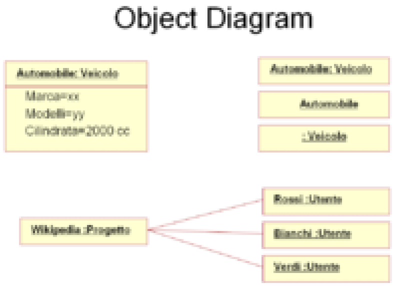
WIKIPEDIA

Object diagram

An **object diagram** in the Unified Modeling Language (UML), is a diagram that shows a complete or partial view of the structure of a modeled system at a specific time.



Example of an Object diagram.

Contents

Overview

In the Unified Modeling Language (UML), an object diagram focuses on some particular set of objects and attributes, and the links between these instances. A correlated set of object diagrams provides insight into how an arbitrary view of a system is expected to evolve over time. In early UML specifications the object diagram is described as:

> "*An object diagram is a graph of instances, including objects and data values. A static object diagram is an instance of a class diagram; it shows a snapshot of the detailed state of a system at a point in time. The use of object diagrams is fairly limited, namely to show examples of data structure*."[1][2]

The latest UML 2.5 specification does not explicitly define object diagrams,[3] but provides a notation for instances of classifiers.[4]

Object diagrams and class diagrams are closely related[5] and use almost identical notation.[6] Both diagrams are meant to visualize static structure of a system. While class diagrams show classes, object diagrams display instances of classes (objects).[7] Object diagrams are more concrete than class diagrams. They are often used to provide examples or act as test cases for class diagrams. Only aspects of current interest in a model are typically shown on an object diagram.

Object diagram topics

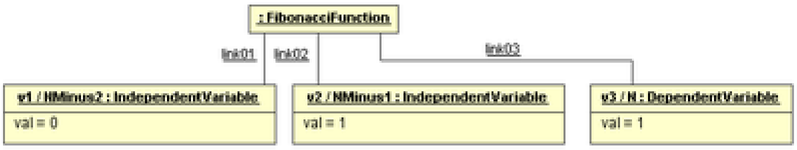
Instance specifications

Each object and link on an object diagram is represented by an *InstanceSpecification*. This can show an object's classifier (e.g. an abstract or concrete class) and instance name, as well as attributes and other structural features using *slots*. Each *slot* corresponds to a single attribute or feature, and may include a value for that entity

The name on an instance specification optionally shows an instance name, a ':' separator, and optionally one or more classifier names separated by commas. The contents of slots, if any, are included below the names, in a separate attribute compartment. A link is shown as a solid line, and represents an instance of an association.

Object diagram example

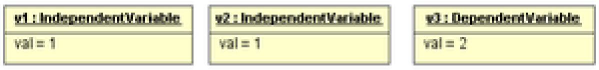
Consider one possible way of modeling production of the Fibonacci sequence.


: FibonacciFunction
link01
link02
link03
v1 / NMinus2 : IndependentVariable
val = 0
v2 / NMinus1 : IndependentVariable
val = 1
v3 / N : DependentVariable
val = 1

Initially, when n=2, and f(n-2) = 0, and f(n-1) =1, then f(n) = 0 + 1 = 1.

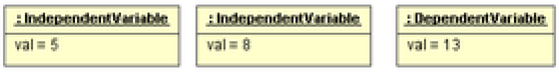
In the first UML object diagram on the right, the instance in the leftmost instance specification is named *v1*, has *IndependentVariable* as its classifier, plays the *NMinus2* role within the *FibonacciSystem*, and has a slot for the *val* attribute with a value of *0*. The second object is named *v2*, is of class *IndependentVariable*, plays the *NMinus1* role, and has *val = 1*. The *DependentVariable* object is named *v3*, and plays the *N* role. The topmost instance, an anonymous instance specification, has *FibonacciFunction* as its classifier, and may have an instance name, a role, and slots, but these are not shown here. The diagram also includes three named links, shown as lines. Links are instances of an association.

In the second diagram, at a slightly later point in time, the *IndependentVariable* and *DependentVariable* objects are the same, but the slots for the *val* attribute have different values. The role names are not shown here.



After the first iteration, when n = 3, and f(n-2) = 1, and f(n-1) = 1, then f(n) = 1 + 1 = 2.

In the last object diagram, a still later snapshot, the same three objects are involved. Their slots have different values. The instance and role names are not shown here.



After several more iterations, when n = 7, and f(n-2) = 5, and f(n-1) = 8, then f(n) = 5 + 8 = 13.

Usage

If you are using a UML modeling tool, you will typically draw object diagrams using some other diagram type, such as on a class diagram. An object instance may be called an *instance specification* or just an *instance*. A link between instances is generally referred to as a *link*. Other UML entities, such as an aggregation or composition symbol (a diamond) may also appear on an object diagram.

References

1. Object Management Group (2001) UML specification 1.4 (http://www.omg.org/spec/UML/1.4), September 2001
2. Anne Banks Pidduck, John Mylopoulos, Carson C. Woo (2002) *Advanced Information Systems Engineering*. p.776.
3. Classification of UML 2.5 Diagrams (http://www.uml-diagrams.org/uml-25-diagrams.html) on uml-diagrams.org. Retrieved Dec 7, 2012
4. Object Management Group (2015) UML specification 2.5 (http://www.omg.org/spec/UML/2.5), Section 9.8.4 on notation of InstanceSpecification, March 2015
5. Marcus Fontoura, Wolfgang Pree & Bernhard Rumpe (2002) *The UML profile for framework architectures*. p.19
6. Kassem A. Saleh (2009) *Software Engineering*. p.47
7. Bianca Scholten (2007) *The Road to Integration: A Guide to Applying the ISA-95 Standard in Manufacturing*. p.155

External links

- General information on UML including the official UML 2.0 specification. See especially the section on InstanceSpecification.
- The Expressive Power of Object Diagrams, an illustration from the real world
- Modal object diagrams, an extension of object diagrams with existential/universal and positive/negative modes.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Object_diagram&oldid=855126778"