

Obtención del modelo de comportamiento

Antes de continuar con el diseño lógico de cómo funcionará la aplicación software, hay que estudiar y definir su comportamiento como una “caja negra”. El *comportamiento del sistema* es una descripción de *qué* hace el sistema sin explicar cómo lo hace. Una parte de esa descripción es un diagrama de secuencia del sistema. Otras partes comprenden los casos de uso y los contratos del sistema.

Diagrama de secuencia del sistema

Los casos de uso describen cómo interactúan los actores externos con el sistema software. Durante esta interacción, un actor genera eventos sobre un sistema, normalmente, solicitando alguna operación como respuesta.. Ese evento de solicitud inicia una operación sobre el sistema.

Es deseable aislar e ilustrar las operaciones que un actor externo solicita a un sistema, porque constituyen una parte importante de la comprensión del comportamiento del sistema. UML incluye los *diagramas de secuencia* (ver Seminario de diagramas de interacción) como notación que puede representar las interacciones de los actores y las operaciones que inician.

Un *diagrama de secuencia del sistema* (DSS) es un diagrama de secuencia de UML en el que se muestra cómo los eventos generados por los actores provocan la ejecución de una operación por el sistema, siendo visto este como una caja negra (ver transparencia 40 del tema 2.4)

Los pasos a seguir para la elaboración de los DSS, se muestran en las transparencias 41, 42, 43, 44, y 45 del tema 2.4. Como podéis comprobar los DSS se generan para cada caso de uso, a partir de la especificación de eventos que aparece en la descripción extendida de dicho caso de uso.

UML no define nada denominado diagrama de secuencia “del sistema”, sino simplemente diagrama de secuencia. La calificación se utiliza para subrayar su aplicación y para representar sistemas como cajas negras.

Contratos

Los casos de uso son el principal mecanismo para describir el comportamiento del sistema y, normalmente, es suficiente. Sin embargo, algunas veces se necesita una descripción más detallada. Los contratos describen el comportamiento detallado del sistema en función de los cambios de estado de los objetos del Modelo conceptual, después de la ejecución de una operación.

Secciones de un contrato

La descripción de cada sección del contrato se muestra en la transparencia 46 del tema 2.4

Nombre: se refiere al nombre de la operación tal y como aparece en el correspondiente diagrama de secuencia del sistema; hay que incluir parámetros de entrada (entre paréntesis y separados por comas) y parámetro de salida, si lo hubiera. El formato general para el nombre de la operación es el siguiente:

parámetro de salida = nombre de la operación (entrada1, entrada2,)

Responsabilidad: es una breve descripción informal de lo que debe hacer la operación

Tipo: clase o concepto responsable de la operación; en nuestro caso siempre será la clase que representa al sistema

Notas: sección opcional que permite hacer aclaraciones y comentarios

Excepciones: Situaciones que pueden causar un error y hacer que la ejecución de la operación se interrumpa.

Salida: mensaje o datos que proporciona la operación como salida

Precondiciones: Suposición acerca del estado del sistema o de los objetos del modelo conceptual antes de ejecutar la operación.

Es importante no confundir excepciones con precondiciones, aunque en ocasiones una misma situación se puede describir como excepción o como precondición, pero no como ambas. Por ejemplo, supongamos una operación que requiere consultar los datos de un cliente proporcionando como parámetro de entrada su DNI. Es evidente que hay que comprobar que el valor del parámetro que se proporcione existe. Esa comprobación se podría expresar como una excepción de la operación, es decir como una condición que si no se verifica puede ocasionar que se interrumpa la ejecución de la operación.

La misma condición se podría expresar como una precondición; en este caso, la condición se tiene que verificar antes de empezar la ejecución de la operación, y si no se satisface, la operación ni siquiera se inicia. Esa es la diferencia fundamental: una excepción se comprueba cuando ya se está ejecutando una operación, y una precondición se comprueba antes de iniciar la ejecución de la operación.

Poscondiciones

Las poscondiciones describen cambios en el estado de los objetos del modelo conceptual. Los cambios de estado del modelo conceptual comprenden la creación o eliminación de instancias, formación o destrucción de asociaciones y cambio en los atributos.

Las poscondiciones no son acciones que se ejecutarían durante la operación; más bien son declaraciones sobre los objetos del modelo conceptual que son verdad cuando la operación ha terminado.

La cualidad importante es ser declarativo y enunciar con un estilo orientado al cambio en lugar de orientado a la acción, puesto que las poscondiciones son declaraciones sobre los estados o resultados, en lugar de una descripción de las acciones a ejecutar, o un diseño de una solución.

Los contratos son una herramienta excelente para el análisis de requisitos que describen los cambios de estado que requiere una operación del sistema (en función de los objetos del modelo conceptual) sin tener que describir *cómo* se van a llevar a cabo. En otras palabras, uno se puede centrar, analíticamente en *qué* debe suceder, en lugar de en *cómo* se va a realizar.

Es conveniente expresar las poscondiciones en pasado, para resaltar que son declaraciones sobre un cambio de estado en el pasado.

Elaboración de un contrato

Para hacer contratos:

1. Identificar las operaciones del sistema a partir de los DSSs.
2. Construir un contrato para las operaciones del sistema complejas y quizás sutiles en sus resultados, o que no estén claras en el caso de uso.
3. Para describir las poscondiciones utilizar las siguientes categorías:
creación y eliminación de instancias
modificación de atributo
formación y destrucción de asociaciones.

Sobre la escritura de los contratos

- Establecer las poscondiciones de forma declarativa con una sentencia pasiva expresada en pasado, para destacar que se trata de una declaración de un cambio de estado en lugar del diseño de la manera en la que se va a realizar.
- Recordar que hay que establecer una relación entre los objetos existentes o aquellos creados recientemente mediante la definición de la formación de asociaciones. Por ejemplo, no es suficiente que se cree una instancia de *Proyecto* cuando tenga lugar la operación *definirProyecto* (ver transparencias 51 y 52 del tema 2.4). Después de que se complete la operación, también debería cumplirse que la nueva instancia creada se asoció con la instancia correspondiente de *Profesor*.

El problema más común es olvidarse de incluir la *formación de asociaciones*. En concreto, cuando se crean nuevas instancias, es muy probable que se necesiten establecer asociaciones con varios objetos.