among those use cases.



Interaction diagrams

Interaction diagrams, a subset of behavior diagrams, emphasize the flow of control and data among the things in the system being modeled:

- Communication diagram: shows the interactions between objects or parts in terms of sequenced messages. They represent a combination of information taken from Class, Sequence, and Use Case Diagrams describing both the static structure and dynamic behavior of a system.
- Interaction overview diagram: provides an overview in which the nodes represent communication diagrams.
- Sequence diagram: shows how objects communicate with each other in terms of a sequence of messages. Also indicates the lifespans of objects relative to those messages.
- Timing diagrams: a specific type of interaction diagram where the focus is on timing constraints.



The Protocol State Machine is a sub-variant of the State Machine. It may be used to model network communication protocols.

The standards have been cited as being ambiguous and inconsistent.^[13] The UML 2.0 standard still suffers many issues.

Capabilities of UML and implementation language mismatch

Typical of other notational systems, UML is able to represent some systems more concisely or efficiently than others. Thus a developer gravitates toward solutions that reside at the intersection of the capabilities of UML and the implementation language. This problem is particularly pronounced if the implementation language does not adhere to orthodox object-oriented doctrine, since the intersection set between UML and implementation language may be that much smaller or equal in size.^[citation needed]

Dysfunctional interchange format

While the XMI (XML Metadata Interchange) standard is designed to facilitate the interchange of UML models, it has been largely ineffective in the practical interchange of UML 2.x models. This interoperability ineffectiveness is attributable to several reasons. Firstly, XMI 2.x is large and complex in its own right, since it purports to address a technical problem more ambitious than exchanging UML 2.x models. In particular, it attempts to provide a mechanism for facilitating the exchange of any arbitrary modeling language defined by the OMG's Meta-Object Facility (MOF). Secondly, the UML 2.x Diagram Interchange specification lacks sufficient detail to facilitate reliable interchange of UML 2.x notations between modeling tools. Since UML is a visual modeling language, this shortcoming is substantial for modelers who don't want to redraw their diagrams. The Diagram Definition OMG project is another alternative.

Cardinality notation

As with database Chen, Bachman, and ISO ER diagrams, class models are specified to use "look-across" cardinalities, even though several authors (Merise,^[14] Elmasri & Navathe^[15] amongst others^[16]) prefer same-side or "look-here" for roles and both minimum and maximum cardinalities. Recent researchers (Feinerer, Dullea et. alia) have shown that the "look-across" technique used by UML and ER diagrams is less effective and less coherent when applied to n-ary relationships of order >2.

In Feinerer it says "Problems arise if we operate under the look-across semantics as used for UML associations. Hartmann investigates this situation and shows how and why different transformations fail." (*Although the "reduction" mentioned is spurious as the two diagrams 3.4 and 3.5 are in fact the same*) and also "As we will see on the next few pages, the look-across interpretation introduces several difficulties which prevent the extension of simple mechanisms from binary to n-ary associations."

Exclusive

The term "unified" applies only to the unification of the many prior existing and competing object-oriented languages. Important well known and popular techniques, almost universally used in industry, such as data flow diagrams and structure charts were not included in the specification.^[citation needed]

Modeling experts have written criticisms of UML, including Brian Henderson-Sellers and Cesar Gonzalez-Perez in "Uses and Abuses of the Stereotype Mechanism in UML 1.x and 2.0".^[17]

Complexity

UML has been criticized for being extremely complex compared to other tools like DOT language.
Wikipedia:Verifiability

References

This article is based on material taken from the Free On-line Dictionary of Computing prior to 1 November 2008 and incorporated under the "relicensing" terms of the GFDL, version 1.3 or later.

- [1] Marc Hamilton (1999) *Software Development: A Guide to Building Reliable Systems* p.48
- [2] Satish Mishra (1997). "Visual Modeling & Unified Modeling Language (UML) : Introduction to UML" (http://www2.informatik.hu-berlin.de/~hs/Lehre/2004-WS_SWQS/20050107_Ex_UML.ppt). Rational Software Corporation. Accessed 9 November 2008.
- [3] Grady Booch, Ivar Jacobson & James Rumbaugh (2000) OMG Unified Modeling Language Specification (<http://www.omg.org/docs/formal/00-03-01.pdf>), Version 1.3 First Edition: March 2000. Retrieved 12 August 2008.
- [4] Andreas Zendler (1997) *Advanced Concepts, Life Cycle Models and Tools for Object-Oriented Software Development*. p.122
- [5] Objectory AB, known as Objectory System, was founded in 1987 by Ivar Jacobson. In 1991, It was acquired and became a subsidiary of Ericsson.
- [6] John Hunt (2000). *The Unified Process for Practitioners: Object-oriented Design, UML and Java*. Springer, 2000. ISBN 1-85233-275-1. p.5.door
- [7] Jon Holt Institution of Electrical Engineers (2004). *UML for Systems Engineering: Watching the Wheels* IET, 2004, ISBN 0-86341-354-4. p.58
- [8] http://en.wikipedia.org/w/index.php?title=Template:UML_diagram_types&action=edit
- [9] *UML Superstructure Specification Version 2.2*. OMG, February 2009.
- [10] Iman Poernomo (2006) " The Meta-Object Facility Typed (<http://calcium.dcs.kcl.ac.uk/1259/1/acm-paper.pdf>)" in: *Proceeding SAC '06 Proceedings of the 2006 ACM symposium on Applied computing*. pp. 1845-1849
- [11] "Ivar Jacobson on UML, MDA, and the future of methodologies" (http://www.infoq.com/interviews/Ivar_Jacobson) (video of interview, transcript available), 24 October 2006. Retrieved 2009-05-22
- [12] See the ACM article "*Death by UML Fever*" (<http://www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=130>) for an amusing account of such issues.
- [13] Génova et alia 2004 "Open Issues in Industrial Use Case Modeling"
- [14] Hubert Tardieu, Arnold Rochfeld and René Colletti La methode MERISE: Principes et outils (Paperback - 1983)
- [15] Elmasri, Ramez, B. Shamkant, Navathe, Fundamentals of Database Systems, third ed., Addison-Wesley, Menlo Park, CA, USA, 2000.
- [16] ER 2004 : 23rd International Conference on Conceptual Modeling, Shanghai, China, 8-12 November 2004 (http://books.google.com/books?id=odZK99osYIEC&pg=PA52&img=1&pgis=1&dq=genova&sig=ACfU3U3tDC_q8WOMqUJW4EZCa5YQywoYLw&edge=0)
- [17] B. Henderson-Sellers; C. Gonzalez-Perez (2006). "Uses and Abuses of the Stereotype Mechanism in UML 1.x and 2.0". in: *Model Driven Engineering Languages and Systems*. Springer Berlin / Heidelberg.

Further reading

- Ambler, Scott William (2004). *The Object Primer: Agile Model Driven Development with UML 2* (<http://www.ambysoft.com/books/theObjectPrimer.html>). Cambridge University Press. ISBN 0-521-54018-6.
- Chonoles, Michael Jesse; James A. Schardt (2003). *UML 2 for Dummies*. Wiley Publishing. ISBN 0-7645-2614-6.
- Fowler, Martin. *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3rd ed.). Addison-Wesley. ISBN 0-321-19368-7.
- Jacobson, Ivar; Grady Booch; James Rumbaugh (1998). *The Unified Software Development Process*. Addison Wesley Longman. ISBN 0-201-57169-2.
- Martin, Robert Cecil (2003). *UML for Java Programmers*. Prentice Hall. ISBN 0-13-142848-9.
- Noran, Ovidiu S. "Business Modelling: UML vs. IDEF" (<http://www.cit.gu.edu.au/~noran/Docs/UMLvsIDEF.pdf>) (PDF). Retrieved 2005-12-28.
- Penker, Magnus; Hans-Erik Eriksson (2000). *Business Modeling with UML*. John Wiley & Sons. ISBN 0-471-29551-5.

External links

- UML Resource Page (<http://www.uml.org/>) of the Object Management Group – Resources that include the latest version of the UML specification from the group in charge of defining the UML specification
 - Death by UML Fever (<http://queue.acm.org/detail.cfm?id=984495>) – An ACM queue article about the abuse of UML
 - Understanding the Unified Modeling Language (UML) (<http://www.methodsandtools.com/archive/archive.php?id=76>) – Introductory article for UML. Retrieved 2011-01-29
 - UML 2.5 resource site (<http://www.uml-diagrams.org/>) - Includes latest notation documentation.
 - ITU-T standard UML profile (<http://www.itu.int/rec/T-REC-Z.109/en>)
-

Article Sources and Contributors

Unified Modeling Language *Source:* <http://en.wikipedia.org/w/index.php?oldid=584848419> *Contributors:* 0, 212.33.53.xxx, 217.162.105.xxx, A. Parrot, AGK, Abelson, Abune, Acather96, Acockrum, AdrianLozano, AdultSwim, AfIn, Afxid, Agentv, Aholub, Alansohn, Alexey.kudinkin, Alexf, Alksentrs, Allan McInnes, Allens, Aludstartups, Ancheta Wis, Anders Nilsson, Andre Engels, Andres Agudelo, Andrewman327, Angela, Anish7, Antandrus, Antonielly, Aquatics, Arjun024, Armand3496, Arnabbbh, Astatine211, BarretB, BarryNorton, Bebates, Bechemel, Beetstra, Beland, Ben modeler, BenAveling, BenFrantzDale, Beorhast, Bergsten, Bernard.szlachta, Bevo, Bhanks, Birge, Black Falcon, BoP, Bob22232, Bobdc, Booyabazooka, Borgx, Boson, Brick Thrower, BrownHornet, Bryan Derksen, Cactus.man, Caesura, Camw, Can't sleep, clown will eat me, CardinalDan, CarrieVS, Cathness, Cesingle07, Cdiggins, Chairboy, ChrisG, ChristianPetro, Clarepawling, Classicaecon, Closedmouth, Cloud ponderer, Cmdrjameson, Cncmaster, Comatose51, Cometstyles, Conti, Corb3126, Corrie.engelbrecht, Craigy144, Crazyinas, CrniBombarder!!!, Crysb, Cyberhitesh, Danakil, Danim, Daveryan, David KELLER, David Martland, DavidSaff, Dayewalker, Dbarashev, De-Wiki-observer, DePiep, DeQuire, Deeray82, Demian12358, Demiurg, DennisDaniels, Deon Steyn, DerHexer, Derek Ross, Dethomas, Dhbernstein, Dirk Riehle, Dirkbike, Discospinster, Diyoey, Dmccreary, Dobbin2000, Doradus, Doral, Duncharris, Dvdtrgn, EPHERE, Eastlaw, Edrawing, Educres, Edward, ETyrant, Emakarov, Emerks, Emmjoyit, EngineerScotty, Enviroboy, Erkan Yilmaz, EuroCarGT, Everykyng, Ewebxml, Falcon8765, Famarsha, Fatespeaks, Fdp, Femto, Fieldday-sunday, Figureskatingfan, Firien, FishSpeaker, Fl, Folajimi, Fotomut, FranckBarbier, Frentos, GLari, Gardar Rurak, GeneralCheese, George100, GeorgeBills, Gerald Mylog, Glenford, Glrx, Gorpik, Gparent, Graphity, Gregbard, Hadal, HamburgerRadio, Hannes Hirzel, Happysailor, Harriv, Harshadsamant, Hedleya, Herbee, Hervegirod, Hhielscher, Hhtracy, Hohohoneysingh, Hongooi, Hpetya, Hu12, Hydrogen Iodide, IMSoP, ITocapa, Intelliw, Iridescent, Isarra (HG), Iypen, Ixf64, J Di, JDP90, JLaTondre, JWenting, Ja 62, Jakohn, Janestochter, Jancewic, Jay, Jearroll, Jdmarshall, Jeffq, Jems421, JensMDD, JeremySmyth, Jerome Charles Potts, Jim1138, Jjalexand, Jmcrispin, Jconnor, JohnnyGeetar, Jojalozzo, Jonik, Jonkerz, JorgeGG, Jorgjimen, Josef Šabl cz, Jyouus!, Jpbowen, Jpiw, Jridell, Jrschnur, Julesd, Julthep, JustAnotherJoe, Kadamwiki, Kakurady, Karategeek6, Kataniza, Kenneth Vergil, Khalid hassani, Khishig.s, Kieran0, Kilva, Kimrew, Kingpin13, Kire87, Kishorekumar 62, Kmccoy, Koavf, KrakatoaKatie, Krishvanth, Kronn8, Ksumedh, Kuratowski's Ghost, KyleSummers, Kyng, Lafeber, Lashiec, Learn4life, Lecafe1, LeeHunter, Leggattst, Leibniz, Leszek Jafczuk, Levin, LiDaobing, Ligulem, Lloydsmart, Logonos, Loqi, Luckyherb, Lwoodyiii, M4gnum0n, MDE, MME, MVerkerk, Malleus Fathuorum, Manu31415, Mardus, Maria C Mosak, Mark Foskey, Mark Renier, Mark.murphy, Marsve, MartinSpamer, Martinig, Martpol, Materialsscientist, MaxHund, Maximumbrownidge, Mayumashu, Mdd, MegoMars, Mgillett, MicahDCochran, Michael Hardy, MichaelBillington, Mike Rosoft, Mikhail Ryazanov, Mintleaf, Mirosamek, Misteror, Mjchonoles, Mo0, Monymirza, Morgankevinj huggle, Mortense, MrOllie, Mrgamedesign, Mrockman, Msa, Mxn, Myguest, Myselffram, Mysidia, Naayagan, Nagarjunag, Narcissus, Nasa-verve, Nashrul Hakiem, Nealmbc, Niceguyede, Nick1nildram, Nikai, Nioski, Nixeagle, Noah Salzman, O watkins, OVERMIND, Obradovic Goran, Octahedron80, Ohnoitsjamie, Oicunayberight, Oleg Alexandrov, Oli Filth, Opptic, Ottomachin, Ozten, Pengo, Peterkimrowe, Petrb, Pgerts, Philip Trueman, PhilippWeissenbacher, Philippe, Piano non troppo, Pierre5018, Piksou, Pindakaas, Pmerson, Poccil, Pointillist, Pojo, Polluks, Poor Yorick, Pp3223, Ppelleti, PrestonH, Provalente, Progdev, Project2501a, QVT, RJFJR, Rajesh.jadhav.18, RayGates, Rdeboer, Reaper Eternal, Red-eyed demon, Remi0o, Rich Farmbrough, Richard-VS-545, Riverdusty, Rob Burbidge, Robertbowerman, Rodasmith, Rodii, Ronark, Ronz, Roux, Royboycrashfan, Rspeer, Rsrikanth05, Ryadav, Ryan shell, S ried, S.K., SAE1962, Sae1962, Sahuagin, Salix alba, Salty-horse, Salvat, Samliew, Sannse, Sarav62, SarekOfVulcan, Sbwoodside, Scieurine, Scott Martin, ScottWambler, Seanhan, Seattlenow, SebastianBreier, Sega381, Senak, Serkanbirkun, Shadowjams, Shanes, Siyu LI, Sketch-The-Fox, Slashme, Sleepyhead81, Smsarmad, Snoyes, Softzen, Spacelib, Spasemunki, Sprachpfeleg, SpyMagician, Steven Kelly, Stiang, StuffOfInterest, Suffusion of Yellow, Sven Manguard, TastyPoutine, Taw, Tawaregs08.it, Tayal01, Techman224, Technopilgrim, TeleOffice, Tellyaddict, TerraFrost, Tevirselahe, That Guy, From That Show!, TheAMMollusc, Thejavaguy, Thowa, Thumperward, TurnerRupert, Timo Honkasalo, Timothyclethbridge, Tobias Hoevekamp, Toddfast, Tom Jenkins, TommyG, TowerDragon, Treisijs, Triwbe, Twasserman, Um9penmc, VOGELLA, Vanished user g454XxNpUVWvxzlr, Vardhanw, Vespristiano, Victor Snezhko, Vim-Hogar, Vocaro, Vrsane, Wael Ellithy, Walter Görlitz, WalterGR, Wht p0wer, Wikipelli, Willking1979, Windknigh, Wiretse, Wmahan, Woohookitty, Xactium, Xihir, Xmen253, Xtifr, Yakushima, Yamamoto Ichiro, Yamla, Yellowdesk, Zanemoody, Zappernapper, Zergwyn, Zoicon5, Zolkash, Zuckon, Šedý, 甲 虫, 1118 anonymous edits

Image Sources, Licenses and Contributors

File:UML Diagrams.jpg *Source:* http://en.wikipedia.org/w/index.php?title=File:UML_Diagrams.jpg *License:* Creative Commons Attribution-ShareAlike 3.0 Unported *Contributors:* Kishorekumar 62

File:OO Modeling languages history.jpg *Source:* http://en.wikipedia.org/w/index.php?title=File:OO_Modeling_languages_history.jpg *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* User:Mdd

File:UML diagrams overview.svg *Source:* http://en.wikipedia.org/w/index.php?title=File:UML_diagrams_overview.svg *License:* Public Domain *Contributors:* Derfel73; Pmerson

File:BankAccount1.svg *Source:* <http://en.wikipedia.org/w/index.php?title=File:BankAccount1.svg> *License:* GNU Free Documentation License *Contributors:* Donald Bell

File:Policy Admin Component Diagram.PNG *Source:* http://en.wikipedia.org/w/index.php?title=File:Policy_Admin_Component_Diagram.PNG *License:* GNU Free Documentation License *Contributors:* Kishorekumar 62

File:Composite Structure Diagram.png *Source:* http://en.wikipedia.org/w/index.php?title=File:Composite_Structure_Diagram.png *License:* Public Domain *Contributors:* [w:User:KenSWebbUser:KenSWebb]

File:Deployment Diagram.PNG *Source:* http://en.wikipedia.org/w/index.php?title=File:Deployment_Diagram.PNG *License:* GNU Free Documentation License *Contributors:* Kishorekumar 62

File:Object diagram.png *Source:* http://en.wikipedia.org/w/index.php?title=File:Object_diagram.png *License:* Public Domain *Contributors:* Mac9

File:Package Diagram.PNG *Source:* http://en.wikipedia.org/w/index.php?title=File:Package_Diagram.PNG *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* Kishorekumar 62

File:Activity conducting.svg *Source:* http://en.wikipedia.org/w/index.php?title=File:Activity_conducting.svg *License:* GNU Free Documentation License *Contributors:* Gwaur

Image:UML state diagram.png *Source:* http://en.wikipedia.org/w/index.php?title=File:UML_state_diagram.png *License:* GNU Free Documentation License *Contributors:* Anakin101, Flappiefh, LeonardoG, Mdd, Pcdude2143, Quibik, Sbwoodside, 3 anonymous edits

Image:UML_Use_Case_diagram.svg *Source:* http://en.wikipedia.org/w/index.php?title=File:UML_Use_Case_diagram.svg *License:* Creative Commons Attribution-Share Alike *Contributors:* Slashme

Image:Kommunikations diagramm-2.png *Source:* http://en.wikipedia.org/w/index.php?title=File:Kommunikations_diagramm-2.png *License:* GNU Free Documentation License *Contributors:* Gubaer

Image:lau-diagramm-1.png *Source:* <http://en.wikipedia.org/w/index.php?title=File:lau-diagramm-1.png> *License:* GNU Free Documentation License *Contributors:* Original uploader was Gubaer at de.wikipedia

Image:CheckEmail.svg *Source:* <http://en.wikipedia.org/w/index.php?title=File:CheckEmail.svg> *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* Sae1962, Skim, Zerodamage

Image:M0-m3.png *Source:* <http://en.wikipedia.org/w/index.php?title=File:M0-m3.png> *License:* Public Domain *Contributors:* Gubaer

License

Creative Commons Attribution-Share Alike 3.0
//creativecommons.org/licenses/by-sa/3.0/