



UMSNH

INGENIERÍA DE PROGRAMACIÓN
"PROPUESTA DE PROYECTO FINAL"

Alumno:

Jonathan Daniel Sandoval Saucedo

Facultad de Ingeniería Eléctrica

Sección: 403

Propuesta de aplicación de la ingeniería del software basada en componentes en el desarrollo de software

Introducción

Los principios de reutilización de software han estado presentes a lo largo de muchos años, y para un ingeniero de software es conveniente contar con un conjunto de unidades software o aplicaciones modulares que se puedan ensamblar para formar un sistema nuevo más grande y complejo, en vez de tener que desarrollar el sistema nuevo desde cero.

A estas unidades software se les conoce como componentes. Al poder utilizar estos componentes software, que ya han sido probados y verificados se puede disminuir el tiempo de desarrollo y hacer sistemas informáticos más confiables y seguros. Cuando se requiera construir un nuevo sistema, se buscarían los componentes en el repositorio, se combinarían y adaptarían, en vez de codificar y construir las mismas aplicaciones cada vez. Las empresas desarrolladoras de software y la industria en general, deben aplicar este modelo de desarrollo basado en componentes para poder crear su propia biblioteca de componentes para hacer más fácil, seguro y rápido el desarrollo de nuevos sistemas informáticos.

El ensamblaje de componentes lleva a una reducción en el tiempo de desarrollo, una reducción en los costos del proyecto y un aumento en el índice de productividad. Aunque estos resultados están en función de la robustez de la biblioteca de componentes, no hay duda que el ensamblaje de componentes proporciona ventajas significativas para los ingenieros de software y las empresas en general. La ingeniería del software basada en componentes (ISBC) es un proceso que se centra en el diseño y construcción de sistemas basados en computadora que utilizan componentes de software reutilizables. La ISBC lucha por conseguir un conjunto de componentes de software preconstruidos y estandarizados que estén disponibles para encajar en un estilo arquitectónico específico para algún dominio de aplicación.

La aplicación se ensambla entonces con estos componentes y no las piezas por separado de un lenguaje de programación convencional. La ISBC, enmarca un conjunto de actividades necesarias para llevar a cabo la construcción de un sistema a partir de componentes reutilizables, en forma general estas actividades son:

- Ingeniería de Dominio: Consiste en la creación de la biblioteca de componentes. Establece el conjunto de componentes de software que el ingeniero del software puede reutilizar.
- Desarrollo basado en componentes:

Enmarca las actividades necesarias para la construcción del sistema informático a partir de componentes. La presente investigación se centró en la ingeniería de dominio, más específicamente en la forma de crear un componente software reutilizable que se puede usar en el desarrollo de muchas aplicaciones de software. En el presente artículo se pretende, en primer lugar, aclarar la definición de componente, se plantea un componente software reutilizable para ser implementado en java versión empresarial; en la sección IV se plantea el proceso de desarrollo de componentes; por último, se plantean los resultados y conclusiones del trabajo realizado.

Definición de componente de software

En esta sección se aclararán los conceptos y definiciones más importantes del término componente, y se especificarán los requisitos que una unidad software debe cumplir para considerarse como componente.

En primer lugar, el concepto de componente muchos autores lo definen de forma diferente: Según Szyperski, “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”. También dice que “Un componente es un módulo y una serie de recursos. Un módulo es una serie de clases, procedimientos y funciones. Un recurso es una colección congelada de elementos tipados”.

Meyer define en siete criterios que debe cumplir un elemento software para poder definirse como componente:

- 1) “Debe ser usado por otros elementos software.
- 2) Debe ser usado por clientes sin la intervención del desarrollo de componentes.
- 3) Incluir una especificación de todas las dependencias.
- 4) Incluir una especificación de las funcionalidades que ofrece.
- 5) Es usado solamente con base en su especificación.
- 6) Se puede componer con otros componentes para formar un software.
- 7) Puede ser integrado rápidamente y fácilmente”.

Después de ver las definiciones que diferentes autores dan al concepto de componente, se puede definir a un componente como una unidad software utilizada para ensamblar o componer un sistema más grande y complejo, un componente debe contener una especificación que permita identificarlo y reutilizarlo. Un elemento software para poder clasificarse como componente debe cumplir con los siguientes criterios:

- Debe ser una unidad software sin dependencia estructural de otras unidades.
- Debe tener un alto grado de cohesión.
- Debe tener una identificación, una descripción y una especificación de las funcionalidades que realiza.

- Debe seguir un modelo de componentes.
- Debe poder ensamblarse de forma rápida y fácil con otros componentes para formar un sistema más grande.
- Debe tener una interfaz que permita utilizar, modificar y adaptar las funcionalidades que maneja.
- Debe estar certificado y probado para asegurar que cumple con las funcionalidades para lo que fue implementado.
- Debe permitir el mantenimiento y la actualización de forma individual, sin afectar estructuralmente al sistema que compone.

Después de revisar el concepto de componente es importante mirar la definición de Modelo de Componentes, que es otro de los términos más utilizados en el ámbito del desarrollo de software basado en componentes.

Un modelo de componentes define la forma como se especifica un componente y la forma como se ensambla el componente, también incluye una serie de tipos de componente, sus interfaces y una especificación de los patrones aceptables de interacción entre tipos de componentes.

-Análisis de requisitos del área de aplicación

En esta fase se pretende establecer el dominio de aplicación y el modelo de requisitos para la biblioteca de componentes que se pretende crear. Para establecer el dominio de aplicación es importante identificar y delimitar las áreas posibles de aplicación, estas áreas se definen dependiendo de la necesidad que la empresa de desarrollo de software tenga para implementar componentes y su grado posible de reutilización por los desarrolladores de software. Después de seleccionar las áreas posibles de aplicación es importante evaluar los siguientes criterios (Grado de utilización, dificultad de implementación, complejidad, tolerancia al cambio, nivel de abstracción, necesidad de implementación y posibilidad de encontrar componentes ya desarrollados), para poder decidir el dominio de aplicación de la biblioteca de componentes. Para cada criterio se define una escala y se da un puntaje de evaluación. Esta evaluación la debe realizar el arquitecto software de la organización; para el caso, el jefe de la DSI. Los resultados de esta fase son la descripción del dominio de aplicación y el modelo de casos de uso.

-Modelo de selección de componentes

Con el modelo de selección de componentes se pretende elegir los componentes que se van a implementar para el dominio de aplicación establecido, de acuerdo con los casos de uso definidos en fase anterior. Los resultados de esta fase son los componentes software reutilizables candidatos para ser implementados.

-Definición de la arquitectura

Una vez definidos y seleccionados los componentes, es necesario definir el estilo arquitectónico que se va a utilizar en su implementación.

La arquitectura software de los componentes define la forma como se distribuyen e interrelacionan los elementos estructurales del componente y define el medio de comunicación entre ellos.

-Procedimiento de desarrollo, pruebas y especificación de los componentes

Con el procedimiento de desarrollo, pruebas y especificación se pretende definir los pasos necesarios para implementar los componentes software reutilizables, para garantizar su calidad y describir detalladamente las funcionalidades, propiedades y elementos del componente por medio de su especificación. En la Fig. 6 se puede apreciar el procedimiento propuesto.

-Distribución de los componentes

Los resultados de la implementación del componente son: archivos de tipo jar para las entidades y para la implementación de los servicios; ear para la implementación de los servicios, un archivo de configuración de la conexión con la base de datos, archivos jar para la implementación de los servicios Web y para la implementación de los componentes personalizados para la interfaz de usuario, una ficha de especificación y modelos del componente.

Para la distribución se debe crear un archivo por medio del cual se pueda instalar en el servidor de aplicaciones el componente de forma automática, luego empaquetar los archivos en un archivo .tar o .zip para ser colocado en el servidor de versiones de la organización. En el servidor de versiones de la empresa de desarrollo el programador encontrará un único archivo con toda la documentación y especificación del componente.