

# Construcción de Refactorizaciones para Eclipse: Dynamic Refactoring Plugin 3.0

Raúl Marticorena<sup>1</sup>, Míryam Gómez<sup>1</sup>, Iñigo Mediavilla<sup>1</sup>, y Yania Crespo<sup>2</sup>

<sup>1</sup> Universidad de Burgos, EPS Edificio C, C/Francisco Vitoria s/n, Burgos, España,  
rmartico@ubu.es, mgs0110@alu.ubu.es, ims0011@alu.ubu.es,

<sup>2</sup> Universidad de Valladolid, Campus Miguel Delibes, Valladolid, España,  
yania@infor.uva.es

**Resumen** La amplia mayoría de entornos de desarrollo han incluido funcionalidad para refactorizar el software. Sin embargo, aún no ha sido objetivo fundamental de este tipo de herramientas proveer funcionalidad para la facilitar la construcción más dinámica de refactorizaciones por parte del usuario del entorno. En este trabajo se presenta una herramienta desarrollada como plugin para Eclipse IDE que facilita a los desarrolladores la ejecución y construcción dinámica de refactorizaciones. Adicionalmente la herramienta permite que estas refactorizaciones sean exportadas/importadas de un entorno Eclipse a otro. Así, se brinda la posibilidad al desarrollador de personalizar las refactorizaciones en función de las necesidades de su organización, y de las necesidades que surgen a lo largo de sus tareas de desarrollo y mantenimiento. Por otra parte, la herramienta construida también permite caracterizar las refactorizaciones de las que dispone y, en base a ello, ofrece un mecanismo de consultas de refactorizaciones disponibles según diferentes criterios.

**Palabras clave:** refactorización, plugin, Eclipse IDE, construcción, caracterización

## 1. Características Principales

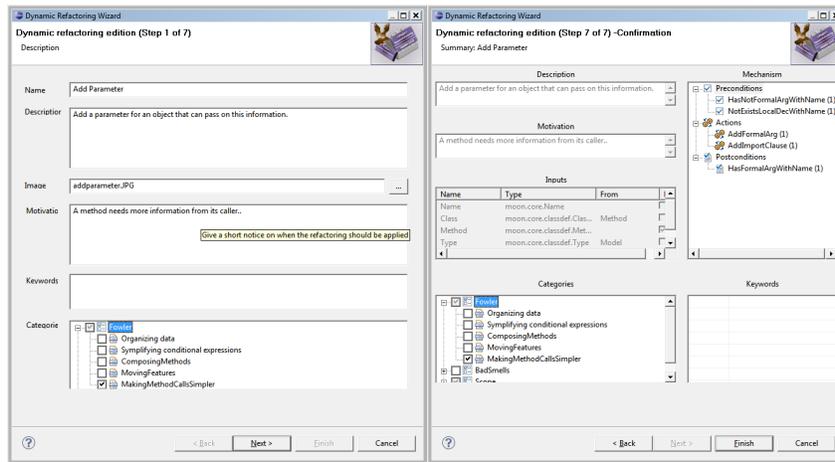
Para dar soporte a la representación del código, ejecución de las refactorizaciones y recuperación del código, se partió inicialmente de una notación minimal orientada a objetos, MOON [1]. A partir de dicha notación se desarrolló el metamodelo MOON [2,3] y la correspondiente extensión que soporta el lenguaje JAVA objeto de refactorización, denominada JAVAMOON.

Toda la información se almacena como instancias de un grafo del metamodelo JAVAMOON. El grafo puede ser recorrido para comprobar el cumplimiento de precondiciones/postcondiciones (consultas) y realizar las transformaciones (acciones) asociadas a la refactorización, transformando el estado del código. Al recorrer de nuevo el grafo, el código refactorizado puede ser regenerado.

La refactorización se define con la herramienta en un fichero XML que posteriormente es interpretado, cargando las clases necesarias desde los repositorios para la ejecución de la misma. Dicha labor la realiza el motor de refactorizaciones [4]. La interfaz gráfica de la refactorización se genera también en tiempo de ejecución.

## 1.1. Edición de Refactorizaciones

Se dispone en la zona superior de un menú, con las opciones de creación (**New**), edición (**Edit**) y borrado (**Delete**) de las refactorizaciones existentes. En la creación o edición se trabaja con un asistente (*wizard*) en siete pasos (ver Fig. 1 (a)-(b)) que permite construir la refactorización paso a paso.



(a) Paso 1 de 7

(b) Paso 7 de 7

Figura 1: Pasos inicial y final en el asistente en la creación y edición de refactorizaciones

Para incluir una nueva refactorización, o editar una existente, se sigue el siguiente proceso:

**Paso 1:** incluir información que caracterice a la refactorización como: nombre, objetivo, categorías, palabras claves, etc.

**Paso 2:** seleccionar las entradas de usuario indicando elementos del meta-modelo (de MOON o JAVAMOON).

**Paso 3:** seleccionar precondiciones indicando predicados de los repositorios (opcional).

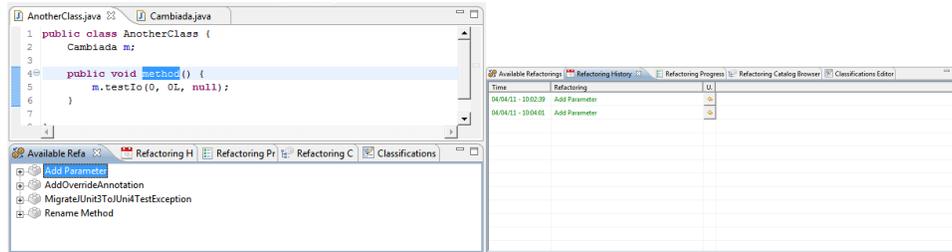
**Paso 4:** seleccionar acciones que transformen el código.

**Paso 5:** seleccionar postcondiciones indicando predicados de los repositorios (opcional).

**Paso 6:** incluir códigos de ejemplo (opcional).

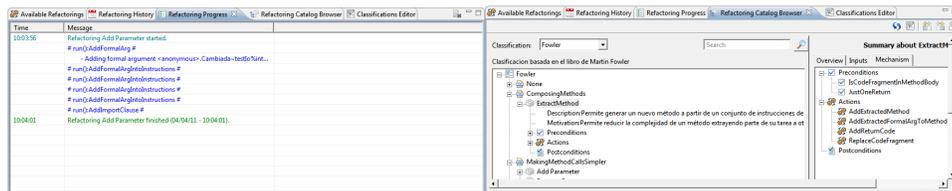
**Paso 7:** confirmar cambios.

El uso de un asistente o *wizard* evita errores manuales y la existencia de los elementos del repositorio se revisan por reflexión.



(a) Refactorizaciones disponibles

(b) Histórico de Refactorizaciones



(c) Progreso de la Refactorización

(d) Visor del Catálogo de Refactorizaciones

Figura 2: Vistas incluidas en el Plugin.

## 1.2. Vistas Incorporadas y Opciones Adicionales

- **Vista de Refactorizaciones Disponibles (*Available Refactorings*).**  
 En base al elemento seleccionado en el editor de código, esta vista ayuda al usuario indicando sólo aquellas refactorizaciones que son aplicables sobre el elemento actual (ver Fig. 2 (a)) y permitiendo su selección para ejecutarlas.
- **Vista de Histórico de Refactorizaciones (*Refactoring History*).** Permite la visualización del histórico de refactorizaciones aplicadas así como la aplicación de operaciones de “deshacer” (ver Fig. 2 (b)).
- **Vista de Progreso de la Refactorización (*Refactoring Progress*).** Se visualiza el progreso de la ejecución de las refactorizaciones (ver Fig. 2 (c)).
- **Vista de Catálogo de Refactorizaciones (*Refactoring Catalog Browser*).** Permite la visualización y consulta avanzada, incluyendo filtros, de las refactorizaciones del catálogo (ver Fig. 2 (d)).
- **Editor de Clasificaciones (*Classification Editor*).** Permite la edición de las clasificaciones, categorías y palabras claves en las que se puede catalogar una refactorización.

Este plugin también permite la importación/exportación de las refactorizaciones entre distintas instalaciones de ECLIPSE así como la importación/exportación de un conjunto de refactorizaciones aplicadas como un *Plan de Refactorizaciones* [5].

## 2. Conclusiones y Líneas de Trabajo Futuro

La herramienta se encuentra en su tercera iteración y su funcionalidad se sigue extendiendo y ampliando en la actualidad, contando con una base firme a partir de los trabajos previos (ver Agradecimientos). Sin tratarse todavía de una herramienta de producción, ofrece posibilidades al programador que usa habitualmente las refactorizaciones y sirve como demostración de lo planteado en trabajos teóricos previos [3, 4, 6, 7]. Aún con ciertas limitaciones de rendimiento a mejorar, el soporte del lenguaje JAVA es completo, incorporando los últimos elementos incorporados en la gramática (anotaciones, enumeraciones, genéricos, bucles foreach, varargs, etc.)

## Agradecimientos

La versión actual es una evolución de los trabajos previos *Plugin de Refactorización para ECLIPSE*, realizado por Sonia Fuente de la Fuente y Enrique Herrero Paredes, junio 2008, y su posterior revisión en el trabajo *Plugin de Refactorización para ECLIPSE. Dynamic Refactoring 2.0*, realizado por Laura Fuente de la Fuente, junio 2009. Ambos desarrollados en la Universidad de Burgos.

## Referencias

1. Yania Crespo. *Incremento del Potencial de Reutilización del Software mediante Refactorizaciones*. PhD thesis, Universidad de Valladolid, 2000. Available at <http://giro.infor.uva.es/docpub/crespo-phd.ps>.
2. Carlos López and Yania Crespo. Definición de un Soporte Estructural para Abordar el Problema de la Independencia del Lenguaje en la Definición de Refactorizaciones. Technical Report DI-2003-03, Departamento de Informática. Universidad de Valladolid, septiembre 2003. Available at <http://giro.infor.uva.es/docpub/lopeznzal-tr2003-03.pdf>.
3. Yania Crespo, Carlos López, and Raúl Marticorena. Un Framework para la reutilización de la definición de refactorizaciones. In *Actas JISBD'04, IX Jornadas de Ingeniería del Software y Bases de Datos, Málaga, Spain, ISBN 84-688-89830*, November 2004.
4. Raúl Marticorena and Yania Crespo. Dynamism in Refactoring Construction and Evolution. A Solution Based on XML and Reflection. In *3rd International Conference on Software and Data Technologies (ICSOFT)*, pages 214 – 219, July 2008.
5. Javier Pérez and Yania Crespo. Perspectives on automated correction of bad smells, 2009.
6. Carlos López, Raúl Marticorena, and Yania Crespo. Caracterización de Refactorizaciones para la Implementación en Herramientas. In *Actas JISBD'06, XI Jornadas de Ingeniería del Software y Bases de Datos, Sitges, Barcelona, 2006. ISBN:84-95999-99-4*, pages 538–543, oct 2006.
7. Carlos López, Raúl Marticorena, Yania Crespo, and Javier Pérez. Towards a language independent refactoring framework. In *1st ICSOFT 06 International Conference on Software and Data Technologies. Setubal, Portugal. ISBN: 972-8865-69-4*, volume 1, pages 165–170, sep 2006.