



Lectura aprendiendo UML en 24 horas (Parte II)

UNIDAD 2

Presenta: Mauricio Uriel Puga Martínez

Docente: Eduardo Flores Gallegos

Desarrollo, Aplicación y Consultoría de sistemas

Contenido

Hora 13 Diagramas de distribución.....	2
Hora 14 Nociones de los fundamentos del UML.	3
Hora 15 Adaptación del UML en un proceso de desarrollo.....	3
Hora 16 Presentación del caso por estudiar.	3
Hora 17 Elaboración de un análisis de dominio.....	3
Hora 18 Recopilación de las necesidades del sistema.	4
Hora 19 Desarrollo de los casos de uso.	4
Hora 20 Orientación a las interacciones y cambios de estado.	4
Hora 21 Diseño del aspecto, sensación y distribución.....	4
Hora 22 Noción de los patrones de diseño.	5
Hora 23 Modelado de sistemas incrustados.....	5
Hora 24 El futuro de UML.	5
Bibliografía	6

Hora 13 Diagramas de distribución.

El diagrama UML de distribución muestra cómo va a lucir el sistema cuando sea conjugado (además son útiles para representar redes). En estos diagramas los sistemas se representan por cubos y cada cubo representa un nodo. Las conexiones exitosas se representan por líneas que unen a cada cubo. Hay dos tipos de nodos, procesador el cual ejecuta componentes y dispositivos que no lo pueden hacer; los dispositivos por lo general interactúan con el mundo.

Hora 14 Nociones de los fundamentos del UML.

“El UML consta de cuatro capas: usuario, modelado, metamodelado y metametamodelado (que van desde específicos hasta generales). Cuando analice un sistema, típicamente trabajará en las primeras dos capas, cuando aprenda los conceptos UML, por lo general se encontrará con una tercera. La cuarta esta orientada a los teóricos y diseñadores de lenguajes, en lugar de los usuarios de lenguajes y analistas del sistema”

La hora 14 es un repaso de los conceptos generales vistos en las horas anteriores.

Hora 15 Adaptación del UML en un proceso de desarrollo

Una metodología de desarrollo estructura los segmentos y actividades de un proyecto de desarrollo de sistema. Pero esto en el pasado no se solía comprender; no había entendimiento de parte de los usuarios tanto como de los desarrolladores por lo que se tendía a utilizar el método clásico de cascada el cual consiste en análisis diseño, codificación y distribución.

Los defectos del modelo “cascada” van en que no se aprovecha al máximo la asimilación que se tiene de un proyecto durante la vida de este, además de que el tiempo de codificación es tan grande que se le resta al análisis y diseño.

GRAPPLE (Directivas para el Rápido Diseño de Aplicaciones), es un patrón para el proceso de desarrollo el cual consta de cinco segmentos (Recopilación de necesidades, análisis, diseño, desarrollo y distribución). Cada segmento consta de diversas acciones, y cada una de ellas arrojan como resultado un producto del trabajo. Los diagramas UML forman productos del trabajo para varias acciones.

Hora 16 Presentación del caso por estudiar.

En esta hora se aplicó un estudio de caso en donde se aplicó UML, en un proceso de desarrollo. El escenario que se planteó consto de un consorcio ficticio en el cual se decidió incorporar tecnología de computo en el restaurante. Como primeros pasos se tuvieron que aplicar los primeros segmentos de GRAPPLE que son comprender el dominio y el recopilar las necesidades. En el mismo escenario planteo que los expertos tenían acceso a sus procesos de negocios, además de que dar una guía para cómo realizar una entrevista y de aquí realizar el modelo UML.

Hora 17 Elaboración de un análisis de dominio.

La entrevista anteriormente realizada da por resultados los fundamentos para el análisis del dominio. Los verbos, sustantivos y “construcciones verbales” se utilizan como candidatos para el desarrollo de clases inicial que definen al dominio del escenario planteado. Las clases a utilizar se agregan según se vayan a utilizar.

El modelador de objetos agregara <<sustancia>> a este diagrama mediante la derivación de clases abstractas, asociaciones y multiplicidades. La derivación de agregado u objetos compuestos ayudan a organizar el modelo. El tema de las entrevistas continua ya que estas ayudaran a formar el cuerpo completo del modelo, aunque es posible empezar a agregar atributos y operaciones en este punto.

Hora 18 Recopilación de las necesidades del sistema.

En esta parte y continuando con el escenario, se generó la visión del sistema para el consorcio ficticio, además de que se decidió que la agilización en la transmisión de la información era clave para el éxito del sistema y se decidió aplicar el uso de tecnologías para lograrlo.

También, en esta parte los desarrolladores se reúnen con los usuarios potenciales y los expertos del dominio para obtener los requerimientos del sistema, donde se obtuvo un diagrama de paquetes en donde cada paquete representa una sección principal de funcionalidad. Los casos de uso dentro de un paquete se basan en tal funcionalidad.

Hora 19 Desarrollo de los casos de uso.

En esta hora se trata de transmitir la importancia de que se debe comprender cada uno de los casos de uso a detalle para la comprensión del sistema.

“Un diagrama de caso de uso involucrará las especificaciones de una descripción del propio caso de uso, derivar las condiciones previas y resultante, y especificar los pasos. Un aspecto importante del análisis del caso de uso es que los componentes del sistema a ser evidentes.”

Hora 20 Orientación a las interacciones y cambios de estado.

Continuando con el proyecto del escenario planteado; se puso la atención a los componentes de sistema, para definirlos y saber el cómo iban a interactuar, con la finalidad de ayudar a los programadores en la parte de la codificación de los objetos del sistema y la forma en la que se comunican entre sí.

Una vez modelado la cooperación entre componentes, el sistema está más cercano a realizarse. Conforme se vayan modelando los componentes se puede encontrar casos en los que es mejor modificar algunos casos de uso.

Hora 21 Diseño del aspecto, sensación y distribución.

Cuando un proyecto se encuentra en el segmento de diseño, se hace énfasis en la interfaz de usuario y la distribución del sistema. El diseño de estas va guiados por los casos de uso y son de gran importancia. El diseño de la interfaz del usuario se ve como una forma <<artística>> y de una investigación científica.

La finalidad del diseño de la interfaz de uso es que se puedan completar cada uno del caso de usos planteados en las secciones pasadas.

“De forma simultánea con varios de los procesos del proyecto, el ingeniero en sistemas del equipo se orientará a la arquitectura física. La arquitectura esta conducida por los casos de uso dado que el uso del

sistema finalmente determinara la naturaleza física y la disposición del mismo. El ingeniero de sistema otorga un diagrama de distribución UML que muestra los nodos, los componentes del software que hay en cada nodo y las conexiones entre nodos. Aunque los detalles de distribución aparecerán en las etapas avanzadas del proceso GRAPPLE, no hay razón para no empezar en ellos en etapas previas.”

Hora 22 Noción de los patrones de diseño.

En una clase parametrizada se tienen parámetros desvinculados; Al vincular estos parámetros dará como resultado la creación de una clase. Cualquier clasificador UML podrá estar parametrizado. LA colaboración parametrizada sirve como la representación de un patrón de diseño (una solución que es útil en diversos dominios).

“Un patrón de diseño, se ocupa de que los objetos pasen una petición entre ellos hasta que puedan manejarla.”

Los patrones de diseño surgen en la etapa del diagrama de actividades. Los patrones de diseño ofrecen varias ventajas, permite utilizar las soluciones ya probadas con facilidad, incorpora componentes sólidos en los diseños y documentar claramente los documentos que genera.

Hora 23 Modelado de sistemas incrustados.

Un sistema incrustado es una computadora que se encuentra de otro dispositivo, por ejemplo, en un electrodoméstico. La programación de un sistema incrustado requiere un gran conocimiento del dispositivo donde estará el sistema. Hay dos tipos de sistemas incrustados, el tolerante el cual no tiene que cumplir con plazos rigurosos, y el estricto que los tiene que cumplir.

El tiempo, los subprocesos que son programas sencillos que son parte de una aplicación, y las interrupciones (dispositivos de hardware que dejan saber que a la CPU que ha ocurrido algo) son conceptos importantes dentro de los sistemas incrustados.

Un sistema operativo en tiempo real (RTOS) dirige el tráfico entre subprocesos e interrupciones, además que administra el tiempo que ocupa el CPU en cada subproceso. Un núcleo puede ser preferencial el cual da prioridad a los subprocesos más importante o puede ser cooperativo.

Hora 24 El futuro de UML.

En la última hora se dan ejemplos de donde se puede aplicar el modelado UML.

En este capítulo se aprendió que para modelar una GUI se tiene que establecer un diagrama híbrido que muestre las relaciones de los componentes y sus casos de uso, Eso ayuda a mostrar la evolución de una GUI conforme toma forma, y les da importancia a los diagramas de caso de uso.

En un sistema experto, las bases de conocimiento son las reglas de conocimiento; el componente que contiene el conocimiento de un experto en algún dominio humano, para lo cual se sugiere un diagrama en el cual se visualicen las reglas y restricciones internas. En un diagrama, un cuadro dividido en comportamientos modela la regla. Los comportamientos contienen un identificador, otro un resumen de la

parte del if, otro la parte del then y otras dos muestran las reglas relacionadas y se utilizan líneas para las reglas adyacentes entre las partes adecuadas de las reglas.

Bibliografía

[1] J. Schuller, Aprendiendo UML en 24 Horas, Juarez, Mexico: Prentice Hall.

