**UML: Unified Modeling Language**

El objetivo de UML es “proporcionar a desarrolladores de software, arquitectos de sistemas e ingenieros de software de herramientas para el análisis, diseño e implementación de sistemas basados en software, así como modelar procesos de negocio y similares. El modelado captura las partes esenciales del sistema Introducción. UML es un lenguaje con un alcance muy grande y que cubre diversos conjuntos de dominios arquitectónicos en el diseño de aplicaciones. Por ello, no todas sus capacidades de modelados son necesariamente útiles en todos los dominios o aplicaciones.

La herencia permite definir nuevas clases partiendo de otras ya existentes. Las clases que derivan de otras heredan automáticamente todo su comportamiento, pero además pueden introducir características particulares propias que las diferencian. La abstracción permite seleccionar las características relevantes dentro de un conjunto e identificar comportamientos comunes para definir nuevos tipos de entidades en el mundo real. La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella podemos llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se quiere atacar. El encapsulamiento permite reunir a todos los elementos.

Un objeto no es más que un conjunto de variables (o datos) y métodos (o funciones) relacionados entre sí. Los objetos en programación se usan para modelar objetos o entidades del mundo real

En base a su comportamiento, se pueden establecer varias categorías de objetos: Activos – Pasivos Activo: posee un hilo de ejecución propio y puede iniciar una actividad

Pasivo: no puede iniciar una actividad, pero puede enviar estímulos una vez que se le solicita un servicio Clientes – Servidores, Agentes Cliente es el objeto que solicita un servicio.

Agregacion. Los componentes y la clase que constituyen son una asociación que conforma un todo. Puede presentarse una agregación como una jerarquía dentro de la clase completa en la parte superior, y los componentes por debajo de la línea de ella. Una línea conectara el todo de un componente mediante un rombo sin relleno que se colocara en la línea más cercana del todo.

Restricción de las agregacionesEn ocasiones el conjunto de componentes posibles en una agregación se establece dentro de una relación O, para modelar esto, se utilizaría una restricción: la palabra O centro de llaves con una línea discontinua que conecte las dos líneas que conforman al todo. ComposicionesUna composición es un tipo muy representativo de una agregación.

El símbolo de una composición es el mismo que el de una agregación, excepto que el rombo esta relleno. ContextosCuando se modela un sistema se podría producir, con frecuencia, agrupamiento de clases, como agregación y composiciones.

Las composiciones figuran en gran medida dentro de los diagramas de contexto. Un diagrama de contexto es como un mapa detallado de alguna sección de un mapa de mayores dimensiones. Pueden ser necesarias varias secciones para capturar toda la información detallada. Interfaces y realizacionesuna interfaz es un conjunto de operaciones que especifica cierto aspecto de la funcionalidad de una clase, y es un conjunto de operaciones que una clase presenta a otras. (Captura el conjunto reutilizable.) La interfaz puede establecer un subconjunto de las operaciones de una clase y no necesariamente todas ellas para modelar una interfaz se hace con un símbolo rectangular, una interfaz no tiene atributos.

La relación entre una clase y una interfaz se conoce como una realización. Esta relación esta modelada como una línea discontinua con una punta de flecha en forma de triángulo sin rellenar que adjunte y apunte a la interfaz

Una línea puede realizar más de una interfaz, y una interfaz puede ser realizada por más de una clase. Visibilidad este concepto está muy relacionado con las interfaces y la realización.

En el nivel privado solo la clase original puede utilizar el atributo u operación. La realización como podría imaginar, implica que el nivel público se aplique a cualquier operación en una interfaz. La protección de operaciones mediante cualquiera de los otros niveles tal vez no tendría sentido, dado que una interfaz se orienta a ser realizada por diversas clases. Para indicar el nivel público, anteceda el atributo u operación con un signo de suma (+), para revelar un nivel protegido antecédalo con un símbolo (#), y para indicar el nivel privado antecédalo con un (-).

Hay dos tipo de ámbitos, el de instancia y el de archivador. En el primero cada instancia cuenta con su propio valor en un atributo u operación.

Un atributo u operación con el ámbito de archivador, aparece con sus nombres subrayados. Este tipo de ámbito se utiliza con frecuencia cuando un grupo especifico de instancias (ningunas otras) tiene que compartir los valores exactos de un atributo privado. Un caso de uso describe un conjunto de secuencias, en las cuales cada secuencia representa la interacción de cosas fuera del sistema (actores) con el sistema (y sus abstracciones clave).

Por ejemplo, un caso de uso central de un banco es procesar préstamos. Los diagramas de caso de uso son centrales para modelar el comportamiento de un sistema, subsistema o una clase. Cada uno muestra un conjunto de casos de uso y actores y sus relaciones.

Los diagramas de caso de uso son importantes para visualizar, especificar y documentar el comportamiento de un elemento. Hacen al sistema, subsistema y clases accesibles y entendibles al presentar una vista externa de cómo aquellos elementos pueden ser usados en contexto. Sistemas de software intensivos pueden ser como lo anterior. Si eres un usuario, pudieras tener una aplicación y solicitarte que la uses. Si la aplicación sigue las convenciones normales del sistema operativo que acostumbras, puede ser que hagas algo útil después de un rato, pero no tendrás una comprensión de su comportamiento más detallado y complejo de esta manera. Similarmente, si eres un desarrollador, puede ser que tengas una aplicación heredada o un conjunto de componentes y te dicen que los uses. Estarás presionado para saber cómo usar los elementos hasta que te hayas formado un modelo conceptual de su uso.

Es importante recordar que el diagrama de secuencias se realiza a partir de la descripción de un caso de uso. El diagrama de colaboración es un tipo de diagrama de interacción cuyo objetivo es describir el comportamiento dinámico del sistema de información mostrando cómo interactúan los objetos entre sí.

Los diagramas de actividad con un caso especial de las máquinas de estados. Si con éstas representábamos el comportamiento de un solo elemento, con los diagramas de actividad representamos procesos en los que interviene más de un clasificador y consideramos necesario representar cómo colaboran esos clasificadores entre sí.