Fundamentos de la programación con componentes (Parte I)

Luis G. Montané Jiménez
Tópicos Selectos de Computación I
Facultad de Estadística e Informática
6 de Agosto de 2014

Agenda

- **Introducción**
- Componentes
- Arquitectura de Software
- Estructura de un componente

Introducción

- Actualmente sistemas complejos y de alta calidad se deben construir en períodos de tiempo cortos

- Esto obliga un enfoque de reutilización más organizado

Conceptos básicos

- Sistema
  - Extensible
  - Abierto
- Programación
  - Orientada a objetos-POO
  - Orientada a componentes POC

Conceptos básicos

- ***Sistema***
  - Conjunto de mecanismos y herramientas que permiten la creación e interconexión de componentes de software, junto con una colección de servicios para facilitar las labores de los componentes que residen y se ejecutan en él.

Conceptos básicos

- Un Sistema es independientemente **extensible si:**
  - Puede combinarse con **extensiones independientemente desarrolladas**
- Un Sistema es **abierto si:**
  - **Es concurrente, reactivo, independientemente extensible**
  - Permite a componentes heterogéneos ingresar o abandonar el sistema de forma dinámica

Conceptos básicos

- Los ***Sistemas abiertos*** son inherentemente evolutivos, y la vida de sus componentes es más corta que la del propio sistema.

- El desarrollo de aplicaciones para este tipo de sistemas se ve afectado por una serie de problemas específicos:
  - Gestión de la evolución del propio sistema y de sus componentes
  - Falta de una visión global del sistema
  - Dificultad para garantizar la seguridad y confidencialidad de los mensajes
  - Heterogeneidad de los componentes, o su dispersión, lo que puede implicar retrasos y errores en las comunicaciones.

Necesidad de nuevos modelos

- La programación tradicional se ha visto incapaz de tratarlos de una forma natural.
- La POO ha sido el sustento de la ingeniería del software para los sistemas cerrados.
- Sin embargo, se ha mostrado insuficiente al tratar de aplicar sus técnicas para el desarrollo de aplicaciones en entornos abiertos.

POO

- En particular, se ha observado que no permite expresar claramente la distinción entre los aspectos computacionales y meramente composicionales de la aplicación.

- Hace prevalecer la visión de **objeto** sobre la de **componente**, estos últimos como unidades de composición independientes de las aplicaciones.

POO

- Asimismo, tampoco tiene en cuenta los factores de mercadotecnia necesarios en un mundo real, como la distribución, adquisición e incorporación de componentes a los sistemas.

Agenda

- Introducción
- **Componentes**
- Arquitectura de Software
- Estructura de un componente.

Programación orientada a componentes (POC)

- Nace como una extensión natural de la **orientación a objetos** para los entornos abiertos.

- Este paradigma promueve el desarrollo y utilización de componentes reutilizables dentro de lo que sería un mercado global de software.

Programación orientada a componentes (POC)

- La **I**ngeniería de **S**oftware **B**asada en **C**omponentes (ISBC) se centra en el diseño y construcción de sistemas basados en computadoras que utilizan componentes de software reutilizables.

Componentes

- Para contar con un mercado de componentes de software es necesario que los componentes estén **empaquetados** de forma que permitan su distribución y composición con otros componentes, especialmente con aquellos desarrollados por terceras partes.

Definición de componentes

- Componente
  - Una parte remplazable, casi independiente y no trivial de un sistema que cumple una función clara en el contexto de una arquitectura bien definida.

Definición de componentes

Componente en ejecución

- Un paquete dinámico de unión de uno o más programas gestionados como una unidad, a los que se accede a través de interfaces documentadas.

- Las interfaces de un componente determinan tanto las operaciones que el componente implementa como las que precisa utilizar de otros componentes durante su ejecución.

Definición de componentes

Componente de software

- Una unidad de composición que sólo depende del contexto contractual de forma específica y explícita.

Definición de componentes

Componente de negocio

- La implementación de software de un concepto comercial “autónomo” o de un proceso comercial.

Definición de componentes

Szyperski, 2002

- "Un **componente** es una unidad de composición de aplicaciones de 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".

Definición de componentes

Definición WCOP'96

- Unidad de composición con interfaces especificadas contractualmente, con dependencias explícitas de acuerdo al contexto. Un componente de software puede ser desplegado de forma independiente y puede participar en composiciones de terceras partes.

Características

- **Identificable:** Debe tener una identificación que permita acceder fácilmente a sus servicios y que permita su clasificación.
- **Remplazable:** Se puede remplazar por nuevas versiones u otro componente que lo mejore.
- **Acceso a través de su interfaz:** Debe asegurar que estas no cambiaran a lo largo de su implementación.
- **Servicios no varían:** Las funcionalidades ofrecidas en su interfaz no deben variar, pero su implementación sí.

Características

- **Documentado:** Un componente debe estar correctamente documentado para facilitar su búsqueda si se quiere actualizar, integrar con otros, adaptarlo, etc.
- **Genérico:** Sus servicios deben servir para varias aplicaciones.
- **Reutilizado dinámicamente:** Puede ser cargado en tiempo de ejecución en una aplicación.
- **Independiente de la plataforma:** Hardware, Software, S.O.

Clases

- Las clases no hacen referencia explícita a sus dependencias y requisitos. Las clases suelen construirse mediante herencia de implementación, y por tanto no suelen permitir una instanciación e instalación por separado de la clase base y de sus clases hijas.

Módulos

- Un módulo es un **conjunto de clases**, opcionalmente junto a otros **elementos NO orientados a objetos**, como pueden ser procedimientos y funciones.

Paquete

- Un paquete es un conjunto de clases, usualmente **agrupadas conceptualmente**.

- Los paquetes no suelen ser ejecutables, y pueden ser consideradas como la versión orientada a objetos de las librerías tradicionales.

Recursos y Frameworks

- Un recurso es una colección no modificable de elementos de algún tipo [Szyperski, 1998].

- Frameworks o marcos de trabajo suelen estar **compuestos de componentes**, de los cuales unos están fijados por el propio marco, y otros son los que proporciona el usuario para especializar el marco de trabajo.

Modelo de componentes

- Define la forma de sus interfaces y los mecanismos para interconectarlos entre ellos.

- Determinan los mecanismos de composición
  - Ejemplos:
    - DCOM, COM, JavaBeans, CORBA

Plataforma de componentes

- Es un entorno de desarrollo y de ejecución de componentes que **permite aislar la mayor parte de las dificultades conceptuales y técnicas** que conlleva la construcción de aplicaciones basadas en los componentes de ese modelo.

- En este sentido, podemos definir una plataforma como una implementación de los mecanismos del modelo, junto con una serie de herramientas asociadas.

Composición de componentes

- Ingredientes arquitectónicos
  - Modelo de intercambios de datos (transferencia de datos)
  - Automatización (facilitar la interacción entre componentes)
  - Almacenamiento estructurado (organización en almacenamiento)
  - Modelo de objetos subyacente (asegura interoperabilidad entre los componentes)

Actividad I

1. Definir un diagrama de clases para un auto o casa inteligente (caja de fusibles, control de puertas, dispositivos domésticos, vehículos, semáforos, señalamientos).

2. Generar a partir del diagrama de clases un diagrama de componentes (¿Terminal?, ¿Banco?, ¿Cocina?)

3. Enviar diagramas por correo electrónico:

4. Asunto: TSCI_NombreApellidos_Actividad1