

Framework para la integración de herramientas de desarrollo basado en reutilización

Francisco Javier Pérez García¹ and Yania Crespo González-Carvajal¹

Departamento de Informática, Universidad de Valladolid, ,
{jperez, yania}@infor.uva.es, <http://giro.infor.uva.es>

Resumen Este trabajo presenta la definición, diseño y puesta en marcha de un entorno distribuido para el desarrollo de software basado en reutilización que permite la integración tanto de herramientas establecidas como de nuevas herramientas. Se ha definido una arquitectura genérica de colaboración entre entidades distribuidas, un protocolo de comunicación que les permite solicitar servicios e intercambiar objetos pertenecientes a un modelo de componente reutilizable, y se han establecido sus servicios básicos. Se ha utilizado un diseño basado en frameworks para permitir la reutilización de mecanismos y servicios, la independencia de la implementación y la tecnología particular de las entidades, la inclusión de nuevos servicios y funcionalidades, la utilización de diferentes modelos de componente reutilizable, etc.

1. Introducción

Entre los diferentes elementos que se requieren para la introducción de la reutilización sistemática de software en los procesos de desarrollo se necesita un soporte operativo. Para implantar un modelo de reutilización en entornos reales de desarrollo se debe contar con la infraestructura y las técnicas apropiadas, y éstas deben permitir la integración de herramientas. Por otra parte, los entornos de producción de software responden cada vez más a las características del mundo moderno en el que se impone la distribución de los recursos técnicos y humanos, la coexistencia de tecnologías heterogéneas, etc.

Este trabajo aborda la integración de herramientas de desarrollo de software basado en reutilización, en el que el elemento característico es la necesidad de intercambiar elementos y servicios relacionados con un modelo de reutilización. Se ofrece un soporte para la utilización de cualquier modelo de componente reutilizable independientemente de las herramientas utilizadas. Para ello, se define una arquitectura en la que se pueden integrar los diferentes tipos de herramientas que se pueden encontrar en un entorno de desarrollo basado en reutilización. Se define un protocolo de intercambio y se definen los servicios genéricos que deben ofrecer, de manera que se puedan establecer instancias predefinidas del framework para cada tipo de herramientas.

Los contenidos del trabajo se estructuran como sigue: En la sección 2 se presenta la arquitectura propuesta y los servicios que sus entidades pueden ofrecer. En la sección 3 se introduce el protocolo de comunicación. En la sección 4 se explica la forma en que el diseño permite la adaptabilidad y escalabilidad del entorno. En la sección 5 se describe la puesta en marcha de un ejemplo de entorno distribuido. Finalmente en la sección 6 se concluye y se describen las líneas de acción inmediatas.

2. Arquitectura del entorno

El entorno de desarrollo distribuido basado en reutilización, se compone de un conjunto de entidades genéricas, denominadas **Nodos**, capaces de intercambiar servicios y objetos pertenecientes al modelo de componente reutilizable empleado mediante un protocolo particular.

2.1. Tipos de Nodos y Servicios Predefinidos

A partir de las funciones y las tareas que necesitan realizarse en un entorno de desarrollo *para* y *con* reutilización [1,2], y de las herramientas necesarias para ello, se han establecido inicialmente cuatro tipos de nodos:

Tool Herramientas interactivas, utilizadas directamente por los usuarios. Se tratará de herramientas CASE con funcionalidades específicas para reutilización y para trabajar con un modelo de componente reutilizable concreto. Sólo se ha definido un servicio, que permite preguntar a un nodo Tool por información relativa a un usuario, como datos personales, datos de acceso a un repositorio, etc.

Broker Nodos encaminadores que actúan como distribuidores de mensajes entre los demás nodos. Realizan diversas funciones de gestión del protocolo. Entre las tareas más importantes de las que llevan a cabo, está la de validar los mensajes intercambiados para garantizar que sólo contienen objetos y modelos¹ pertenecientes al modelo de componente reutilizable empleado. Mantienen un registro de los nodos activos en un entorno, con información, enviada por los propios nodos, acerca de la ubicación y el estado de cada uno. Pueden también controlar de acceso a los servicios del entorno según el rol del usuario solicitante.

Processor Nodos específicos para realizar un servicio concreto sobre objetos y/o modelos y devolver el resultado. Interaccionan exclusivamente con otros nodos. Se instanciarán para ofrecer servicios específicos, que no se han predefinido inicialmente.

Repository Nodos del entorno de desarrollo que ofrecen servicios de repositorio referencial. Se han definido sólo los servicios básicos: consulta, recuperación, modificación y almacenamiento.

El entorno se ha diseñado para trabajar con modelos de componente reutilizable referenciales. Para poder integrar repositorios de contención y permitir el acceso transparente a componentes reutilizables almacenados en repositorios de este tipo, se ha planteado un quinto tipo de nodo, esbozando una propuesta que permita su desarrollo en trabajos posteriores.

FileBroker Nodos que ofrecen servicios de repositorio de contención local y de recuperación transparente de componentes reutilizables, ubicados en el repositorio del proveedor o en el del propio nodo.

Además, se han definido una serie de servicios comunes: todos los nodos identifican a sus usuarios mediante sesiones, y proporcionan información acerca de su estado, los servicios que ofrecen y las funciones que pueden ejecutar.

¹ Se dice objetos y modelos para referirse a objetos que representan un asset o conjuntos de objetos –assets– conectados según las relaciones del modelo de reutilización al que pertenecen

3. Protocolo de intercambio de objetos y servicios

La coexistencia de herramientas heterogéneas trabajando según un modelo de componente reutilizable y la necesidad de comunicación entre ellas, hace necesario el desarrollo de un protocolo común.

La definición del protocolo (formato de mensajes, conexiones, etc.) debe ser lo suficientemente general, para que pueda ser utilizado por el mayor conjunto posible de herramientas y aplicaciones. Por otra parte, ciertos aspectos del protocolo deberán ser específicos para poder recoger el modelo de componente reutilizable, de forma que, por ejemplo, se puedan realizar las comprobaciones sobre la corrección respecto al modelo de los elementos intercambiados.

Para permitir que los mensajes puedan procesarse con independencia de la tecnología utilizada en la implementación del protocolo, se ha optado por utilizar XML como lenguaje de codificación.

Las características más importantes que ofrece el formato diseñado para los mensajes son:

- Permite el intercambio de cualquier conjunto de objetos pertenecientes al modelo de componente reutilizable junto con las relaciones entre ellos.
- Permite que mediante una única petición y a partir de un mismo conjunto de datos de entrada, se puedan realizar varias operaciones simples (servicios), que constituyan conjuntamente una operación más compleja.
- Se delimitan y separan los elementos dependientes del modelo de reutilización.

El conjunto principal de datos de trabajo de los servicios es el modelo de componente reutilizable. Además de estos datos, los servicios pueden tomar como argumentos y generar como resultados, datos pertenecientes a tipos simples, concretamente a los tipos básicos definidos en el lenguaje Java [3].

Para permitir el intercambio de objetos pertenecientes al modelo de reutilización, se ha trasladado un modelo de componente reutilizable completo a XML, partiendo de la descripción semiformal del modelo en la forma de un diagrama de clases UML. El formato resultante, permite intercambiar conjuntos de objetos y las relaciones entre ellos, hace posible la ampliación posterior del modelo y puede ser adaptado para utilizar diferentes modelos de componente reutilizable

El modelo de componente reutilizable para el que se ha adaptado inicialmente el protocolo, es el modelo de reutilización de MECANO, del grupo GIRO de la Universidad de Valladolid [4].

Para la solicitud de servicios, se utiliza dentro de los mensajes un lenguaje sencillo que permite solicitar la ejecución combinada de varios de ellos. Una petición de servicios realizada mediante este lenguaje, está constituida por los identificadores de cada servicio, enlazados mediante operadores que expresan ejecución en secuencia, encadenamiento (composición de servicios) y operaciones condicionales. El protocolo permite también indicar el nodo en el que se desea ejecutar cada servicio, lo que ofrece la posibilidad por ejemplo, de encadenar servicios entre varios nodos.

Las posibilidades de adaptación y extensión del protocolo son varias. La más importante es, quizá, la posibilidad de extender el modelo de componente reutilizable o incluso de utilizar otros modelos. También existe la posibilidad de utilizar el protocolo

a través de diferentes medios de transmisión (conexiones) y la posibilidad de aplicar filtros de compresión y encriptación para mejorar el rendimiento y la seguridad del mismo.

4. Diseño y solución basado en frameworks

Para permitir la reutilización de mecanismos y servicios a la vez que la independencia del modelo de reutilización respecto a la implementación y la tecnología particular de cada nodo, se ha utilizado un diseño basado en frameworks [5].

Este diseño ofrece, por un lado, la escalabilidad de funciones y servicios en los nodos, permitiendo que un nodo incorpore nuevos servicios, y por otro lado, la adaptabilidad de los servicios, permitiendo que la funcionalidad independiente del modelo de reutilización pueda recaer en diferentes soluciones tecnológicas.

El framework implementa las funcionalidades básicas para trabajar con objetos del modelo de componente reutilizable y con el protocolo del entorno y las funciones necesarias para ejecutar y solicitar los diferentes servicios. Para ello se han diseñado de forma independiente:

- las funciones relacionadas con la comunicación
- las estructuras de datos que representan el formato de los mensajes
- las estructuras de datos que representan el modelo de componente reutilizable
- el núcleo de los nodos, que gestiona toda la funcionalidad
- la definición abstracta de las funciones que es capaz de ejecutar el nodo

Gracias a la adaptabilidad del nodo, se permite configurar qué funciones y servicios será capaz de ejecutar, a la vez que el diseño escalable permite que un desarrollador incluya nuevas funciones y servicios. Inicialmente, se han elaborado las configuraciones por defecto para instanciar los diferentes tipos de nodos con los servicios predefinidos correspondientes.

Los servicios predefinidos, se implementan con puntos de variabilidad que permiten añadir la parte dependiente de la implementación.

La ejecución de un servicio comprende diversas tareas que pueden agruparse en: las que dependen exclusivamente del protocolo y el modelo de reutilización (en particular del modelo de componente reutilizable), y las que dependen de determinadas soluciones tecnológicas que realizan las funciones de más bajo nivel. Se han desarrollado las primeras y se han definido como métodos abstractos las segundas, introduciendo la lógica que las une como métodos plantilla.

Por ejemplo, en el caso de un nodo Repository, las funciones de más bajo nivel son implementadas por un almacén de datos, que ofrecerá servicios de almacenamiento, consulta, modificación, etc. Este almacén de datos, puede ser cualquier solución comercial de bases de datos orientadas a objetos o relacionales, y la forma en que se soliciten las operaciones y se obtengan los resultados puede variar considerablemente de unas a otras. Incluso para almacenar componentes pertenecientes a un mismo modelo de componente reutilizable, pueden existir bases de datos con diferentes esquemas.

Por el contrario, existen una serie de funciones independientes del almacén de datos utilizado. En el servicio de consulta, estas funciones incluirán: la elaboración de

consultas en un lenguaje intermedio, al estilo de sql, la ejecución de dichas consultas utilizando la tecnología particular y la presentación de los resultados.

5. Puesta en marcha de un entorno distribuido para el desarrollo basado en reutilización

A lo largo de los últimos años en nuestro grupo hemos venido desarrollando trabajos que han pasado por la definición del Modelo de Reutilización GIRO (MRG) basado en la definición de un modelo de componente reutilizable denominado Mecano [4,6,7], de un proceso de desarrollo basado en reutilización [8], así como en complementar éstos con trabajos específicos en las fases de Ingeniería de Requisitos [9], diseño e implementación basado en frameworks [10] y en certificación de la calidad de componentes reutilizables [11]. Por otra parte, se había trabajado en el desarrollo y puesta en marcha de la biblioteca de reutilización GIRO [12].

Una vez definidos todos los aspectos de la propuesta de entorno genérico para el desarrollo basado en reutilización, se ha obtenido una primera versión de implementación de éste. A partir de esta versión, se realiza la puesta en marcha de una instancia específica de entorno distribuido para el desarrollo basado en reutilización según el modelo MRG. Esto tiene un doble propósito, hacer una primera validación del entorno genérico propuesto, a la vez que dotar al modelo MRG de un soporte operativo útil, que permitiera la integración de herramientas establecidas de desarrollo de software con las herramientas específicas que hemos desarrollado.

Constituyendo una configuración específica que cubre el soporte a las diferentes vistas del proceso de desarrollo, se ha realizado la instanciación del framework para herramientas propias: el entorno R^2 , una herramienta para asistir en la obtención de certificación de calidad, y herramientas que controlan el código implementado en Java y Eiffel desde el punto de vista de las relaciones definidas en el modelo de Mecano. Por otra parte, se destacan las instanciaciones realizadas para las herramientas CASE externas (altamente utilizadas) Together y Rational Rose, mediante sus mecanismos de extensibilidad.

Se han realizado pruebas internas, aunque aun no he han podido realizar pruebas de puesta en marcha en un entorno real de desarrollo.

6. Conclusiones y trabajo futuro

En este trabajo se ha presentado una propuesta de entorno distribuido como soporte al desarrollo basado en reutilización. Dicha propuesta consta de la definición de una arquitectura genérica, un protocolo y el diseño e implementación de un framework que permite, mediante su instanciación, la reutilización de mecanismos y servicios a la vez que la independencia de la implementación y la tecnología particular de los nodos específicos que se conecten. El modelo de componente reutilizable empleado es un modelo definido en nuestro grupo.

Se ha realizado la puesta en marcha de un entorno de pruebas mediante la adaptación de las herramientas ya existentes para integrarlas mediante el entorno de desarrollo,

algunas herramientas son propias y específicas para trabajar con el modelo de reutilización de nuestro grupo y otras son herramientas CASE establecidas que se han adaptado a través de sus mecanismos de extensibilidad.

Es deseable, y se plantea a llevar a cabo en el futuro, realizar pruebas de implantación real, dónde entre otros, se puedan realizar análisis de rendimiento y eficiencia, lo cual requeriría también de estudios empíricos y elaboración de experimentos sobre el impacto de su utilización en entornos reales de desarrollo.

Queda abierta también la posible evolución del entorno de desarrollo y explorar cómo se adapta la arquitectura y el protocolo a diferentes modelos de componente reutilizable.

Referencias

1. Girardi, M.R.: Main approaches to software classification and retrieval. En: Actas del Curso de Ingeniería del Software y Reutilización: aspectos dinámicos y generación automática. (1998)
2. Karlsson, E., ed.: Software reuse. A holistic approach. John Wiley & Sons (1995)
3. Arnold, K., Gosling, J.: The Java Programming Language. Java Series. Sun Microsystems (1996)
4. García, F.: Modelo de reutilización soportado por estructuras complejas de reutilización denominadas mecanos. PhD thesis, Departamento de Informática y Automática, Universidad de Salamanca (2000)
5. Fayad, M.E., Schmidt, D.C.: Building Application Frameworks: Object-Oriented Foundations of Framework Design (1999) John Wiley & Sons.
6. García, F.J., Marqués, J.M., Laguna, M.Á., Maudes, J.M.: Influencia de las relaciones entre elementos software reutilizables en la generación de mecanos. En Toval, A., Nicolás, J., eds.: Actas de Las III Jornadas En Ingeniería Del Software (JIS'98), Murcia, 11-13 de Noviembre de 1998 (1998) 155–166
7. García, F.J., Marqués, J.M., Maudes, J.M.: Mecano: Una propuesta de componente software reutilizable. En Díaz, O., Lopistéguy, P., eds.: Actas de Las II Jornadas de Ingeniería Del Software, Donostia-San Sebastián, España, 3-5 septiembre de 1997 (1997) 232–244
8. Laguna, M., González-Baixauli, B., López, O., García, F.: Introducing Systematic Reuse in Mainstream Software Process. En: Euromicro 2003 (accepted), Antalya, Turkey (2003)
9. López Villegas, O.: Reutilización de Requisitos para Incremento de la Calidad y Productividad en el Desarrollo de Especificaciones. PhD thesis, Departamento de Informática, Universidad de Valladolid (2003)
10. Prieto, F., Crespo, Y., Marqués, J., Laguna, M.: Mecanos y análisis de conceptos formales como soporte para la construcción de frameworks. En: Actas de las V Jornadas de Ingeniería de Software y Bases de Datos (JISBD'2000), 8-10 Noviembre de 2000 - Valladolid (España) (2000) 163–175
11. Manso, M.E., García, F.J., Romay, M.P., Marqués, J.M.: Modelo de cualificación y auditorías de elementos reutilizables de un repositorio. En Botella, P., Hernández, J., Saltor, F., eds.: Actas de Las IV Jornadas de Ingeniería Del Software Y Bases de Datos (JISDB'99), 24-26 Noviembre de 1999 - Cáceres (España) (1999) 355–366
12. Hernández, C., García, F., Laguna, M.: La biblioteca de reutilización giro. En: I Jornadas de Trabajo DOLMEN, 12-13 Junio de 2001- Sevilla (España) (2001) 102–112