

PERFILES UML PARA LA DEFINICION DE PATRONES DE DISEÑO DE COMPORTAMIENTO

1. **A. Cortez, A. Garis, D. Riesco.**

2. Consejo de Investigaciones (CIUDA) – UDA.

3. Instituto de Investigaciones – Facultad de Ciencias Administrativas y Sociales. UDA.

1. **Departamento de Informática - Facultad de Ciencias Físico-Matemáticas y Naturales. UNSL.**

4. cortezalberto@gmail.com, {agaris,driesco}@unsl.edu.ar

Resumen

Los patrones de diseño como herramienta de la ingeniería de software, brindan un importante aporte. El estudio de los patrones de comportamiento (según la clasificación Gof) implica conocer la interacción entre los objetos y sus responsabilidades. El presente trabajo propone el uso de de perfiles UML y restricciones OCL para la definición de patrones de comportamiento. Dicho enfoque habilita la especificación y validación de patrones tanto en modelos estáticos como dinámicos.

Palabras clave: patrones de comportamiento, perfiles UML, especificaciones.

Contexto

La línea de investigación planteada se desarrolla en el marco del proyecto 22/F822

"Ingeniería de Software: Conceptos, Métodos y Herramientas en un contexto de Ingeniería de Software en Evolución", Facultad de

Ciencias Físico-Matemáticas y Naturales. UNSL. (Universidad Nacional de San Luis).

Introducción

El lenguaje de modelado UML es un estándar utilizado para especificar y documentar sistemas y particularmente para especificar patrones de diseño. Siendo necesario lograr una especificación precisa, el presente trabajo propone emplear perfiles UML como un mecanismo eficiente para lograr este objetivo. Los perfiles UML [8,9,10]. son proporcionadas por el propio UML para extender su sintaxis y su semántica, de manera de expresar los conceptos específicos de un determinado dominio de aplicación. La idea es logra una definición tipo de los patrones de diseño de comportamiento y poder aplicarlo en los modelos de sistemas. Para poder aplicar estos conceptos en el modelo los perfiles deben tener como base de descripción y aplicación: los diagramas de

clase y los diagramas de secuencia. Los primeros permiten reflejar las características estructurales y los segundos las características de comportamiento. De esta manera se pueden validar las especificaciones expresadas en los perfiles UML sobre un determinado conjunto de diagramas de un modelo.

El mecanismo aplicado para la especificación consiste de un perfil UML por cada patrón de comportamiento que contiene restricciones definidas en OCL. [1,4,6,7,9,10,11]. La implementación se llevó a cabo utilizando el concepto de arquitectura definido por Garis [2,5]. Utilizar una arquitectura para la implementación otorga un mayor uso de la reutilización. Se lleva a la práctica en este trabajo, la estructura propuesta de tres niveles: nivel de patrón general, nivel de comportamiento y nivel del patrón particular; definiendo los perfiles UML en una herramienta en particular. El enfoque permite no solo la especificación de elementos estructurales, sino también de comportamiento; habilitando su utilización tanto en diagramas de clases como en diagramas de secuencia. Adicionalmente, el mecanismo posibilita la verificación de consistencia y la existencia de ciertas interacciones en los modelos.

Líneas de investigación y desarrollo

En este trabajo se plantean como líneas de investigación:

- Lenguajes para la formalización de patrones de comportamiento.
- Metodología para la verificación de patrones de diseño en modelos.
- Herramientas para la especificación y verificación de patrones de diseño.

Para la formalización de patrones se investigaron diferentes trabajos de formalización y se seleccionó como adecuados los perfiles UML y el lenguaje OCL. Respecto de las herramientas investigadas, a los fines de cumplir el objetivo de simplicidad y versatilidad se escogió la herramienta RSA (Rational Software Developer) de IBM [12,13,14,15]. Se pretende determinar si es posible el uso integral de diagramas UML para la validación del modelo en forma sistémica. En este caso era importante poder integrar los diagramas ya que se trata de verificar tanto las características estructurales como las de comportamiento del modelo para su verificación.

Resultados y Objetivos

El presente trabajo tiene como objetivo lograr la especificación de todos los patrones de diseño clasificados como de comportamiento según Gof [3]. Como resultado de la investigación bibliográfica se obtuvo una

definición clara de cada patrón. Se implementó una arquitectura de patrones en el software RSA. La arquitectura contiene:

1. Un perfil con los elementos necesarios para definir la estructura de un patrón tipo definido por estereotipos con restricciones a ser verificadas en el diagrama de clases de un modelo.
2. Un perfil con estereotipos y restricciones que valida la consistencias entre el

diagrama de clase y de secuencia de un modelo. Y verifica la existencia de ciertas interacciones en el diagrama de secuencia para un patrón dado.

3. Un perfil para un patrón de comportamiento particular al que le han sido aplicados los perfiles definidos en los puntos anteriores y además contiene algunas restricciones propias del patrón.

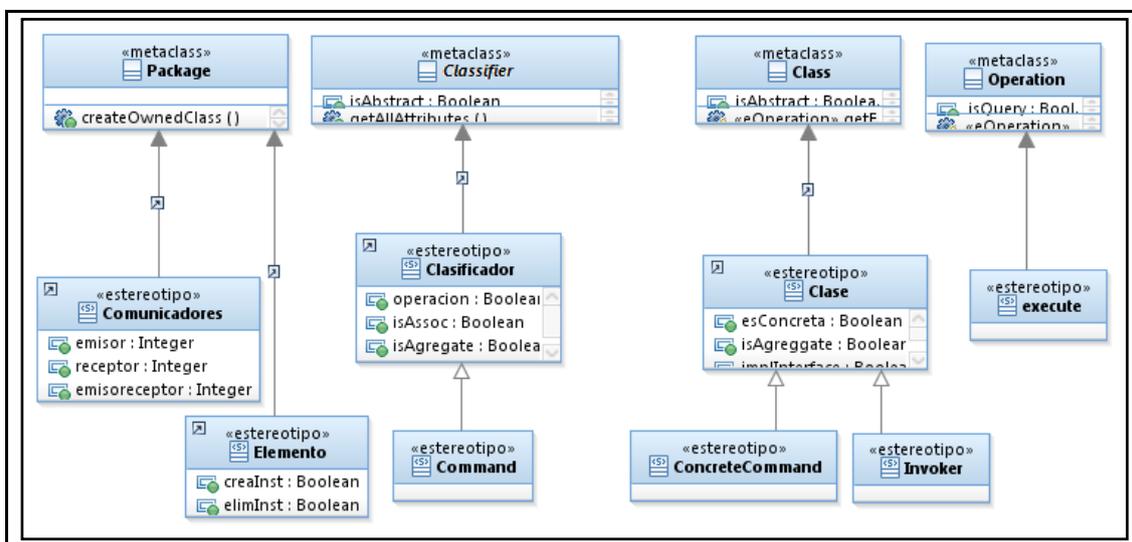


Figura 1 Perfil del patrón Command

En la figura 1 se observan los perfiles mencionados (punto 1 y 2) aplicados al perfil del patrón Command. Las metaclasses Classifier ,Class y Package con sus respectivos estereotipos. Se muestra un ejemplo donde se aplicaron los estereotipos : Command , ConcreteCommand , Invoker y execute. El ejemplo es el caso de un socket (enchufe), que comunica los programas del cliente y del servidor en una red. En este caso

se devuelve la nota de los estudiantes. Se reciben mensajes que contienen números y se envían mensajes con datos de los estudiantes. Se debe implementar el código de la conexión, eliminando del objeto DAO (usado para la conexión de Acceso a Datos) la responsabilidad del manejo de conexiones. Como el código de manejo de la conexión no se reusa , el recurso es factorizarlo a través del patrón de diseño Command.

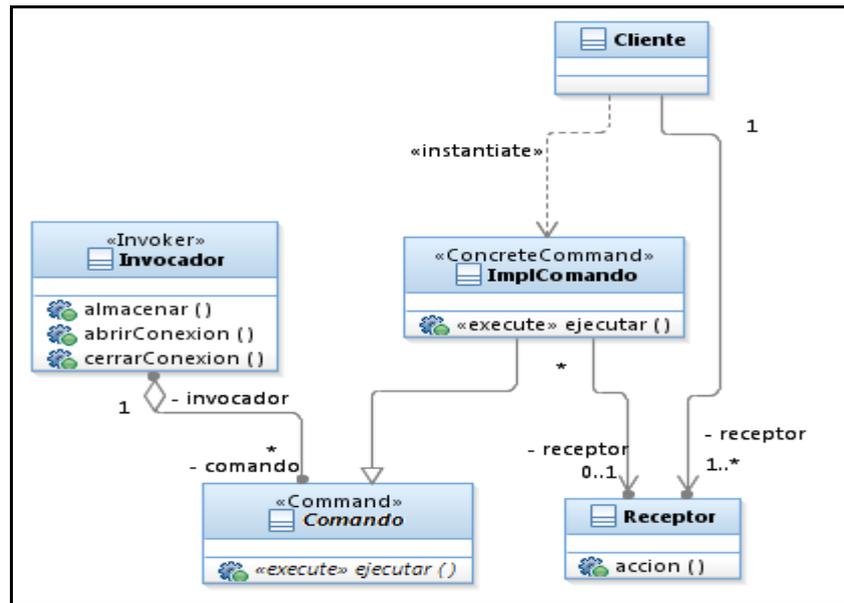


Fig. 2. Diagrama de clases usando perfiles

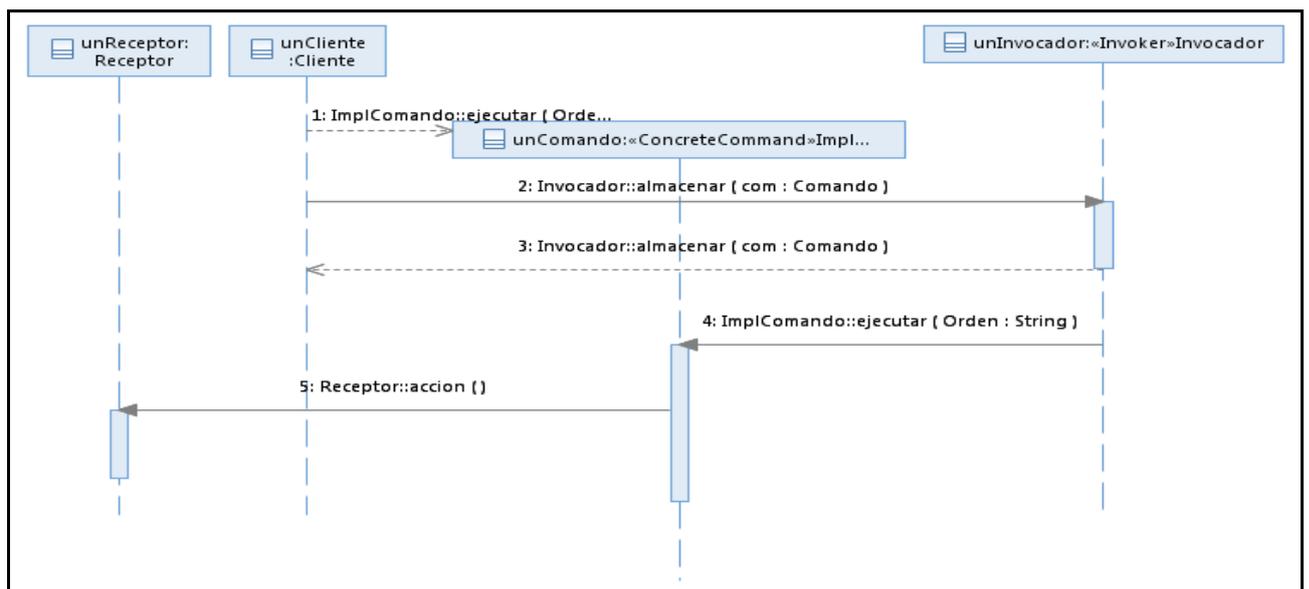


Fig. 3. Diagrama de Secuencia : Creación y Ejecución de Comando.

Se puede agregar a los perfiles definidos perfiles de la clasificación estructural por ejemplo, ya que el framework es una estructura sencilla y flexible. Un resultado del trabajo es la definición de una metodología para verificar las interacciones entre objetos. Se espera el mejorar las especificaciones definidas e incentivar la

investigación en aspectos asociados que sirvan como base para desarrollos futuros.

Se ha logrado obtener:

- Un modelo para la detección de los elementos de un diagrama coincidentes con un patrón. La propuesta aquí presentada, lo logra aplicando el perfil de un determinado patrón a un modelo determinado.

- Un framework para validar un modelo asociado a un patrón.

Uno de los propósitos del proyecto es la mejora en el desarrollo de modelos, mediante la detección de necesidades y la ampliación del campo de soluciones.

Formación de Recursos Humanos

En el marco de este proyecto se está desarrollando una tesis de maestría en la Facultad de Ciencias Físico-Matemáticas y Naturales de la Universidad de San Luis.

Este trabajo también se vincula al Consejo de Investigaciones y al Instituto de Investigaciones de la Universidad del Aconcagua aportando conocimiento a las líneas de investigación “Herramientas de generación de código automáticas” y “Arquitectura Dirigida por Modelos (MDA)”

Referencias

- [1] L. Fuentes, A. Vallecillo y J.M. Troya. 2002. Using UML Profiles for Documenting Web-Based Application Frameworks. *Annals of Software Engineering*, 13:249–264, 2002.
- [2] Debnath N., Garis A., Riesco D., Montejano G. 2006. Defining Patterns using UML Profiles. *ACS/IEEE International Conference on Computer Systems and Applications*, IEEE Press, www.ieee.org, pp.1147-1150.
- [3] Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John 1994. *Design Patterns: Elements of Reusable software*, Addison-Wesley.
- [4] García Carlos Diego. 2009. Tesis "Implementación de técnicas de evaluación y refinamiento para OCL 2.0 sobre múltiples lenguajes basados en MOF".
- [5] Garis, Ana. Perfiles UML para la definición de Patrones de Diseño 2007. Tesis de maestría. Universidad Nacional de San Luis (UNSL).
- [6] Gogolla, Martin, Bohling, Jorn and Richters, Mark 2003. Validation of UML and OCL Models by Automatic Snapshot Generation". Proc. 6th Int. Conf. Unified Modeling Language (UML'2003). Springer, Berlin, pages 265-279.
- [7] MDA 2007. Model-Driven Architecture. Disponible en www.omg.org/mda.
- [8] Catalog of UML Profile Specifications. 2008. http://www.omg.org/technology/documents/profile_catalog.htm.
- [9] Martínez, Liliana 2008. Componentes MDA para patrones de diseño. Tesis de Maestría. Universidad Nacional de La Plata.
- [10] OMG Superstructure 2010. Standard document URL: <http://www.omg.org/spec/UML/2.3/Superstructure>
- [11] UML Profile for Patterns Specification 2010. Disponible en <http://www.omg.org/spec/UML/2.3/>.
- [12] Swithinbank, Peter. Chessell, Mandy. Dr. Gardner, Tracy. Griffin, Catherine. Man, Jessica. Wylie, Helen. Yusuf, Larry. 2005. *Patterns: Model-Driven Development Using IBM Rational Software Architect*
- [13] Fernandez Saez, Pedro Antonio. Un análisis crítico sobre la aproximación Model-Driven Architecture. 2009. Tesis de maestría en sistemas inteligentes.
- [14] Qué hay de Nuevo en IBM Rational Software Architect 8.0 <http://www.ibm.com/developerworks/ssa/rational/library/10/whats-new-in-rational-software-architect-8/>
- [15] MDT-UML2-Tool-Compatibility <http://wiki.eclipse.org/MDT-UML2-Tool-Compatibility>.