

Tecnologías de componentes y proceso de diseño de aplicaciones basado en componentes

“Programación orientada a objetos :
Lenguajes, Tecnologías y Herramientas”
Master de Computación

Patricia López
Grupo de Computadores y Tiempo Real

Santander, 2009

Origen del desarrollo basado en componentes (CBD)

- Origen: En 1968 Douglas McIlroy identifica la necesidad de una sub-industria de componentes software como solución a la denominada crisis del software.
- Su implantación no se realiza hasta los 90
 - En la actualidad está plenamente implantado de algunos dominios de aplicación: Ofimática, comercio electrónico, etc.
- Problemas que trata de solucionar el desarrollo basado en componentes:
 - Abordar la creciente complejidad del software
 - Reducir el tiempo de adaptación a cambios
 - Creciente importancia de sistemas abiertos y distribuidos
- El CBD es una evolución natural de la orientación a objetos

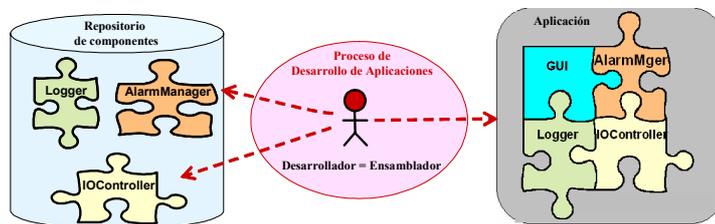
Santander, 2009

Patricia López

2

Objetivo del desarrollo basado en componentes

- Construir aplicaciones mediante **ensamblado** de módulos software **reutilizables**, que han sido diseñados previamente con **independencia** de las aplicaciones en las que van a ser utilizados.
 - Industrialización del campo del desarrollo software
 - Cambio en el papel de los diseñadores de aplicaciones => De programadores a ensambladores
 - Mercado de componentes software reutilizables



Santander, 2009

Patricia López

3

Ingeniería software basada en componentes (CBSE)

- CBD => Cambio profundo en los procesos de desarrollo de software
- Ingeniería Software basada en Componentes (CBSE):
 - Nueva disciplina dentro de la IS
 - Define los métodos y herramientas que dan soporte al desarrollo de sistemas software por ensamblado de componentes:
 - Soporte para la construcción de componentes software reutilizables.
 - Soporte para la construcción de sistemas por composición de dichos componentes.
 - Soporte para el mantenimiento de este tipo de sistemas, bien por sustitución o por introducción de nuevos componentes.

Santander, 2009

Patricia López

4

Definición de componente

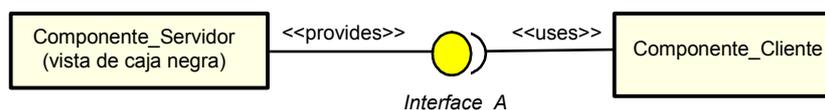
- Existen muchas definiciones de componente software, pero una de las más difundidas es la de Szyperski [1998]:

*“Un **componente** es una unidad de **composición** de aplicaciones software, que posee un **conjunto de interfaces** y un **conjunto de requisitos**, y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros componentes de forma **independiente**, en tiempo y espacio”*

- Las tres características principales de un componente software son:
 - **Aislamiento**: Un componente puede ser instalado de forma independiente en una plataforma.
 - **Componibilidad**: Un componente puede ser compuesto con otros para formar aplicaciones.
 - **Opacidad**: Un componente se maneja siempre de forma opaca, sin que los diseñadores de aplicaciones que lo manejan ni el entorno tengan que acceder a sus detalles internos para hacer uso de él.

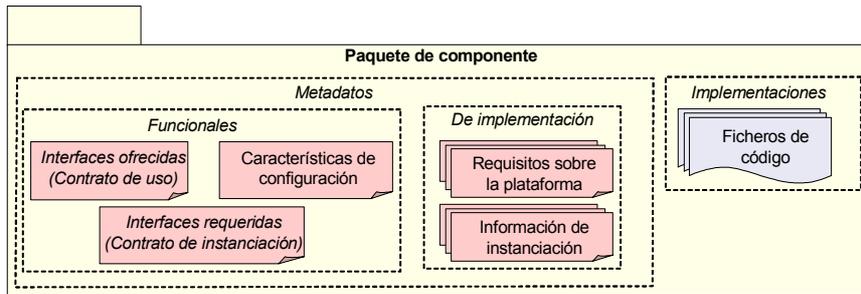
Importancia de las interfaces en CBD

- Los servicios que ofrece o requiere un componente se expresan mediante interfaces
 - Representan el contrato entre el componente cliente y el componente servidor
- Concepto base en desarrollo basado en componentes => Separación entre interfaz e implementación



Manejo opaco de componentes => Metadatos

- El componente se distribuye como un paquete que almacena todos los elementos que lo constituyen, y ofrece información introspectiva (metadatos) para poder hacer uso de él de forma opaca.

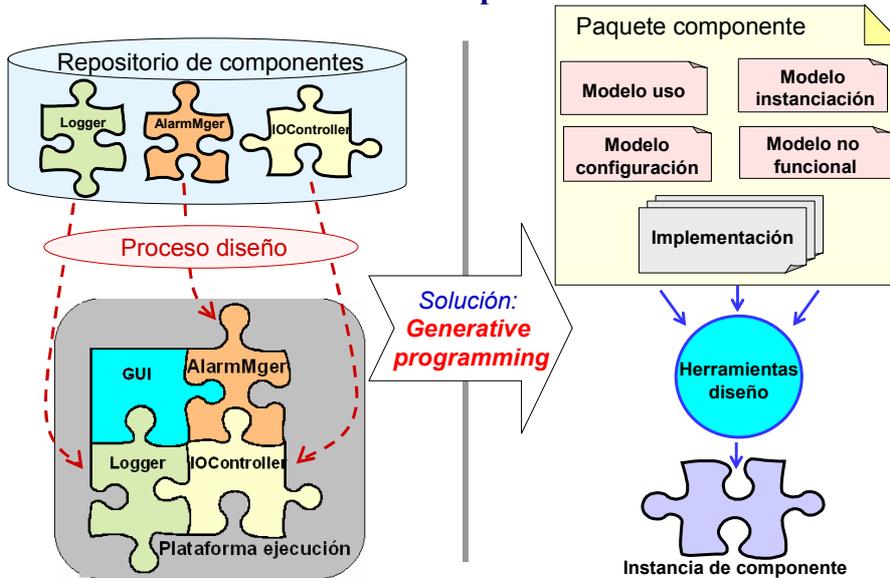


Santander, 2009

Patricia López

7

Proceso de diseño basado en componentes



Santander, 2009

Patricia López

8

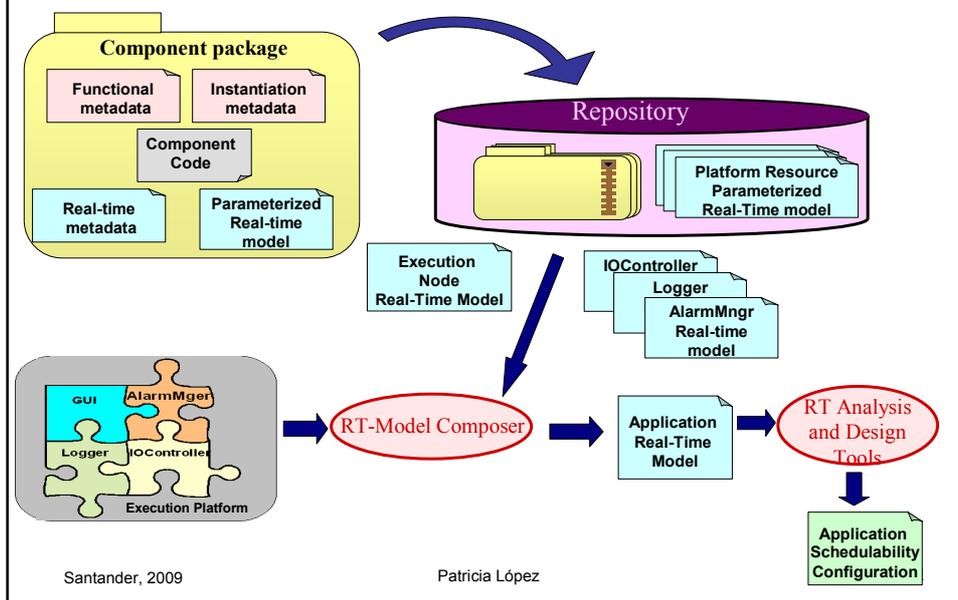
Tecnologías de componentes: Modelo + Plataforma

- La forma concreta de especificar, implementar, o empaquetar un componente depende de la tecnología utilizada.
- Una **tecnología** de componentes incluye:
 - **Modelo de componentes:** Define un marco conceptual para el desarrollo de componentes y de aplicaciones generadas por ensamblado:
 - Define de forma exacta que se entiende por componente, y como se especifica el componente y sus interfaces
 - Define los modos de interacción entre componentes
 - **Plataforma de componentes:** Entorno que de una infraestructura de soporte para la ejecución de aplicaciones basadas en componentes
 - Se basa en un determinado modelo de componentes
 - Incluyen servicios de soporte para los componentes: gestión de su ciclo de vida, persistencia, comunicación entre ellos, etc.
- Ejemplos de tecnologías de componentes:
 - De Microsoft: COM, COM+, .NET (Plataforma Windows)
 - De Sun Microsystems: JavaBeans, EJB (Multiplataforma, pero sólo en Java)
 - De OMG: CCM (Multilenguaje y multiplataforma)

Componentes de tiempo real

- En el ámbito de los sistemas de tiempo real el CBD evoluciona de forma mucho más lenta
 - Las tecnologías estándar son demasiado lentas y no predecibles
- Es necesario hacer compatibles los dos aspectos
 - Construcción de aplicaciones por composición de componentes software reutilizables y manejados de forma ópaca
 - Construcción de aplicaciones con comportamiento temporal predecible
- Para ello se requiere
 - Metodologías de modelado que ofrezcan propiedades de componibilidad y reutilización
 - Basadas en parametrización
 - Inclusión de información acerca del comportamiento temporal en los paquetes de componentes
 - Tecnologías de componentes con comportamiento predecible

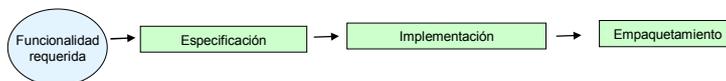
Modelado de aplicaciones de tiempo real basadas en componentes



Cambio en el proceso de desarrollo de aplicaciones

- En el desarrollo basado en componentes se distinguen dos procesos independientes pero complementarios (los productos de salida del primero son los productos de entrada del segundo):

Proceso de desarrollo de componentes



Proceso de desarrollo de aplicaciones



Ventajas del uso de CBD

- Mejora de la productividad
 - Gracias a la reutilización de software
 - En una primera fase se produce el efecto contrario, al invertir tiempo y esfuerzo en conseguir reusabilidad
- Disminución de la complejidad del software
- Mejora de los tiempos de acceso al mercado
 - Consecuencia de los dos aspectos anteriores
- Incremento de la calidad del software
 - Siempre que se emplee un mercado de componentes certificados
- Mejora del mantenimiento
 - Errores mas fáciles de detectar y subsanar
 - Posibilidad de actualización dinámica

Limitaciones actuales del CBD

- Disponibilidad de componentes
 - Sólo existe en algunos campos: GUIs, ofimática, etc
- Falta de estandarización
 - Intereses empresariales
 - Interoperabilidad a través de pasarelas “bridges”
- Falta de procesos de certificación que garanticen la calidad de los componentes
- Falta de un proceso de ingeniería software basado en componentes bien definido
 - Riesgo para las empresas de cambiar sus procesos de desarrollo