

## Modelo de la estructura de objetos

Una vez terminados los diagramas de interacción (comunicación) para la realización (diseño) de los casos de uso, es posible identificar las clases e interfaces que participan en la solución software, y añadirles detalles de diseño, como los métodos. UML proporciona la notación para representar los detalles de diseño en los diagramas de clase.

El punto de partida para la elaboración del modelo de la estructura de objetos es el conjunto de diagramas de interacción (comunicación) que constituyen el modelo de interacción. Además, como guía para el desarrollo de este modelo se usará el modelo conceptual, resultado de la fase de análisis y especificación de requisitos.

### Diagrama de clases de diseño

En la práctica, los diagramas de clases de diseño (DCD) normalmente se crean en paralelo a la creación de los diagramas de interacción. Al comienzo del diseño se podrían esbozar muchas clases, nombres de métodos y relaciones aplicando los patrones para asignar responsabilidades, antes de la elaboración de los diagramas de interacción. Es posible y deseable elaborar algo de los diagramas de interacción, a continuación actualizar los DCD, extender después los diagramas de interacción algo más, y así sucesivamente.

Además de las asociaciones y atributos básicos, el diagrama se amplía para representar, por ejemplo, los métodos de cada clase, información del tipo de los atributos, visibilidad de los atributos y navegación entre los objetos.

Un *diagrama de clases de diseño* representa las especificaciones de las clases e interfaces software (p. e., las interfaces java) en una aplicación. Como información general se tiene:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Información acerca del tipo de los atributos.
- Navegabilidad.
- Dependencias.

El Proceso Unificado (PU) no define de manera específica ningún artefacto denominado “diagrama de clases de diseño”. El PU define el Modelo de Diseño, que contiene varios tipos de diagramas, entre los que se encuentran los diagramas de interacción, y los de clases. Los diagramas de clases del Modelo de Diseño contienen “clases de diseño”. De ahí que sea común hablar de “diagramas de clases de diseño”, que es más corto, e implica “diagrama de clases en el Modelo de Diseño”.

En el modelo conceptual, una *clase* no representa una definición software, sino que se trata de una abstracción de un concepto del mundo real, acerca del cual se está interesado en realizar una declaración. En cambio, los DCD expresan, para la aplicación software, la definición de las clases como componentes software

## Elaboración del diagrama de clases de diseño

Los pasos a seguir en la elaboración del diagrama de clases de diseño, se muestran en la transparencia 4.

## Identificar y representar las clases software

El primer paso en la creación de los DCD es identificar aquellas clases que participan en la solución software. Se pueden encontrar examinando todos los diagramas de interacción y listando las clases que se mencionen en dicho diagrama.

Las clases identificadas tomarán sus atributos del modelo conceptual y de los diagramas de interacción (comunicación)

Como ejemplo, vamos a considerar el diagrama de comunicación de la transparencia 6 (este diagrama es el que se desarrolló para la operación *definirProyecto* en el tema 3.3). Las clases que se identifican son (marcadas con un círculo rojo en la transparencia 5): *SAP*, *SAPExcepcion*, *Proyecto*, *Profesor*, *PlanEstudios* y *Asignatura*.

El siguiente paso es dibujar un diagrama de clases para estas clases e incluir los atributos que se identificaron previamente en el modelo conceptual y que también se utilizan en el diseño (ver transparencia 7)

## Identificar y añadir las operaciones (métodos)

Se pueden identificar los nombres de los métodos analizando los diagramas de interacción. Por ejemplo, si se envía el mensaje *crear* a una instancia de la clase *Proyecto*, entonces la clase *Proyecto* debe definir un método *crear* (ver transparencia 10). En general, el conjunto de todos los mensajes enviados a una clase X a lo largo de todos los diagramas de interacción, contiene la mayoría de los métodos que debe definir la clase X.

En la transparencia 9, se muestran (dentro de un rectángulo rojo) todos los envíos de mensaje que hay que considerar para definirlos como métodos en las clases identificadas. El resultado de la asignación de métodos se ilustra en la transparencia 10. Como se puede observar, el método se define en la clase a la que va dirigido el mensaje correspondiente.

## Añadir tipos de atributos y parámetros

Opcionalmente, se podrían mostrar todos los tipos de atributos, parámetros de los métodos y los valores de retorno. La cuestión de sobre si se muestra, o no, esta información se debe considerar en el siguiente contexto:

- Si el DCD se está creando en una herramienta CASE con generación automática de código, son necesarios todos los detalles y de modo exhaustivo.
- Si se está creando para que lo lean los desarrolladores de software, los detalles exhaustivos de bajo nivel podrían afectar negativamente por el nivel de ruido.

Por ejemplo, ¿es necesario mostrar todos los parámetros y la información de sus tipos? Depende de lo obvio que sea la información para la audiencia a la que está destinada.

El DCD de la transparencia 11 muestra más información sobre los tipos.

## Identificar y representar las asociaciones y su navegabilidad

Todos los enlaces estereotipados con <<A>>, en un diagrama de interacción, deben tener su correspondiente asociación en el DCD. En la transparencia 13 se muestran, dentro de un círculo rojo, los estereotipos asociación que dan lugar a las asociaciones correspondientes en el DCD. Una asociación se representa en un DCD con una línea continua

La *navegabilidad* es una propiedad que indica que es posible navegar unidireccionalmente a través de la asociación desde los objetos de la clase origen a la clase destino

La navegabilidad la proporciona la dirección del envío de mensaje y la multiplicidad la existencia de multiobjetos, como destino del envío (ver transparencia 14). La navegabilidad, se representa en un DCD con una punta de flecha, que indica hacia donde se puede navegar.

### Identificar y representar las relaciones de dependencia

Todos los enlaces estereotipados con <<L>>, <<P>> o <<G>> estarán en el DCD como una relación de dependencia, Las relaciones de dependencia se representan en un DCD como flechas de línea discontinua. El sentido de la flecha lo indica el sentido del envío correspondiente en el diagrama de comunicación.

Las relaciones de dependencia están implícitamente incluidas en las asociaciones, razón por la cual cuando entre dos clases de diseño se haya definido una asociación y una relación de dependencia, no es necesario representar la relación de dependencia (ver transparencia 16).