

Creación y evaluación de modelos LSP en un contexto MDA

Ana Funes¹, Elizabeth Reinoso², Marcelo Castro², Aristides Dasso¹,

¹Universidad Nacional de San Luis, Ejército de los Andes 950

5700 San Luis, Argentina

{afunes, arisdas}@unsl.edu.ar,

²Universidad Nacional de Jujuy,

{mcastro, edrreinoso}@fi.unju.edu.ar

Resumen

En la línea de investigación aquí presentada, nos ocupamos de la propuesta y desarrollo de una herramienta que permita la creación de modelos de agregación, propios del método LSP (*Logic Score of Preference*), y su uso para la evaluación y comparación de sistemas, todo esto en un contexto MDA.

Contexto

El presente trabajo de investigación se encuentra enmarcado en una colaboración entre investigadores de la línea “Métodos Formales y Prototipos Evolutivos” del Proyecto de Incentivos código 22/F822 “Ingeniería de Software: Conceptos, Métodos y Herramientas en un contexto de Ingeniería de Software en Evolución”, de la Universidad Nacional de San Luis e investigadores del Proyecto “Desarrollo de una herramienta para automatizar el proceso de Gobierno electrónico” código 08/D110 del sistema de incentivos, perteneciente a la línea prioritaria “Ingeniería de Software”, de la Universidad Nacional de Jujuy.

Palabras clave: Método LSP, Evaluación de sistemas, MDA, DSL.

Introducción

El método LSP (Logic Score of Preference) ([8], [9], [10], [11], [12], [13]) es un método para la creación y aplicación de modelos en la evaluación, optimización, comparación y selección de sistemas complejos de todo tipo (no sólo sistemas basados en computadoras).

El método propone:

- (a) La creación de un modelo de los requerimientos del usuario, llamado *árbol de preferencias*. En él deben determinarse cuáles son los principales atributos del sistema, llamados *variables de performance*, y los valores posibles que dichas variables pueden tomar.
- (b) La definición de funciones, llamadas *criterios elementales*, que transforman los valores de las variables de performance en valores en el intervalo [0,100], los cuales representan el porcentaje de cumplimiento del correspondiente requerimiento, llamados *preferencias elementales*.
- (c) La creación de una estructura de agregación, cuya entrada son las preferencias elementales. Este modelo consiste en la agregación, en uno o más niveles, de las preferencias elementales y de las preferencias intermedias por medio de operadores lógicos GCD (Generalized Conjunction Disjunction) de una Lógica Continua [10], [11]. El modelo completo final de esta estructura de agregación (o

función de criterio LSP) devuelve un único valor global que es un indicador del grado de cumplimiento con los requerimientos del sistema.

Una vez que la estructura de agregación ha sido calibrada, cada sistema bajo evaluación puede ser evaluado.

Por otro lado, la Ingeniería Dirigida por Modelos, más conocida por la sigla MDE (Model Driven Engineering), es un enfoque de desarrollo de software que se basa en el uso de modelos de software como forma primaria de expresión con el objetivo de desarrollar, mantener y hacer evolucionar el software por medio de transformaciones de modelos [16], [22], [27].

Cuando construimos modelos y herramientas de modelado, es necesario conocer la estructura y las reglas de buena formación del lenguaje en el cual los modelos serán expresados. Tales modelos son llamados metamodelos [16]. Contar con metamodelos precisos es un prerequisite fundamental para poder llevar a cabo transformaciones automáticas entre modelos así como para poder definir Lenguajes de Dominio Específico (Domain Specific Language, DSL). En este sentido, MDE combina los siguientes conceptos:

- Lenguajes de Dominio Específico, que formalizan la estructura de la aplicación, el comportamiento y los requisitos dentro de un dominio particular. Estos lenguajes son descriptos usando metamodelos, los cuales definen las relaciones existentes entre los elementos de un dominio.
- Motores de transformación y generadores. Analizan ciertos aspectos de los modelos y después crean varios tipos de artefactos, tales como código fuente, entradas de simulación, descripciones de uso XML o representaciones alternativas de dichos modelos.

Mediante los DSL se consiguen notaciones de modelado distintas para cada tipo de sistema, las cuales están definidas formalmente por medio de su metamodelo. De esta manera, el ingeniero de software

cuenta con herramientas específicas para cada tipo de sistema, lo cual le permite modelarlos de una manera más detallada y precisa de acuerdo al dominio al que pertenecen.

Mediante los motores de transformación se facilita la evolución de estos modelos, pudiendo llevar a cabo transformaciones de unos a otros, según reglas de transformación.

El consorcio OMG ha desarrollado la propuesta Arquitectura Dirigida por Modelos (Model Driven Architecture, MDA) [21], [24], [25] como una implementación de MDE. MDA nace con la idea establecida de separar, en el sistema, por un lado la especificación y por el otro la lógica operacional; es decir, separar la especificación del sistema de los detalles que definen cómo el sistema usa las capacidades de la plataforma tecnológica donde es implementado.

Por lo tanto, el desarrollador sólo se preocupa de la lógica del negocio y las herramientas específicas generan todo el código relacionado con las plataformas de implementación.

En este trabajo, presentamos una línea de investigación en la que, adhiriendo a los principios de MDA, se desarrolla una herramienta de software que facilite la creación y evaluación de modelos de sistemas, definidos de acuerdo al DSL del método LSP.

Un elemento crucial en MDA es el uso de la tecnología MOF (Meta Object Facility) [26] y la definición de metamodelos que sean instancias del meta-metamodelo MOF. Un metamodelo es un modelo para definir modelos, que nos dice cuáles son los elementos que podemos usar para construir nuestros modelos así como cuáles son las relaciones existentes entre dichos elementos. Cada uno de estos metamodelos define un DSL, que presenta una solución al modelado de distintos tipos de sistemas de software. Por ejemplo, existe el metamodelo UML para modelar la

arquitectura de sistemas discretos orientados a objetos, o el metamodelo SPEM para modelar procesos de software.

En nuestro caso, es necesario definir nuestro DSL o metamodelo LSP que nos permita especificar cuáles son los elementos con los que contamos para construir modelos LSP, así como cuáles son las relaciones existentes entre dichos elementos de modelado.

Una herramienta imprescindible para que MDA tenga éxito es el desarrollo de editores específicos para metamodelos. Hasta ahora como no han existido herramientas de esta naturaleza se ha optado por el uso de perfiles UML para crear lenguajes de modelado, pero la tendencia está cambiando.

Entre los generadores de editores para MDA se destaca el proyecto GMF (Graphical Modeling Framework) [15] dentro de Eclipse, que ofrece un marco en la plataforma Eclipse para el desarrollo de editores gráficos basado en el uso de modelos sobre EMF (the Eclipse Modeling Framework) [14] y GEF (Graphical Editing Framework) [19][20].

Resultados y Objetivos

Este trabajo de investigación tiene como objetivo principal el desarrollo de una herramienta dirigida por modelos que facilite la creación y mantenimiento de modelos de evaluación de sistemas basados en un DSL.

El DSL que se pretende emplear es el que propone el método LSP y que es usado para la creación de modelos de evaluación de sistemas complejos.

Dicha herramienta no sólo deberá permitir la creación de diversos modelos (nivel M1 en la jerarquía MOF) como instancias del metamodelo LSP (nivel M2 en MOF) sino que también deberá permitir la instanciación a nivel M0 para cada sistema que se desee evaluar.

Asimismo, un objetivo que se desprende de lo anterior, ya que se trata de una aproximación MDA, es la definición formal del DSL por medio del metamodelo LSP, correspondiente a un modelo M2 en la jerarquía MOF.

Líneas de Investigación y Desarrollo

Este trabajo es llevado a cabo dentro de la línea de “Métodos Formales y Prototipos Evolutivos” del proyecto de incentivos de la Universidad Nacional de San Luis, código 22/F822: “Ingeniería de Software: Conceptos, Métodos y Herramientas en un Contexto de Ingeniería de Software en Evolución”. En esa misma línea de investigación, hemos ya aplicado el método LSP para la construcción de diversos modelos de evaluación tales como los presentados en [2], [5], [6], [3], [4], [7], [17], [18], [23].

Colaboran, en el trabajo acá presentado, investigadores de la línea “Ingeniería de Software” del proyecto de incentivos 08/D110 “Desarrollo de una herramienta para automatizar el proceso de Gobierno electrónico” de la Universidad Nacional de Jujuy. Asimismo, esta investigación se encuentra relacionada con trabajos previos en el área del desarrollo de modelos de evaluación (ver [1], [27]), los que también han sido realizados dentro del mismo marco de colaboración.

En lo que hace al desarrollo de métodos y herramientas, esta investigación tiene como objetivo el concretar la construcción, en un contexto MDA, de una herramienta de software que sirva para la creación y evaluación de modelos LSP.

Formación de Recursos Humanos

La línea de investigación relativa a la evaluación sistemas, métodos y herramientas, en la que venimos trabajando

desde hace unos años, ha dado lugar a tesis de maestría en Ingeniería de Software y tesina de licenciatura, ambas defendidas en la Universidad Nacional de San Luis.

En particular, el trabajo aquí presentado forma las bases de una tesis de maestría en Ingeniería de Software de la Universidad Nacional de San Luis, que se encuentra en etapa de desarrollo.

Referencias

- [1] M. Castro, A. Dasso, A. Funes. "Modelo de Evaluación para Sitios de Gobierno Electrónico". 38 JAIIO/SIE 2009, Simposio de Informática en el Estado 2009, Mar del Plata, Argentina, August 26-28, 2009.
- [2] A. Dasso, A. Funes. "A Model for E-voting Systems Evaluation". 40 JAIIO/SIE 2011, August 29 to September 2, 2011. Córdoba, Argentina.
- [3] Dasso, A. Funes, M. Peralta, C. Salgado, "User Oriented Evaluation Models for DBMSs", 33 JAIIO (ASIS 04), Córdoba, Argentina, 20-24 de Septiembre, 2004.
- [4] Dasso, A. Funes, M. Peralta, C. Salgado, "Una Herramienta para la Evaluación de Sistemas", Workshop de Investigadores en Ciencias de la Computación, WICC 2001, Universidad Nacional de San Luis, San Luis, Argentina, Mayo 2001.
- [5] N. Debnath, A. Dasso, A. Funes, G. Montejano, D. Riesco, R. Uzal, "The LSP Method Applied to Human Resources Evaluation and Selection", Journal of Computer Science and Information Management, Publication of the Association of Management/International Association of Management, Volume 3, Number 2, 2000, ISBN 1525-4372, pp.1-12.
- [6] N. Debnath, A. Dasso, A. Funes, R. Uzal, J. Paganini. "E-government Services Offerings Evaluation Using Continuous Logic". 2007 ACS/IEEE International Conference on Computer Systems and Applications, AICCSA '2007, Amman, Jordan. Sponsored by IEEE Computer Society, Arab Computer Society, and Philadelphia University, Jordan. May 13-16, 2007.
- [7] N. Debnath, M. Peralta, C. Salgado, A. Funes, A. Dasso, D. Riesco, G. Montejano, R. Uzal, "Web Programming Language Evaluation using LSP", CAINE03 Proceedings, Las Vegas, USA, 11-13 de Noviembre, 2003. ISBN: 1-880843-49-8, pp 302-305.
- [8] J. J. Dujmovic, "A Method for Evaluation and Selection of Complex Hardware and Software Systems", The 22nd International Conference for the Resource Management and Performance Evaluation of Enterprise Computing Systems. CMG96 Proceedings, vol. 1, pp.368-378, 1996.
- [9] J. J. Dujmovic, "Quantitative Evaluation of Software", Proceedings of the IASTED International Conference on Software Engineering, edited by M.H. Hamza, pp. 3-7, IASTED/Acta Press, 1997.
- [10] J. J. Dujmovic, "Continuous Preference Logic for System Evaluation", IEEE Transactions on Fuzzy Systems, Vol. 15, N° 6, December 2007.
- [11] J.J. Dujmovic, "Characteristic forms of generalized conjunction/disjunction"; En Fuzzy Systems, 2008 (FUZZ-IEEE 2008). (IEEE World Congress on Computational Intelligence). 1-6 June 2008, pp. 1075 – 1080, ISSN: 1098-7584, E-ISBN: 978-1-4244-1819-0, Print ISBN: 978-1-4244-1818-3.
- [12] J. J. Dujmovic and A. Bayucan, "Evaluation and Comparison of Windowed environments", Proceedings of the IASTED Interna Conference Software Engineering (SE'97), pp 102-105, 1997.
- [13] J. J. Dujmovic and R. Elnicki, "A DMS Cost/Benefit Decision Model: Mathematical Models for Data management System Evaluation, Comparison, and Selection", National Bureau of Standards, Washington, D.C., No. NBS-GCR-82-374, NTIS No. PB82-170150 (155 pages), 1982.
- [14] Eclipse Modeling Framework Project (EMF). <http://www.eclipse.org/modeling/emf/>. Ultimo acceso: 21/12/2011.
- [15] Eclipse Project. Graphical Modeling Framework (GMF). <http://www.eclipse.org/modeling/gmf/>. Ultimo acceso: 22/06/2011.
- [16] J.M. Favre, Towards a basic theory to model model driven engineering. In: Proc. 3rd Workshop in Software Model Engineering (Satellite workshop at the 7th International Conference on the UML), 2004.
- [17] A. Funes, A. Dasso, J. Dujmovic, G. Montejano, D. Riesco, R. Uzal, "Web Browsers Performance Analysis using LSP Method", Proceedings de la International Conference on Software Engineering Applied to Networking & Parallel/Distributed Computing (SNPD'00), Mayo, 2000, Reims, Francia. ISBN: 0-9700776-0-2, pp. 551-558.
- [18] A. Funes, A. Dasso, C. Salgado, M. Peralta, "UML Tool Evaluation Requirements".

- Argentine Symposium on Information Systems
ASIS 2005. Rosario, Argentina. September 29-30, 2005.
- [19] Graphical Editing Framework (GEF).
http://wiki.eclipse.org/index.php/Graphical_Editing_Framework. Ultimo acceso: 21/12/2011.
- [20] IBM, Create an Eclipse-based application using the Graphical Editing Framework.
<http://www.ibm.com/developerworks/opensource/library/os-gef/>. Ultimo acceso: 21/12/2011.
- [21] S. Mellor, K. Scott, A. Uhl, D. Weise. MDA Distilled, Principles of Model Driven Architecture, Addison-Wesley Professional, 2004, ISBN 0-201-78891-8.
- [22] T. Mens, P. Van Gorp, A Taxonomy of Model Transformation. Electronic Notes in Theoretical Computer Science, Volume 152, 27 March 2006, Pages 125-142.
- [23] G. Montejano, J.J. Dujmovic, R. Uzal, D. Riesco, A. Dasso, A. Funes, "A Prototype Tool for Decision Support based in the LSP Method", Proceedings de IASTED, Las Vegas, Nevada, USA, 6-9 de Noviembre, 2000. ISBN: 0-88986-306-7, pp 1-4.
- [24] Object Management Group, Model-Driven Architecture: Vision, Standards And Emerging Technologies.
http://www.omg.org/mda/mda_files/Model-Driven_Architecture.pdf. Ultimo acceso: 21/12/2011.
- [25] Object Management Group, Model Driven Architecture (MDA) 2.0
<http://www.omg.org/mda/>. Ultimo acceso: 21/12/2011.
- [26] Object Management Group, Meta Object Facility (MOF) 2.0 <http://www.omg.org/mof/>. Ultimo acceso: 21/12/2011.
- [27] M. Castro, "Análisis de las propiedades y atributos propios de sitios de gobierno electrónico", Tesis para la Maestría en Ingeniería del Software. Departamento de Informática, Universidad Nacional de San Luis, 2010.
- [28] D.C. Schmidt, "Model-Driven Engineering". IEEE Computer 39 (2). Febrero 2006