

¿Que es UML?

UML (*Unified Modeling Language*) es un lenguaje para especificar, visualizar, construir y documentar los elementos de un sistema software, así como para modelado de procesos de negocio u otros sistemas no-software. UML reúne una colección de las mejores prácticas en la ingeniería que han sido utilizadas con éxito para modelar sistemas grandes y complejos.

¿Quien ha creado UML?

UML ha sido creado por los expertos en metodología Grady Booch, Ivar Jacobson, y Jim Rumbaugh en Rational Software, utilizando información de otros importantes expertos en metodología, vendedores de software, y usuarios finales. Su objetivo era unificar los diversos sistemas que había y crear un lenguaje de modelado con las mejores características de cada uno.

¿Es UML un estandar?

Si, UML fue adoptado por el OMG (*Object Management Group*) como estandar en noviembre de 1997.

1. What is UML?

The Unified Modeling Language (UML) is the industry-standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. Using UML, programmers and application architects can make a blueprint of a project, which, in turn, makes the actual software development process easier.

2. Who created UML?

UML was created at Rational Software by methodologists Grady Booch, Ivar Jacobson, and Jim Rumbaugh with input from other leading methodologists, many software vendors, as well as end-users. Its aim is to unify the various existing systems into a best-of-breed modeling language.

3. Is UML a standard?

Yes. UML was adopted by the Object Management Group (www.omg.org) as a standard in November, 1997.

4. What can I use UML for?

UML was designed with these primary purposes in mind:

- Business process modeling with use cases
- Class and object modeling
- Component modeling
- Distribution and deployment modeling

5. Do I really need UML? Can't I just describe how my application is designed using regular words?

While it's certainly possible to describe interrelated processes and code architecture in words, many people prefer to use a diagram to visualize the relationship of elements to one another. UML is a standard way to create these diagrams. As a result, it makes it easier for programmers and software architects to communicate.

6. I already use project management software to plan my software development projects. Why should I use UML?

Unlike project management software, which is designed to coordinate the various parts of a project in relation

to a timeline, UML-based diagrams can be used to show the interrelationships within a project from an architectural perspective.

7. How does UML relate to OMT, Booch, OOSE, and other modeling languages?

One of the goals of the design of UML was to unify the various modeling languages into one common standard. Those that are familiar with OMT, Booch, or OOSE will have little trouble picking up UML because many of the concepts are very similar.

8. Are there UML-based tools that I can use today?

Yes. Many UML tools are starting to appear. The most popular visual modeling tool is Rational Rose from Rational Software (www.rational.com/products/rose), the originators of UML.

9. What are some of the companies working on UML?

Hewlett-Packard (www.hp.com)
i-Logix (www.ilogix.com)
IBM (www.ibm.com)
ICON Computing (www.iconcomp.com)
IntelliCorp (www.intellicorp.com)
MCI Systemhouse (uml.systemhouse.mci.com)
Microsoft (www.microsoft.com)
ObjecTime (www.objecttime.com)
Oracle (www.oracle.com)
Platinum Technology (www.platinum.com)
Ptech (www.ptechinc.com)
Rational Software (www.rational.com)
Reich Technologies (www.projexion.com)
Sterling Software (www.sterlingsoftware.com)
Taskon (www.sn.no/taskon)
Unisys (www.unisys.com)

Glosario Semántico del UML

Versión 1.0

13 de Enero de 1997

Sobre este Glosario

Este glosario define los términos utilizados para describir el Lenguaje Unificado para Modelado (Unified Modeling Language : UML). Incluye, además de la terminología específica del UML, términos provenientes de los estándares del OMG y de los métodos de análisis y diseño orientado a objetos.

Alcance

Este glosario incluye términos de las siguientes fuentes primarias :

- Semántica del UML
- Guía de Notación del UML

Además han sido utilizadas las siguientes fuentes secundarias :

- Object Management Architecture object model [OMA]

- CORBA 2.0 [CORBA]
- Object Analysis & Design RFP-1 [OA&D RFP]
- Rational Process (en preparación por Philippe Kruchten, et al.) [RATLPROC]

[OMA], [CORBA] y [OA&D RFP] fueron utilizados para promover la compatibilidad con el OMG y proveer términos del área de objetos distribuidos que complementan al UML. (Cuando hay inconsistencias entre las tres fuentes del OMG, las hemos jerarquizado en el orden en que aparecen aquí.)

[RATLPROC] fue utilizado para proveer términos de arquitectura y proceso que complementan al UML.

Convenciones de la Notación

Las entradas al glosario generalmente empiezan en minúsculas. Se ha usado una letra inicial mayúscula cuando esta es la práctica más frecuente para esa palabra. Las siglas están en mayúsculas, a menos que tradicionalmente aparezcan en minúsculas.

Una palabra o frase entre corchetes dentro de un término compuesto indica que la palabra o frase es opcional cuando se hace referencia al término. Por ejemplo : *[clase] caso de uso* puede referirse como *caso de uso*.

- **Contraste:** <término>. Hace referencia a un término que tiene un significado opuesto o sustancialmente diferente.
- **Ver:** <término>. Hace referencia a un término que tiene un significado similar, pero no sinónimo.
- **Sinónimo:** <término>. Indica que ambos términos tienen el mismo significado.
- **Sigla:** <término>. Indica que el término es una sigla. Usualmente se hace referencia a la forma escrita palabra por palabra, a menos que esta sea usada rara vez.

El glosario tiene muchas referencias cruzadas para ayudar a encontrar términos que figuran en varios lugares. Las referencias cruzadas están subrayadas y son vínculos en la versión hipertexto. La versión hipertexto del Glosario del UML aparecerá en los sitios de internet de Rational Software: <http://www.rational.com/uml> y de Baufest: <http://www.baufest.com>.

Convenciones de la Traducción

Las referencias a términos del OMG se dejaron en inglés.

Acompaña, a la frase en Español correspondiente a cada entrada del glosario, la frase original en Inglés entre paréntesis.

Contenido

Las entradas del glosario están organizadas alfabéticamente. Para acceder a ellas seleccione la primer letra en la lista de abajo. Si está buscando una palabra que empiece con un dígito o con un símbolo, elija el vínculo '#'.

A B C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z #

-A-

abstracción (abstraction)

Las características esenciales que distinguen a una entidad de todas las demás entidades. Una abstracción define una delimitación relativa a la perspectiva del observador.

acción (action)

Un procedimiento, algorítmico o computacional.

activación (activation)

La ejecución de una acción. Contraste: activation [OMA].

actor (actor [class])

Un tipo con un estereotipo predefinido, que denota una entidad externa al sistema que interactúa con casos de uso.

agregación (aggregation)

Una forma especial de asociación que especifica una relación todo – parte entre el agregado (todo) y una parte que lo compone. Contraste: composición.

agregación de composición (composite aggregation)

Sinónimo: composición.

agregado (aggregate [class])

Una clase que representa el "todo" en una relación de agregación (todo – parte). Ver: agregación.

análisis (analysis)

La parte del proceso de desarrollo de software cuyo propósito principal es realizar un modelo del dominio del problema. El análisis hace foco en qué hacer, el diseño hace foco en cómo hacerlo. Contraste: diseño.

argumento (argument)

Un valor específico correspondiente a un parámetro. Sinónimo: parámetro real. Contraste: parámetro.

arquitectura (architecture)

La estructura organizacional de un sistema. Una arquitectura puede ser descompuesta recursivamente en : partes que interactúan entre sí por medio de interfaces, relaciones que conectan las partes, y restricciones para ensamblar las partes.

artefacto (artifact)

Una información que es usada o producida mediante un proceso de desarrollo de software. Un artefacto puede ser un modelo, una descripción o software.

asociación (association)

Una relación que describe un conjunto de vínculos.

asociación binaria (binary association)

Una asociación entre dos clases. Un caso especial de asociación n-aria.

asociación n-aria (n-ary association)

Una asociación entre tres o más clases. Cada instancia de la asociación es una n-tupla de valores de las clases respectivas. Contraste: asociación binaria.

atributo (attribute)

Una propiedad de un tipo, identificada mediante un nombre. Sinónimo: attribute [OMA].

Indice

–B–

booleano (boolean)

Una enumeración cuyos valores son verdadero y falso.

Indice

–C–

capa (layer)

Una forma específica de agrupar paquetes en un modelo al mismo nivel de abstracción.

cardinalidad (cardinality)

La cantidad de elementos de un conjunto. Contraste: multiplicidad.

carril (swimlane)

Un paquete dentro de un diagrama de interacciones que identifica a los responsables de realizar un grupo de acciones. A menudo corresponden a unidades organizacionales en un modelo de negocios.

caso de uso (use case [class])

Una clase que define un conjunto de instancias de caso de uso.

clase (class)

La descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones, y semántica. Una clase es una implementación de un tipo. Sinónimo: class [OMG], Ver: tipo, implementación.

clase abstracta (abstract class)

Una clase que no puede ser instanciada directamente. Contraste: clase concreta.

clase activa (active class)

Una clase cuyas instancias son objetos activos. Ver: objeto activo.

clase asociación (association class)

Un elemento de modelado que tiene propiedades tanto de asociación y como de clase. Una clase asociación puede ser vista tanto como una asociación que también tiene propiedades de clase, o como una clase que también tiene propiedades de asociación.

clase concreta (concrete class)

Una clase que puede ser instanciada directamente. Contraste: clase abstracta.

clase paramétrica (parametrized class)

La descripción de una clase con uno o más parámetros no ligados. Sinónimo: molde.

clasificación dinámica (dynamic classification)

Una variación semántica de la generalización en la que un objeto puede cambiar de tipo o rol. Contraste: clasificación estática.

clasificación estática (static classification)

Una variación semántica de la generalización en la que un objeto no puede cambiar de tipo, o de rol. Contraste: clasificación dinámica.

clasificación múltiple (multiple classification)

Una variación semántica de la generalización en la cual un objeto puede pertenecer a más de una clase en forma directa. Ver : clasificación dinámica.

cliente (client)

Un tipo, clase o componente que solicita un servicio de otro tipo, clase o componente. Sinónimo: client object [OMA]. Contraste: proveedor.

colaboración (collaboration)

Un contexto que sustenta un conjunto de interacciones. Ver: interacción.

componente (component)

Un módulo de software ejecutable que tiene identidad y una interface bien definida. Contraste: component [OMA].

comportamiento (behavior)

Los efectos visibles de una operación o evento, incluyendo sus resultados. Sinónimo: behavior [OMA].

composición (composition)

Una forma de agregación más fuerte, con una mayor posesión y coincidencia de ciclo de vida entre el todo y

las partes. Las partes que no tienen multiplicidad fija pueden ser creadas después del compuesto en sí, pero una vez creadas viven y mueren con él (i.e., comparten su ciclo de vida). Estas partes también pueden ser removidas antes de la muerte del compuesto. La composición puede ser recursiva. Sinónimo: agregación de composición.

compuesto (composite [class])

Una clase que está relacionada con una o más clases por medio de una relación de composición. Ver: composición.

comunicación [asociación de comunicación] (communication association)

En un diagrama de plataforma una asociación entre nodos que implica una comunicación. Ver: diagrama de plataforma.

concurrency (concurrency)

La ocurrencia de dos o más actividades durante el mismo intervalo de tiempo. La concurrencia puede ser lograda ejecutando simultáneamente o en forma intercalada, dos o más hilos de ejecución. Ver: hilo de ejecución.

condición de protección (guard condition)

Una condición que debe ser satisfecha para que una transición asociada a ella se dispare.

contenedor (container)

1. Un objeto que existe para contener otros objetos, y que provee operaciones para acceder o iterar sobre su contenido. Por ejemplo : arreglos, conjuntos, diccionarios.
2. Una componente que existe para contener otras componentes.

contexto (context)

Una vista de un conjunto de elementos del modelo relacionados para un propósito particular, como especificar una operación.

cualificador (qualifier)

Un atributo, o tupla de atributos, cuyos valores particionan el conjunto de objetos que participan en una asociación en un solo objeto a través de la asociación.

Indice

–D–

delegación (delegation)

La habilidad de un objeto para enviar un mensaje a otro objeto en respuesta a un mensaje recibido. La delegación puede ser usada como alternativa a la herencia. Sinónimo: delegation [OMA]. Contraste: herencia.

dependencia (dependency)

Una relación entre dos elementos del modelo, en la cual, si se efectúa un cambio a uno de los elementos (el elemento independiente), posiblemente se afecte al otro elemento (el elemento dependiente).

diagrama (diagram)

La presentación gráfica de una colección de elementos del modelo, frecuentemente plasmada como un grafo conectado de arcos (relaciones) y vértices (otros elementos del modelo). Los siguientes diagramas son parte del UML : diagrama de clases, diagrama de objetos, diagrama de casos de uso, diagrama de secuencias, diagrama de colaboraciones, diagrama de estados, diagrama de actividades, diagrama de componentes, y diagrama de plataforma.

diagrama de actividades (activity diagram)

Un caso especial de diagrama de estados en el que todos, o la mayoría, son estados activos y en el que todas, o la mayoría, de las transiciones son disparadas por la finalización de las acciones de los estados. Contraste: diagrama de estados.

diagrama de casos de uso (use case diagram)

Un diagrama que muestra la relación entre actores y casos de uso dentro de un sistema.

diagrama de clases (class diagram)

Un diagrama que muestra una colección de elementos del modelo declarativos (estáticos), como clases y tipos, y sus contenidos y relaciones.

diagrama de colaboraciones (collaboration diagram)

Un diagrama que muestra interacciones entre objetos organizadas alrededor de los objetos y sus vinculaciones. A diferencia de un diagrama de secuencias, un diagrama de colaboraciones muestra las relaciones entre los objetos. Los diagramas de secuencias y los diagramas de colaboraciones expresan información similar, pero en una forma diferente. Ver: diagrama de secuencias.

diagrama de componentes (component diagram)

Un diagrama que muestra la organización de las componentes y sus dependencias.

diagrama de estados (state diagram)

Un diagrama que muestra una máquina de estados. Ver: máquina de estados.

diagrama de interacciones (interaction diagram)

Un término genérico que se aplica a diversos tipos de diagramas que enfatizan la interacción entre objetos. Incluye : diagrama de colaboraciones, diagrama de secuencias y diagrama de actividades. Ver: diagrama de colaboraciones, diagrama de secuencias, diagrama de actividades.

diagrama de objetos (object diagram)

Un diagrama que contiene objetos y sus relaciones en un momento dado del tiempo. Un diagrama de objetos puede ser considerado un caso especial de un diagrama de clases o de un diagrama de colaboraciones. Ver: diagrama de clases, diagrama de colaboraciones.

diagrama de plataforma (deployment diagram)

Un diagrama que muestra la configuración de los nodos de procesamiento en tiempo de ejecución; y las componentes, procesos y objetos que viven en ellos. Las componentes representan las manifestaciones en tiempo de ejecución de las unidades de código. Ver: diagrama de componentes.

diagrama de secuencias (sequence diagram)

Un diagrama que muestra interacciones entre objetos organizadas en secuencia temporal. En particular muestra los objetos participantes de la interacción y la secuencia de mensajes intercambiados. A diferencia del diagrama de colaboraciones, un diagrama de secuencias incluye la secuencia temporal pero no incluye las relaciones entre los objetos. Pueden existir diagramas de secuencias genéricos (que describen todos los escenarios posibles) y de instancias (que describen un escenario particular). Los diagramas de secuencias y de colaboraciones expresan información similar pero de una manera diferente. Ver: diagrama de colaboraciones.

diseño (design)

La parte del proceso de desarrollo de software cuyo propósito principal es decidir cómo se construirá el sistema. Durante el diseño se toman decisiones estratégicas y tácticas para alcanzar los requerimientos funcionales y la calidad esperada.

disparar (fire)

Causar una transición de estados. Ver: transición.

dominio (domain)

Un área de conocimiento o actividad caracterizada por un conjunto de conceptos y términos comprendidos por los practicantes de esa área.

Indice

–E–

elemento (element)

Un elemento atómico constitutivo de un modelo.

elemento de una vista (view element)

Una proyección textual y/o gráfica de una colección de elementos del modelo.

elemento del modelo (modeling element)

Un elemento que es una abstracción destacada del sistema que está siendo modelado.

elemento derivado (derived element)

Un elemento del modelo que puede ser computado a partir de otros elementos del modelo, pero que es incluido por claridad o por una cuestión de diseño, por más que no agrega información semántica.

elemento generalizable (generalizable element)

Un elemento del modelo que puede participar en una relación de generalización. Ver: generalización.

emisor (sender [object])

El objeto que le pasa un mensaje a un receptor. Contraste: receptor.

enumeración (enumeration)

Una lista de valores, identificados por medio de nombres, utilizados como rango de valores para un tipo de atributo. Por ejemplo : Color = {Rojo, Verde, Azul}.

enviar [un mensaje] (send [a message])

Pasar un mensaje desde un objeto emisor hasta un objeto receptor. Ver: emisor, receptor.

escenario (scenario)

Una secuencia específica de acciones que ilustran un comportamiento. Un escenario puede ser usado para ilustrar una interacción. Ver: interacción.

espacio de nombres (namespace)

La parte de un modelo donde pueden ser definidos y utilizados un conjunto de nombres. Dentro de un espacio de nombres cada nombre tiene un significado único. Ver: nombre.

especificación (specification)

Una descripción declarativa de lo que algo es o hace. Contraste: implementación.

esquema (framework)

Una micro-arquitectura que provee un molde extensible para aplicaciones de un dominio específico.

estado (state)

Una condición o situación en la vida de un objeto, durante la cual satisface una condición, realiza una actividad o está esperando un evento. Contraste: state [OMA].

estado activo (action state)

Un estado con una acción interna y una o más transiciones asociadas a la finalización de la acción interna.

estado compuesto (composite state)

Un estado compuesto por subestados. Contraste: subestado.

estereotipo (stereotype)

Un nuevo tipo de elemento de modelado, que extiende la semántica del metamodelo. Los estereotipos deben basarse en ciertos tipos o clases existentes en el metamodelo. Pueden extender la semántica, pero no la estructura de los tipos o clases preexistentes. Algunos estereotipos están predefinidos en el UML, otros pueden ser definidos por el usuario. Los estereotipos son uno de los mecanismos de extensión del UML.

Contraste: restricción, valor rotulado.

evento (event)

Un acontecimiento significativo. Un evento tiene una ubicación en el tiempo y en el espacio y puede tener parámetros. En el contexto de un diagrama de estado, un evento es un acontecimiento que puede disparar una transición de estados.

evento temporal (time event)

Un evento que ocurre en un tiempo particular. Puede ser especificado por medio de una expresión temporal.
Ver: evento.

exportar (export)

En el contexto de un paquete, hacer que un elemento sea visible fuera del espacio de nombres que lo contiene.
Ver: visibilidad. Contraste: export [OMA], importa.

expresión (expression)

Un literal que evalúa en un valor de un tipo en particular. Por ejemplo : la expresión "(7 + 5 * 3)" evalúa en un valor de tipo número.

expresión booleana (boolean expression)

Una expresión que resuelve en un valor booleano.

expresión de acción (action expression)

Una expresión que resuelve en una colección de acciones.

expresión de tipo (type expression)

Una expresión que evalúa en una referencia a uno o más tipos.

expresión temporal (time expression)

Una expresión que resuelve a un valor temporal, absoluto o relativo.

extiende (extends)

Una relación entre un caso de uso y otro que especifica cómo, el comportamiento definido para el primero, puede ser insertado en el comportamiento definido para el segundo.

Índice

–F–

faceta del modelo (model aspect)

Una dimensión del modelo que enfatiza cualidades particulares del metamodelo. Por ejemplo, la faceta estructural del modelo, enfatiza las cualidades estructurales del modelo.

faceta estructural del modelo (structural model aspect)

Una faceta del modelo que enfatiza la estructura de los objetos del sistema, incluyendo sus tipos, clases, relaciones, atributos y operaciones.

faceta funcional del modelo (behavioral model aspect)

Una faceta del modelo que enfatiza el comportamiento de los objetos de un sistema, incluyendo sus métodos, interacciones, colaboraciones, e historia de estados.

foco de control (focus of control)

Un símbolo de un diagrama de secuencias que muestra el período de tiempo durante el cual un objeto está realizando una acción, ya sea directamente o por medio de un procedimiento subordinado.

Indice

–G–

generalización (generalization)

Una relación taxonómica entre un elemento más general y uno más específico. El elemento más específico es totalmente consistente con el más general y contiene más información. En los lugares donde están permitidas instancias del elemento más general pueden utilizarse instancias del más específico. Sinónimo: generalization [OMA]. Ver: herencia.

Indice

–H–

herencia (inheritance)

El mecanismo por el cual elementos más específicos incorporan la estructura y el comportamiento de elementos más generales. Ver: generalización. Sinónimo: inheritance [OMA].

herencia de implementación (implementation inheritance)

La herencia de la implementación por parte de un elemento más específico. Incluye la herencia de interface. Sinónimo: implementation inheritance [OMA]. Contraste: herencia de interface.

herencia de interface (interface inheritance)

La herencia de la interface de un elemento más general. No incluye herencia de la implementación. Sinónimo: interface inheritance [OMA]. Contraste: herencia de implementación.

herencia múltiple (multiple inheritance)

Una variación semántica de la generalización en la cual un tipo puede tener más de un supertipo. Contraste: herencia simple.

herencia simple (single inheritance)

Una variación semántica de la generalización en la que un tipo puede tener solo un supertipo. Sinónimo: single inheritance [OMA]. Contraste: herencia múltiple.

hilo [de control] (thread [of control])

Un curso determinado a través de la ejecución de un programa, un modelo dinámico o alguna otra representación del flujo de control. Ver: proceso.

Indice

–I–

implementación (implementation)

La definición de cómo está construido o compuesto algo. Por ejemplo: una clase es una implementación de un tipo, un método es una implementación de una operación. Sinónimo: implementation [OMA].

importa (import)

En el contexto de un paquete, la dependencia que muestra a los paquetes cuyas clases pueden ser referidas desde el paquete dado (incluyendo recursivamente a los paquetes embebidos en él). Contraste: import [OMA], exportar.

instancia (instance)

Un individuo descrito por una clase o un tipo. Nota de uso : De acuerdo con la interpretación estricta del metamodelo un individuo de un tipo es una instancia y un individuo de una clase es un objeto. Es aceptable, en un contexto menos formal, referirse a un individuo de una clase como un objeto o una instancia. Ver: tipo. Contraste: objeto.

instancia de caso de uso (use case instance)

Una secuencia de acciones realizadas por el sistema, que producen un resultado observable, valioso para un actor en particular. Usualmente los escenarios ilustran instancias prototípicas de casos de uso. Una instancia de una clase caso de uso. Ver: caso de uso.

interacción (interaction)

Una especificación de comportamiento cuyo fin es lograr un propósito específico. Abarca un conjunto de intercambios de mensajes entre un conjunto de objetos dentro de un contexto particular. Una interacción puede ilustrarse mediante uno o más escenarios.

interface (interface)

La utilización de un tipo para describir el comportamiento visible de una clase, objeto u otra entidad. En el caso de una clase o un objeto, la interface incluye la signatura de las operaciones. Sinónimo: interface [OMA]. Ver: tipo.

Indice

–J–

(vacío)

Indice

–K–

(vacío)

Indice

–L–

línea de vida de un objeto (object lifeline)

Una línea, en un diagrama de secuencias, que representa la existencia de un objeto durante un período de tiempo. Ver: diagrama de secuencias.

lista (list)

Un contenedor cuyo contenido está ordenado.

literal (string)

Una secuencia de caracteres de texto. Los detalles de representación de los literales dependen de la implementación y pueden incluir conjuntos de caracteres que soporten gráficos y caracteres internacionales.

Indice

–M–

máquina de estados (state machine)

Un comportamiento que especifica las secuencias de estados por los que atraviesa un objetos o una interacción durante su ciclo de vida en respuesta a eventos, incluyendo sus contestaciones y acciones.

marca temporal (timing mark)

Una denotación del momento en que un evento o mensaje ocurre. Las marcas temporales pueden ser utilizadas en restricciones.

mensaje (message)

Una comunicación entre objetos que transmite información con la expectativa de desatar una acción. La recepción de un mensaje es, normalmente, considerada un evento.

mensaje asincrónico (asynchronous message)

Un mensaje donde el objeto emisor no se detiene a esperar los resultados. Sinónimo: asynchronous request [OMA]. Contraste: mensaje sincrónico.

mensaje sincrónico (synchronous message)

Un mensaje en el que el objeto emisor detiene su ejecución en espera del resultado. Sinónimo: synchronous request [OMA]. Contraste: mensaje asincrónico.

meta-metamodelo (meta-metamodel)

Un modelo que define el lenguaje para expresar el metamodelo. La relación entre meta-metamodelo y metamodelo es análoga a la relación entre metamodelo y modelo.

metaclase (metaclass)

Una clase cuyas instancias son clases. Las metaclases son típicamente utilizadas para construir metamodelos.

metamodelo (metamodel)

Un modelo que define el lenguaje para expresar un modelo. Una instancia de un meta-metamodelo.

metaobjeto (metaobject)

Un término genérico para referirse a todas las metaentidades que componen un lenguaje de metamodelado. Por ejemplo: metatipos, metaclases, metaatributos y metaasociaciones. Sinónimo: metaobject [OMA].

metatipo (metatype)

Un tipo cuyas instancias son tipos. Los metatipos son típicamente utilizados para construir metamodelos.

método (method)

La implementación de una operación. El algoritmo o procedimiento que permite llegar al resultado de una operación. Sinónimo: method [OMA].

miembro (member)

Una parte de un tipo o una clase que denota o un atributo o una operación.

modelo (model)

Una abstracción semánticamente consistente de un sistema. Ver: sistema.

modelo de casos de uso (use case model)

Un modelo que describe los requerimientos funcionales de un sistema en términos de casos de uso.

módulo (module)

Una unidad de manipulación y almacenamiento de un software. Incluyen : módulos de código fuente, módulos de código binario, módulos de código ejecutable. Ver : componente.

molde (template)

Sinónimo: clase paramétrica.

multiplicidad (multiplicity)

Una especificación del rango permitido de cardinalidades que puede asumir un conjunto. Es posible especificar la multiplicidad de : roles de una asociación, partes de un compuesto, repeticiones, y otros. Una multiplicidad es, esencialmente, un subconjunto (posiblemente infinito) de enteros no negativos. Contraste: cardinalidad.

Indice

–N–

no interpretado (uninterpreted)

Un lugar reservado para un tipo o tipos cuya implementación no está especificada en el UML. Cada valor no interpretado tiene una representación como literal. Ver: any [CORBA].

nodo (node)

Un nodo es un objeto físico existente en tiempo de ejecución, que representa un recurso computacional, que generalmente tiene al menos memoria y habitualmente también capacidad computacional. Objetos de tiempo de ejecución y componentes pueden residir en nodos.

nombre (name)

Un literal utilizado para identificar un elemento del modelo.

nota (note)

Un comentario vinculado a un elemento o colección de elementos. Una nota no tiene semántica. Contraste: restricción.

Indice

–Ñ–

(vacío)

Indice

–O–

objeto (object)

Una entidad delimitada precisamente y con identidad, que encapsula estado y comportamiento. El estado es representado por sus atributos y relaciones, el comportamiento es representado por sus operaciones y métodos. Un objeto es una instancia de una clase. Sinónimo: object [OMA]. Ver: clase, instancia.

objeto activo (active object)

Un objeto que es dueño de un hilo de ejecución y que puede iniciar una actividad de control. Una instancia de una clase activa. Ver: clase activa.

objeto efímero (transient object)

Un objeto que existe solo durante la ejecución del proceso que lo creó. Sinónimo: transient object [OMA].

objeto persistente (persistent object)

Un objeto que existe más allá del proceso que lo creó. Sinónimo: persistent object [OMA].

operación (operation)

Un servicio que puede ser requerido a un objeto para producir un comportamiento. Una operación tiene una signatura, que puede restringir sus parámetros posibles. Sinónimo: operation [OMA].

Indice

–P–

paquete (package)

Un mecanismo de propósito general para organizar elementos en grupos. Es posible incluir paquetes dentro de paquetes. Un sistema puede pensarse como un paquete de nivel superior, con todo el resto del sistema contenido en él.

parámetro (parameter)

La especificación de una variable que puede ser pasada, cambiada y/o devuelta. Un parámetro puede incluir nombre, tipo y dirección. Los parámetros son utilizados para operaciones, mensajes y eventos. Sinónimo: parameter [OMA], parámetro formal. Contraste: argumento.

parámetro formal (formal parameter)

Sinónimo: parámetro.

parámetro real (actual parameter)

Sinónimo: argumento.

participa (participates)

Una relación que indica el rol que una instancia juega en un elemento del modelo. Por ejemplo, una clase participa en una asociación, un actor participa en un caso de uso. Contraste: participate [OMA].

postcondición (postcondition)

Una restricción que debe ser verdadera al terminar una operación.

precondición (precondition)

Una restricción que debe ser verdadera cuando una operación es invocada.

proceso (process)

Un hilo de ejecución que puede ejecutar concurrentemente con otros hilos.

proceso de desarrollo (development process)

Un conjunto parcialmente ordenado de pasos, realizados durante el desarrollo de software con el fin de lograr un objetivo dado; por ejemplo, construir modelos o implementar modelos.

producto (product)

Los artefactos del desarrollo de software. Por ejemplo : modelos, código, documentación, planes de trabajo.

propiedad (property)

Un valor, identificado mediante un nombre, que denota una característica de un elemento. Una propiedad tiene impacto semántico. Algunas propiedades están predefinidas en el UML, otras pueden ser definidas por el usuario. Ver: valor rotulado. Sinónimo: property [OMA].

proveedor (supplier)

Un tipo, clase o componente que provee servicios que pueden ser invocados por otros. Sinónimo: server object [OMA]. Contraste: cliente.

proyección (projection)

El mapeo de un conjunto a un subconjunto de sí mismo.

proyección en una vista (view projection)

La proyección de elementos del modelo en elementos de una vista. Una proyección en una vista provee de la posición y estilo de presentación para cada elemento de la vista.

pseudoestado (pseudo-state)

Un vértice dentro de una máquina de estados que tiene la forma de un estado pero que no se comporta como tal. Los pseudoestado incluyen el inicial, el final y el historia.

punto de variación semántica (semantic variation point)

Un punto de variación en la semántica del metamodelo. Provee, intencionalmente, un grado de libertad de interpretación de la semántica del metamodelo. Ver: variación semántica.

Indice

–Q–

(vacío)

Indice

–R–

receptor (receiver [object])

El objeto que está tratando un mensaje pasado por un objeto emisor. Contraste: emisor.

recibir [un mensaje] (receive [a message])

Tratar un mensaje pasado por un objeto emisor. Ver: emisor, receptor.

referencia (reference)

La denotación de un elemento del modelo.

refinamiento (refinement)

Una relación que representa la especificación completa de algo que ya ha sido especificado con un cierto nivel de detalle. Por ejemplo, una clase del diseño es un refinamiento de una clase del análisis.

relación (relationship)

Una conexión semántica entre elementos del modelo. La asociación y la generalización son ejemplos de relaciones.

requerimiento (requirement)

Una característica, propiedad o comportamiento deseado para un sistema.

responsabilidad (responsibility)

Un contrato u obligación de una clase o tipo.

restricción (constraint)

Una condición u obligación semántica. Algunas restricciones están predefinidas en el UML, otras pueden ser definidas por el usuario. Las restricciones son uno de los tres mecanismos de extensión del UML. Ver: valor rotulado, estereotipo. Contraste: nota.

reutilizar (reuse)

La utilización de un artefacto preexistente.

rol (role)

El nombre que identifica al comportamiento específico de una entidad, cuando ésta participa de un contexto particular. Un rol puede ser estático (e.g. un rol de asociación), o dinámico (e.g. un rol en una colaboración).

rol de asociación (association role)

El rol que juega un tipo o una clase en una asociación.

rol de vinculación (link role)

Una instancia de un rol de asociación. Ver: rol de asociación.

Indice

–S–

señal (signal)

Un evento, identificado mediante un nombre, que puede ser invocado explícitamente. Las señales pueden tener parámetros. Una señal puede ser difundida (broadcast) o dirigida a un objeto, o grupo de objetos, en particular.

signatura (signature)

El nombre y parámetros de una operación, mensaje o evento. Opcionalmente los parámetros pueden incluir un parámetro devuelto como resultado de la operación, mensaje o evento. Sinónimo: signature [OMA].

sistema (system)

Una colección de unidades conectadas entre sí, que están organizadas para llevar a cabo un propósito específico. Un sistema puede describirse mediante uno o más modelos, posiblemente desde puntos de vista distintos.

subclase (subclass)

En una relación de generalización la clase que es la especialización de la otra (la superclase). Ver: generalización. Contraste: superclase.

subestado (substate)

Un estado que es parte de un estado compuesto. Un subestado puede ser concurrente o disjunto. Ver: subestado concurrente, subestado disjunto.

subestado concurrente (concurrent substate)

Un subestado que puede tener cabida simultáneamente a otros subestados concurrentes en el mismo estado compuesto. Ver: estado compuesto. Contraste: subestado disjunto.

subestado disjunto (disjoint substates)

Un subestado que no puede tener cabida simultáneamente con otros subestados concurrentes en el mismo estado compuesto. Ver: estado compuesto. Contraste: subestado concurrente.

subsistema (subsystem)

Un sistema subalterno a un sistema mayor. Un el UML un subsistema es modelado como un paquete de componentes. Contraste: sistema.

subtipo (subtype)

En una relación de generalización el tipo que es la especialización del otro (el supertipo). Ver: generalización. Contraste: supertipo.

superclase (superclass)

En una relación de generalización la clase que es generalización de la otra (la subclase). Ver: generalización. Contraste: subclase.

supertipo (supertype)

En una relación de generalización el tipo que es generalización del otro (el subtipo). Sinónimo: supertype [OMA]. Ver: generalización. Contraste: subtipo.

supratipo (powertype)

Un metatipo cuyas instancias son subtipos de otro tipo. Por ejemplo, la Especie Arbol es un supratipo del tipo Arbol. Los subtipos de Arbol (por ejemplo: Manzano, Ombú, Ceibo) son, por consiguiente, instancias de Especie Arbol.

Indice

–T–

tiempo (time)

Un valor que representa un momento en el tiempo, absoluto o relativo.

tiempo de análisis (analysis time)

Hace referencia a algo que ocurre durante la fase de análisis del proceso de desarrollo de software. Ver: tiempo de diseño, tiempo de modelado.

tiempo de compilación (compile time)

Se refiere a algo que sucede durante la compilación de un módulo de software. Ver: tiempo de modelado, tiempo de ejecución.

tiempo de diseño (design time)

Hace referencia a algo que ocurre durante una fase de diseño del proceso de desarrollo de software. Ver: tiempo de modelado. Contraste: tiempo de análisis.

tiempo de ejecución (run time)

El período de tiempo durante el cual un programa ejecuta. Contraste: tiempo de modelado.

tiempo de modelado (modeling time)

Se refiere a algo que sucede durante la fase de modelado del proceso de desarrollo de software. Incluye el tiempo de análisis y el tiempo de diseño. Nota sobre el uso : Al hablar de sistemas de objetos a menudo es importante distinguir entre tiempo de modelado y tiempo de ejecución. Ver : tiempo de análisis, tiempo de diseño. Contraste: tiempo de ejecución.

tipo (type)

Una descripción de un conjunto de instancias que comparten las mismas operaciones, atributos abstractos, relaciones abstractas, y semántica. Un tipo puede definir la especificación de una operación (como su signatura) pero no su implementación (método). Nota sobre el uso: A veces tipo e interface se usan como sinónimos pero no son términos equivalentes. Sinónimo: type [OMA]. Ver: clase, instancia. Contraste: interface.

tipo primitivo (primitive type)

Un tipo básico predefinido. Por ejemplo : Entero o Literal.

transición (transition)

Una relación entre dos estados que indica que un objeto que está en el primer estado realizará una acción especificada y entrará en el segundo estado cuando un evento especificado ocurra y unas condiciones especificadas sean satisfechas. En dicho cambio de estado se dice que la transición es 'disparada'.

Indice

–U–

unidad de distribución (distribution unit)

Un conjunto de objetos o componentes que son alocados a un proceso o procesador como grupo. En el UML una unidad de distribución puede ser representada mediante un compuesto o un agregado en tiempo de ejecución.

usa (uses)

Una relación entre un caso de uso concreto y un caso de uso abstracto en la cual, el comportamiento definido en el caso de uso concreto, emplea el comportamiento definido en el caso de uso abstracto.

utilidad (utility)

Un estereotipo de un tipo que agrupa variables globales y procedimientos en una declaración de clase. Los atributos y operaciones de una utilidad se convierten en variables y operaciones globales. Una utilidad no es fundamental para el modelado, pero puede ser conveniente durante la programación.

Indice

–V–

valor (value)

Un elemento del dominio de un tipo. Contraste: value [OMA].

valor rotulado (tagged value)

La definición explícita de una propiedad como un par nombre – valor. En un valor rotulado, el nombre es el 'rótulo'. Algunos rótulos están predefinidos en el UML, otros pueden ser definidos por el usuario. Los valores rotulados son uno de los tres mecanismos de extensión del UML. Ver: restricción, estereotipo.

variación semántica (semantic variation)

Una interpretación en particular elegida en un punto de variación semántica. Ver: punto de variación semántica.

vértice (vertex)

El origen o destino de una transición en una maquina de estados. Un vértice puede ser un estado o un pseudoestado. Ver: estado, pseudoestado.

vínculo (link)

Una conexión semántica entre una tupla de objetos. Una instancia de una asociación. Sinónimo: link [OMA]. Ver: asociación.

visibilidad (visibility)

Una enumeración cuyos valores (publico, protegido, privado o implementación) denotan cómo, el elemento del modelo al que se refieren, puede ser observado fuera del espacio de nombres al que pertenece.

vista (view)

Una proyección de un modelo, que es observado desde una perspectiva dada y que omite entidades que no son relevantes para esa perspectiva.

Indice

–W–

(vacío)

Indice

–X–

(vacío)

Indice

–Y–

(vacío)

Indice

–Z–

(vacío)

Indice

– # –

(vacío)

Indice

<u>Documentación del UML</u> <u>getRational!</u> <u>Baufest</u>
