

Jesús J. García Molina¹, Ana Moreira², Gustavo Rossi³

¹ Depto. de Informática y Sistemas, Universidad de Murcia; ² Depto. de Informática, Facultad de Ciencias y Tecnología, Universidade Nova de Lisboa (Portugal); ³ Facultad de Informática, Universidad Nacional de La Plata (Argentina)

<jmolina@um.es>,
<amm@di.fct.unl.pt>,
<gustavo@sol.info.unlp.edu.ar>

1. Introducción a UML

Tras la aceptación del paradigma orientado a objetos (OO) como el más adecuado para producir software de calidad, a principios de los noventa emergieron un buen número de métodos de desarrollo de software OO. En julio de 1993, Jacobson criticó en [1] lo que él denominaba *guerra de métodos* y planteó la necesidad de llegar a una notación estándar de modelado, para evitar la confusión reinante y favorecer el uso de los métodos de software OO. A finales de 1994 se inició un esfuerzo de unificación por parte de los creadores de los tres principales métodos: *Booch*, *Rumbaugh* y *Jacobson*. El Lenguaje Unificado de Modelado (UML, *Unified Modeling Language*) es el resultado de esa colaboración y de las aportaciones de las principales empresas de software.

UML fue adoptado en noviembre de 1997 por OMG (*Object Management Group*) como una de sus especificaciones y desde entonces se ha convertido en un estándar *de facto* para visualizar, especificar y documentar los modelos que se crean durante la aplicación de un proceso software. UML ha ejercido un gran impacto en la comunidad software, tanto a nivel de desarrollo como de investigación.

Su éxito ha sido enorme, como lo prueban, por una parte, su utilización en todo el mundo para construir aplicaciones en todos los dominios y de todos los tamaños, y, por otra, que los entornos de desarrollo más extendidos --como son los de Borland, Microsoft e IBM-- integran herramientas para el modelado con UML. Otras dos especificaciones de OMG relacionadas con UML son el lenguaje OCL (*Object Constraint Language*) y XMI (*XML Metadata Interchange*). OCL es un lenguaje que se utiliza para escribir expresiones sobre modelos, de modo que extiende la potencia expresiva de

Nota del Editor de Novática: por razones de espacio no se incluyen en esta monografía los artículos "*Naturaleza de las relaciones entre actores y casos de uso*", de **Gonzalo Génova Fúster** y **Juan Llorens Morillo**, y "*Métricas para Modelos UML*", de **Marcela Genero**, **Mario Piattini Velthuis**, **José Antonio Cruz-Lemus** y **Luis Reynoso**, que fueron seleccionados por los editores invitados.

Estos artículos han sido publicados en el número 2/2004 de **Upgrade**, <<http://www.upgrade-cepis.org>>, en inglés, y serán publicados en próximos números de **Novática**, en castellano.

Presentación

UML: el lenguaje estándar para el modelado de software

UML y permite crear modelos más precisos y más completos; XMI es un formato para intercambio de modelos basado en XML (*eXtensible Markup Language*).

2. Áreas de trabajo en torno a UML

Los principales objetivos en el diseño de UML fueron éstos: obtener un lenguaje simple pero suficientemente expresivo, que permitiese modelar aplicaciones en cualquier dominio; obtener un lenguaje legible, puesto que sería un lenguaje utilizado por las personas; y permitir la generación automática de código.

Para combinar la simplicidad con la aplicabilidad a cualquier dominio, UML incorpora un conjunto de mecanismos de extensibilidad que permiten definir perfiles (*profiles*) que lo adaptan a un dominio concreto (aplicaciones web, CORBA --*Common Object Request Broker Architecture*-- , modelado del negocio, EJB --*Enterprise Java Bean(s)*-- , ...). La legibilidad se obtiene mediante diagramas visuales que representan los modelos. Por último, la generación de código por parte de herramientas exige una definición formal de UML, que se consigue mediante un *metamodelo* expresado en un metalenguaje denominado MOF (*Meta-Object Facility*). La definición de perfiles y los trabajos sobre formalización de UML han sido dos áreas en las que se ha investigado intensamente.

El Proceso Unificado (*Unified Process*, UP) es un marco para definir procesos software basados en UML definido en Rational, la empresa de sus creadores que fue adquirida por IBM a finales de 2002. En los últimos años se han definido numerosos procesos que se ajustan a los principios del UP: procesos dirigidos por casos de uso, iterativos e incrementales, y centrados en la arquitectura. El proceso más extendido ha sido el RUP (*Rational Unified Process*), definido en Rational.

En estos años, el área de la definición de procesos y técnicas asociadas, como modelado de negocio o casos de uso, ha recibido mucha atención por parte tanto de la industria como del mundo académico. Sin duda, los *casos de uso* ha sido la técnica de modelado que ha gozado de mayor aceptación. Sin embargo, como señala *Booch* [2], todavía sólo un pequeño porcentaje de los desarrolladores utilizan UML y muchos proyectos lo usan tan sólo para documentar sus decisiones de diseño en vez de utilizar los modelos para razonar sobre el sistema, lo que supone su principal beneficio.

La aparición de UML ha supuesto el reconocimiento de la actividad del modelado (diseño

basado en modelos) como una actividad clave para producir software de calidad. Sin embargo, resulta paradójico que al mismo tiempo haya surgido el movimiento del *desarrollo ágil de software* (*Agile Software*), que considera que el valor está en el código y en las personas más que en los modelos y los procesos [3], así como la llamada *programación extrema* (*eXtreme Programming*, XP) [4].

Con la iniciativa MDA (*Model Driven Architecture*), promovida por el OMG, ha surgido un nuevo paradigma de desarrollo, no centrado en el código sino en los modelos, que se denomina *desarrollo guiado por modelos*. Con esta nueva visión, se puede obtener código a partir de modelos centrados en el dominio del problema e independientes de cualquier aspecto relacionado con la plataforma, mediante transformaciones de modelos.

Editores invitados

Jesús J. García Molina es Doctor en Ciencias por la Universidad de Murcia desde 1987. Actualmente es profesor del Depto. de Informática y Sistemas de esa misma universidad y decano de la Facultad de Informática. Ha sido pionero en España en la tecnología de objetos. Desde 1992 su docencia e investigación se han centrado en temas relacionados con la orientación a objetos. En la actualidad su investigación se centra en la ingeniería de modelos y en la especificación de componentes y arquitecturas orientada a la reutilización. Es coordinador de la Sección "Tecnología de Objetos" de **Novática** junto con Gustavo Rossi.

Ana Moreira es Doctora en Métodos Formales por la Universidad de Stirling, Escocia (Gran Bretaña), 1994. Tiene el Master en Investigación por la Facultad de Ciencia y Tecnología, Universidade Nova de Lisboa (Portugal), 1991. Desde 1995 es Profesora Asistente del Depto. de Informática de la Facultad de Ciencias y Tecnología de dicha universidad, donde enseña "Temas Avanzados de Ingeniería de Software". Es investigadora del Centro de Informática y Tecnología de la Información (CITI). Ha publicado docenas de artículos en revistas y congresos relacionadas con la Ingeniería de Software y los métodos formales. Sus áreas de interés incluyen Ingeniería de Requisitos y Desarrollo de Software Orientado a Aspectos, en particular aplicado a modelos de negocios e ingeniería de requisitos.

Gustavo Rossi es Doctor en Informática por la PUC-Rio (Brasil). Es Profesor Titular de la Facultad de Informática de la Universidad Nacional de La Plata (Argentina). Se dedica a la Orientación a Objetos desde fines de los 80 y ha dictado decenas de cursos sobre este tema. Su interés de investigación son las aplicaciones Web adaptables y en particular patrones y *frameworks* en estos contextos. Es coordinador de la Sección "Tecnología de Objetos" de **Novática** junto con Jesús J. García Molina.

La *Ingeniería de Modelos* es un área que cada vez cobra más importancia y que no se reduce al ámbito académico sino que está recibiendo atención por parte de las principales empresas de desarrollo de software, y ya existen IDEs (*Integrated Development Environments*) que incorporan herramientas de transformación de modelos basadas en MDA.

El objetivo sería que las herramientas generasen tanto código como fuese posible a partir de modelos, en la línea que se ha seguido a lo largo de la historia de la programación de liberar al programador de escribir todo aquel código que pueda ser generado por herramientas.

Con la importancia que cobran los modelos y la ingeniería de modelos, como es lógico, hay áreas de investigación que han adquirido especial importancia como son la calidad y evaluación de los modelos, la definición de perfiles, la formalización de las transformaciones de modelos o la creación de herramientas de transformación.

Desde la primera versión adoptada por OMG, UML ha evolucionado a través de diferentes versiones. Pero como ha señalado *Booch* en [2], el crecimiento ha significado mejorar su expresividad y precisión, pero no ha sufrido cambios en sus aspectos esenciales. El pasado octubre se adoptó el estándar UML 2.0, uno de cuyos objetivos principales es proporcionar una sólida base para MDA, proporcionando un mecanismo que facilita la definición de perfiles. Asimismo, incorpora soporte para el desarrollo basado en componentes, se mejora el modelado de las relaciones y las máquinas de estado a través de definiciones más rigurosas y precisas, y también incorpora mejoras relacionadas con los diagramas de secuencias y con el modelado del comportamiento en general.

3. Contenido de la monografía

En esta monografía se recogen una serie de artículos que reflejan el estado actual de UML y sirven para ilustrar diferentes aspectos tales como: la formalización de UML mediante el

metamodelo, OCL, la definición de perfiles, la ingeniería de modelos y el MDA y calidad de los modelos.

Como hemos comentado, los perfiles constituyen el mecanismo que incorpora UML para adaptarlo a un dominio concreto. *L. Fuentes Fernández* y *A. Vallecillo Moreno*, en su trabajo "*Una introducción a los perfiles UML*" nos muestran el conjunto de pasos para crear un perfil y argumentan la importancia que jugarán en MDA. Por otra parte, en "*Diseño Orientado a Aspectos con Theme/UML*", *Siobhán Clarke* presenta el *profile* Theme definido para el modelado orientado a aspectos, una técnica de desarrollo que cada vez cobra más importancia.

Jean Bézivin, un pionero en el terreno de la ingeniería de modelos y buen conocedor de MDA, en su artículo "*En búsqueda de un principio básico para la Ingeniería Guiada por Modelos*" nos ofrece una interesante reflexión sobre cómo identificar las características esenciales de este nuevo paradigma de desarrollo de software. En este mismo sentido, *Morgan Björkander* presenta, en "*Desarrollo basado en modelos y UML 2.0: ¿el final de la programación tal como la conocemos?*", un análisis del significado de MDA y de la Ingeniería de Modelos, y muestra cómo UML 2.0 incluye nuevas características que dan soporte a MDA.

Una presentación de la nueva versión de OCL es realizada por *Heinrich Hussmann y Steffen Zschaler* en su artículo "*El lenguaje de OCL para UML: síntesis y evaluación*".

Jan Jürjens, en su artículo "*Desarrollo de aplicaciones críticas en seguridad con UMLsec: una breve introducción*", analiza los problemas en la creación de sistemas críticos de alta calidad y muestra cómo puede ayudar el modelado con UML, presentando una herramienta que da soporte al enfoque propuesto.

Mariela I. Cortés, Marcus Fontoura y Carlos J.P de Lucena, en su trabajo, "*Utilización de re-factorización y reglas de unificación como*

soporte a la evolución de frameworks" utilizan UML-F, un perfil definido para el diseño de *frameworks*, para mostrar dos técnicas que ayudan en el mantenimiento y evolución de *frameworks*.

Agradecemos a todos ellos su valiosa colaboración, y a los editores de *Novática* y *Upgrade* la oportunidad de haber podido elaborar esta monografía, que esperamos sea de interés para los lectores de ambas revistas.

Referencias

[1] **I. Jacobson**. "Time for a cease-fire in the methods war", *Journal Object-Oriented Programming*, julio 1993

[2] **G. Booch**. "Growing the UML", *Software and System Modeling*, 1 (2), pp. 157-160, diciembre, 2002.

[3] **The Agile Alliance**: <<http://www.agilealliance.org/home>>.

[4] **eXtreme Programming / Programación eXtrema**, monografía de *Novática*, nº 156, marzo-abril 2002, <<http://www.ati.es/novatica/2002/156/nv156sum.html>>.

Referencias útiles sobre UML

Con objeto de que los lectores que lo deseen puedan obtener un conocimiento más amplio del asunto objeto de esta monografía, ofrecemos una *lista no exhaustiva* de recursos sobre el mismo.

Sitios web

- 18,193 Links on Objects & Components: <http://www.cetus-links.org/oo_uml.html>.
- IBM, UML Resource Center: <<http://www.ibm.com/software/rational/uml/>>.
- OMG (*Object Management Group*): <<http://www.omg.org/uml/>>.
- TECHLINQ, Links for Developers: <<http://home.wanadoo.nl/techlinq/uml.html>>.

Libros

- **G. Booch, J. Rumbaugh e I. Jacobson**. *El Lenguaje Unificado de Modelado*, Addison-Wesley, 1999.

- **A. Cockburn**. *Writing effective uses case*, Addison-Wesley, 2001.
- **P. Evtits**. *A UML Pattern Language*, Macmillan Technology Series, 2000.
- **M. Fowler, K. Scott**. *UML Distilled*, 2ª edición, 2000.
- **I. Jacobson, G. Booch, J. Rumbaugh**. *El Proceso Unificado de Desarrollo de Software*, Addison-Wesley, 1999.
- **C. Larman**. *UML y Patrones: Una introducción al análisis y diseño orientado a objetos y al Proceso Unificado*, Prentice-Hall, 2002.
- **J. Rumbaugh, I. Jacobson, G. Booch**. *El Lenguaje Unificado de Modelado. Manual de Referencia*, Addison-Wesley, 1999.
- **P. Stevens**. *Utilización de UML*, Addison-Wesley, 2002.

Congresos

- UML Conference. La séptima edición, <<UML>> 2004, tendrá lugar en Lisboa (Portugal), del 11 al 15 de Octubre, 2004: <<http://www.umlconference.org/>>.

Revistas

- Communications of the ACM, <<http://www.acm.org/cacm/>>:
 - Volume 42, Issue 10 (October 1999): UML.
 - Volume 45, Issue 11 (November 2002): "What UML should be".
- IEEE Software, <<http://www.computer.org/software/>>, September/March 2003: Model-Driven Development.
- Software and System Modeling, <<http://www.sosym.org/>>:
 - October 2003, December 2003 and March 2004: UML 2.0.
 - July 2003: Model engineering.
 - December 2002: Special issue on UML Conference.