

Estilos del Tipo de Vista de Módulos

Tipos de Vista de Módulos

- Las vistas en el tipo de vistas de módulos documentan las principales unidades de implementación del sistema.
- Esencialmente se describen los *módulos* que componen el sistema.
- Pero,
 - ¿Qué es un módulo?
 - ¿Cuál es la diferencia con una componente?
 - ¿Qué relación existe entre estos dos conceptos, si es que hay alguna?

Módulos y Componentes

- Componente:
 - "...pieza de software autocontenida con una interfaz bien definida. Tiene una connotación de instalación y ejecución, o sea que una componente tiene interfaces accesibles durante la ejecución y, en algún momento del ciclo de vida de desarrollo, puede ser instalada en forma independiente."

Herzum & Sims (2000)
- Módulo:
 - esencialmente una entidad de diseño
 - encapsula responsabilidades,
 - define una interfaz de acceso.
- Los elementos que aparecen documentados en una vista de módulos tienen generalmente una manifestación en las vistas de C&C.

Módulos y Componentes

- Módulos

- unidades de diseño;
- encierran funcionalidad o responsabilidad que se accede a través de una interfaz [Parnas72];
- no se replica;
- no interesa la ejecución.

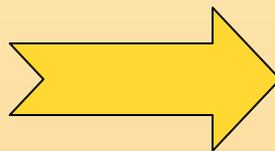
- Componentes

- unidad de ejecución;
- tiene una interfaz definida y accesible durante la ejecución;
- puede instalarse en forma independiente;
- puede replicarse.

Buenas Componentes y Buenos Módulos

- Una componente puede ser autocontenida e instalada en forma independiente solamente si proviene de un módulo que fue diseñado para encapsular apropiadamente su responsabilidad.

Plug and play



Plug and pray

Estilos de Tipo de Vistas de Módulos

- Existen esencialmente cuatro estilos:
 - Descomposición
 - Usos
 - Generalización
 - Capas

Descomposición

- El estilo de descomposición se obtiene al usar el tipo de vista de módulos y relacionar sus elementos mediante la relación *es-parte-de*.
- Se usa este estilo para mostrar cómo se distribuye la responsabilidad entre los módulos y cómo éstos se dividen en submódulos.
- Existen muy pocas restricciones en este estilo, pero es útil considerarlo como un estilo independiente:
 - todo diseño de arquitectura suele comenzar por descomposición,
 - es una herramienta apropiada para comunicación,
 - se comienza a abordar la modificabilidad del sistema.

Criterios de Descomposición

- Lograr ciertos atributos de calidad
 - asignar a ciertos módulos la funcionalidad con la cual está relacionada la exigencia de modificabilidad o performance.
- Decisiones de construir o comprar
 - si se reusan módulos o se los compra, éstos ya están diseñados y el resto de la funcionalidad debe asignarse teniéndolo en cuenta.
- Implementación de líneas de productos
 - debe identificarse los módulos que serán parte de todos o la mayor parte de los productos de aquellos módulos variables, propios de cada producto.

Elementos, Relaciones y Propiedades

- Los elementos básicos son los módulos.
- Ciertas agregaciones pueden llamarse subsistemas.
- La principal relación es la *descomposición*, una forma especializada de *es-parte-de*:
 - cada elemento puede ser parte solamente de un agregado.
- Puede definirse la *visibilidad* de la descomposición como una propiedad:
 - los módulos son visibles sólo para su contenedor (padre) o también para los demás;
 - si ningún módulo puede verse fuera del contenedor, la relación de descomposición se llama *contención*.

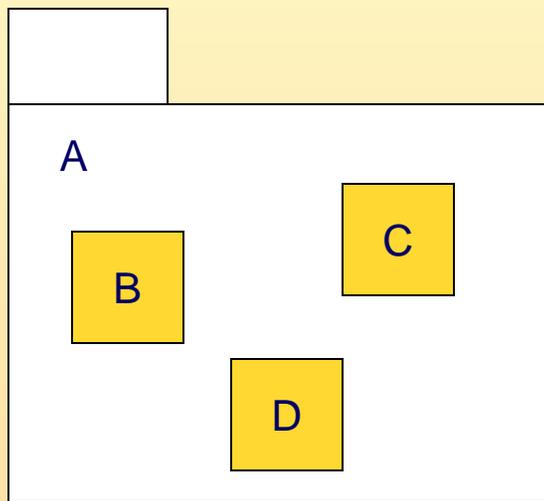
Aplicación de Descomposición

- El estilo de descomposición presenta la funcionalidad del sistema en piezas intelectualmente manejables, y que son refinadas recursivamente hasta obtener más detalles.
- Es un estilo apropiado para aprender acerca de los detalles del sistema.
- El agrupamiento de funcionalidad sirve de base para definir ítems de configuración en un esquema de control de la configuración en el desarrollo.
- Es el paso inicial para la asignación de tareas al equipo de desarrollo.

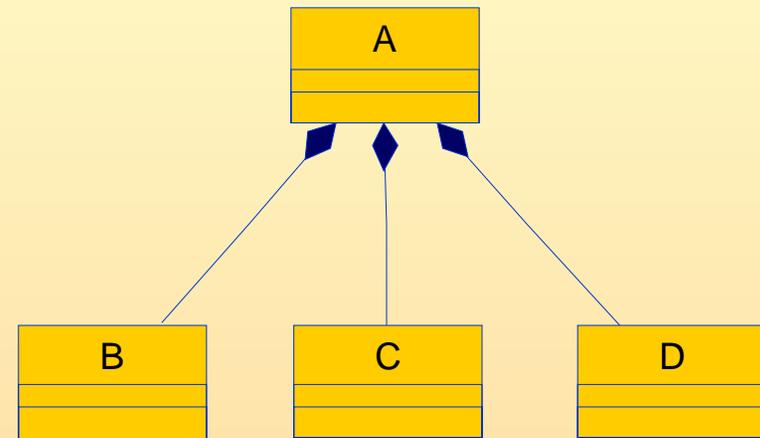
Relación con otros estilos

- Es en general deseable tener una idea de cómo se mapean los módulos y los componentes del sistema (generalmente una relación muchos a muchos).
- El estilo de descomposición está relacionado con el estilo de asignación de trabajo, parte del tipo de vistas de *allocation*.

Representación de Descomposición



Paquetes o subsistemas



Agregación

Descomposición y Subsistemas

- Un subsistema es un conjunto de módulos que:
 - lleva a cabo un conjunto cohesivo de funcionalidad, parte de la misión total del sistema;
 - puede ser ejecutado en forma (más o menos) independiente;
 - puede desarrollarse e instalarse en forma incremental.
- Es una estrategia apropiada para:
 - asignar trabajo a distintos grupos o a subcontratistas,
 - abordar atributos de calidad,
 - instalación incremental de funcionalidad parcial útil.

Resumen de Descomposición

Elementos	Módulos, tal como se definen en los tipos de vistas de módulos. Un módulo que agrega otros módulos puede llamarse subsistema.
Relaciones	Relación de descomposición, la cual es un refinamiento de la relación <i>es-parte-de</i> . Debe documentarse el criterio usado para definir la descomposición.
Propiedades de los elementos	Igual a las definidas para el tipo de vistas de módulos.
Propiedades de las relaciones	Visibilidad, si un módulo es conocido y su funcionalidad disponible para módulos distintos de su padre.
Topología	No pueden existir ciclos en un grafo de descomposición. Un módulo no puede ser parte de más de un módulo en la misma vista.

Estilo de Uso

- Este estilo se basa en la relación *depende-de* entre los módulos.
- Indica qué otros módulos deben existir para que una parte del sistema pueda funcionar correctamente.
- Este estilo apoya el desarrollo y la instalación incremental de partes útiles del sistema.

Elementos, Relaciones y Propiedades

- La relación *usa* es una especialización de *depende-de*
 - un módulo requiere la correcta implementación de otro módulo para su propio correcto funcionamiento.
- En este estilo se hace explícita la forma en que la funcionalidad se mapea a la implementación
 - relación entre elementos de código.
- El estilo de Uso se aplica para
 - planificar desarrollo incremental y extensiones al sistema,
 - depuración y pruebas,
 - análisis del impacto de cambios específicos.

Relación con otros estilos

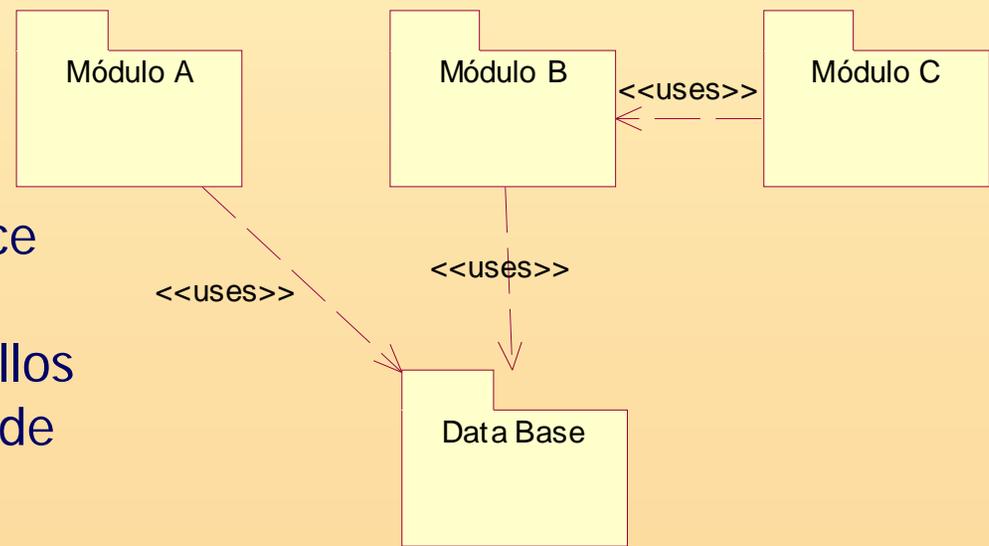
- El estilo de Uso está relacionado con el estilo de capas cuando se usa la relación *autorizado-a-usar*.
- Esta relación limita y ordena qué pueden hacer los implementadores de los módulos diseñados.

Descomposición y Uso



- User Interface tiene una dependencia de uso de la base de datos.

- Al descomponer User Interface en Módulo A, Módulo B y Módulo C, al menos uno de ellos debe tener una dependencia de uso de la base de datos



Resumen de Uso

Elementos	Módulos como en el tipo de vista de módulos.
Relaciones	Relación <i>usa</i> que es un refinamiento de la relación <i>depende-de</i> . Módulo A usa Módulo B si A depende de la presencia de un módulo B funcionando correctamente para satisfacer sus propios requerimientos.
Propiedades de los elementos	Los mismos definidos para el tipo de vista de módulos.
Propiedades de las relaciones	La relación <i>usa</i> puede tener una propiedad que describa en más detalle el tipo de uso que un módulo hace sobre el otro.
Topología	No tiene restricciones topológicas. Sin embargo, si existen ciclos en la relación, no se podrá hacer desarrollo incremental.