

Integración de Métricas de Calidad del Software

Aristides Dasso, Ana Funes
Universidad Nacional de San Luis
Ejército de los Andes 950
San Luis, Argentina
{arisdas, afunes}@unsl.edu.ar

Resumen

En la línea de investigación aquí presentada, nos ocupamos de la aplicación y propuesta de uso de Lógica Continua para la integración de Métricas de Calidad del Software, fundamentalmente empleando la norma ISO/IEC 9126.

Palabras clave: *Lógica Continua. Logic Score of Preferences. LSP. Software Engineering. Métricas de calidad de software. ISO/IEC 9126.*

Contexto

Este trabajo de investigación se encuentra enmarcado dentro 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, en la línea “Métodos Formales y Prototipos Evolutivos” del mismo. Dentro del contexto de desarrollo de métodos y herramientas, esta investigación tiene como objetivo el concretar la construcción de un modelo, que sirva para la integración de métricas de calidad software empleando una Lógica Continua.

Introducción

En trabajos previos hemos desarrollado modelos para evaluación de distintas técnicas, modelos y herramientas, ya sea en el área de recursos humanos [DDF00], herramientas de software [FDD00], [DPS03], [DFP04] o servicios de e-gov [Cas10], [CDF09], entre otros. Dichos modelos han sido creados

aplicando el método Logic Score of Preferences (LSP) el cual hace uso de una Lógica Continua ([Duj07], [Duj96], [Duj97], [DB97]).

En el presente trabajo apuntamos a integrar en un único valor los distintos datos que se obtienen cuando se evalúa la calidad del software, especialmente siguiendo la norma internacional ISO/IEC 9126 [ISO9126].

ISO/IEC 9126 es una norma internacional que establece una lista de características y subcaracterísticas para evaluar la calidad del software. La misma ha sido desarrollada por ISO (International Organization for Standardization) e IEC (International Electrotechnical Commission).

Si bien el estándar mencionado establece una lista de requisitos para evaluar software, creemos importante obtener un único valor que integre los distintos resultados de las métricas empleadas.

Otro aspecto importante a tener en cuenta es que el método de integración empleado sea comprensible para el analista y que, a la vez, resulte de fácil adaptación a distintas situaciones y técnicas de evaluación de la calidad del software.

Resultados y Objetivos

Teniendo en cuenta lo dicho anteriormente, nos hemos planteado como objetivo principal integrar los datos obtenidos a partir de diversas métricas de evaluación de la calidad del software en grupos afines que nos den nuevos valores, los que a su vez pueden volver a agregarse hasta la obtención de un solo indicador global para el objeto de software bajo evaluación. Para ello empleamos

operadores de una Lógica Continua, específicamente la lógica empleada por el método Logic Score of Preferences (LSP), el cual propone la agregación de operadores lógicos GCD (Generalized Conjunction Disjunction) para construir un modelo de evaluación.

Este modelo de evaluación consiste, básicamente, en una estructura de agregación. Dichas estructuras de agregación emplean las distintas funciones (u operadores lógicos GCD) del LSP. Los datos obtenidos por las distintas métricas utilizadas, los cuales son las entradas al modelo, y que de acuerdo al método LSP reciben el nombre de Variables de Performance, son mapeados por medio de funciones, llamadas Criterios Elementales, para obtener valores en el intervalo $[0, 100]$, que reciben el nombre de Preferencias Elementales y que representan el grado de cumplimiento del requerimiento.

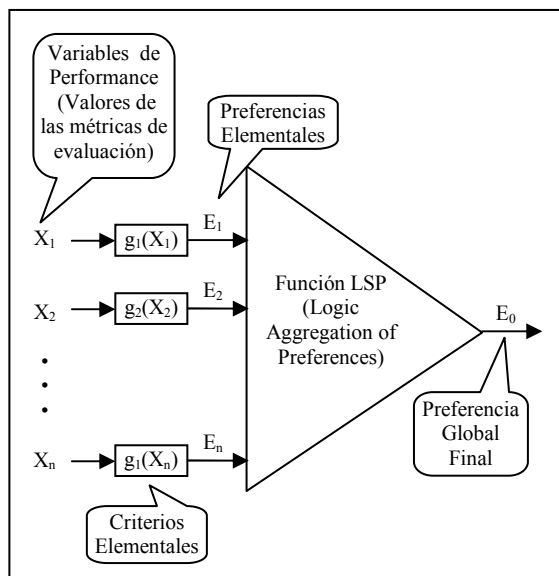


Figura 1. El proceso de evaluación del LSP.

Son estas preferencias, las que agregadas por medio de uno o más niveles de funciones de agregación, las que nos permiten obtener preferencias parciales, correspondientes a grupos de requerimientos, hasta obtener un único valor final o preferencia global final, que representa el grado de cumplimiento de todo el sistema.

En la Figura 1 mostramos una visión global del método LSP con sus correspondientes partes.

Tabla 1. Métrica Internas ISO/IEC 9126

1.	Functionality metrics
1.1.	Suitability metrics
1.2.	Accuracy metrics
1.3.	Interoperability metrics
1.4.	Security metrics
1.5.	Functionality compliance metrics
2.	Reliability metrics
2.1.	Maturity metrics
2.2.	Fault tolerance metrics
2.3.	Recoverability metrics
2.4.	Reliability compliance metrics
3.	Usability Metrics
3.1.	Understandability metrics
3.2.	Learnability metrics
3.3.	Operability metrics
3.4.	Attractiveness metrics
3.5.	Usability compliance metrics
4.	Efficiency metrics
4.1.	Time behaviour metrics
4.2.	Resource utilisation metrics
4.3.	Efficiency compliance metrics
5.	Maintainability metrics
5.1.	Analysability metrics
5.2.	Changeability metrics
5.3.	Stability metrics
5.4.	Testability metrics
5.5.	Maintainability compliance metrics
6.	Portability metrics
6.1.	Adaptability metrics
6.2.	Installability metrics
6.3.	Co-existence metrics
6.4.	Replaceability metrics
6.5.	Portability compliance metrics

En la Tabla 1 mostramos los 6 ítems de evaluación junto a sus subcategorías (nivel dos del árbol) establecidos por la norma ISO/IEC 9126 [ISO9126]. Los 6 atributos claves de calidad del software, que este estándar ha identificado, son:

Funcionalidad: Grado en que el software satisface los atributos de idoneidad, corrección, interoperabilidad, conformidad y seguridad.

Confiabilidad: Cantidad de tiempo que el software está disponible para su uso. Está definido por los la madurez, la tolerancia a los fallos y la facilidad de recuperación.

Usabilidad: Grado en que el software es fácil de usar. Comprende la facilidad de comprensión, facilidad de aprendizaje y operatividad.

Eficiencia. Grado en que el software hace un uso óptimo de los recursos del sistema. Está dado por los atributos tiempo de uso y recursos utilizados.

Facilidad de mantenimiento. Se refiere a la facilidad con que una modificación puede ser

llevada a cabo. Viene dada la facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.

Portabilidad. La facilidad con que el software puede ser llevado de un entorno a otro. Está referido por los siguientes subatributos: facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio.

En el nivel dos y tres del árbol (que no mostramos por razones de espacio), se encuentran las instancias concretas de cada una de las métricas. Dichas métricas son las que contienen los valores concretos y que de acuerdo a LSP son las variables de performance de nuestro modelo de evaluación.

Usando los operadores lógicos GCD podemos combinar las preferencias elementales (obtenidas de la aplicación de los criterios elementales sobre las variables de performance) y construir nuestro modelo de evaluación de acuerdo a las necesidades del usuario del producto de software que se desea evaluar. Así, por ejemplo, si se desea dar una mayor importancia a alguno de los atributos claves, digamos, la confiabilidad (2. Reliability), y se quiere que sean rechazados aquellos modelos que tengan un valor para esta métrica que sea inferior a un umbral determinado u , los correspondientes criterios elementales, que integran este ítem, 2.1. , 2.2. , 2.3. y 2.4., ($ce_{2.i}$) podrían ser definidos de la siguiente manera:

$$ce_{2.i}(x_{2.i}, u_i) = \begin{cases} 0 & \text{si } x_{2.i} < u_i \\ norm(x_{2.i}) & \end{cases}$$

donde:

$ce_{2.i}$ es el criterio elemental para la variable de performance $x_{2.i}$ ($1 \leq i \leq 4$).

$x_{2.i}$ es el valor de la variable de performance 2.i ($1 \leq i \leq 4$).

$norm$ es la función de normalización que devuelve un valor en el intervalo $[0,100]$.

En la Figura 2 podemos ver la estructura de agregación donde dadas las preferencias elementales $ce_{2.i}(x_{2.i}, u_i)$, surgidas de los atributos 2.1. Maturity, 2.2, Fault tolerance,

2.3. Recoverability y 2.4. Reliability compliance, son agregadas todas bajo la función LSP CA, que es una función mandatoria, es decir, si cualquiera de las preferencias elementales, que integran el ítem 2. Reliability, es igual a cero, entonces la función devolverá cero cualquiera sea el valor de las otras variables.

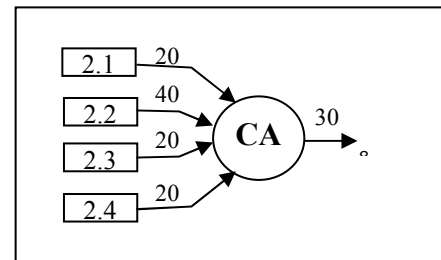


Figura 2. La función conjuntiva mandatoria CA que devuelve 0 si alguna de las entradas 2.i es 0.

Nótese que, además de poder elegirse la función de agregación, cada una de las entradas tiene asociado un peso que juega en el resultado final. En este caso, por ejemplo, el ítem 2.2, tiene un peso superior a los otros por lo que su valor tendrá una mayor importancia en el valor total de la función CA.

Este ejemplo sirve a los efectos de ilustrar, de manera sencilla, una de las tantas posibilidades de integración que nos permite el método LSP.

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” y se encuentra íntimamente relacionado con trabajos previos en el área del desarrollo de modelos de evaluación.

Cabe destacar que en esa misma línea de investigación, hemos también desarrollado modelos de evaluación de otro tipo de sistemas en trabajos tales como [DF10], [CDF09], [Cas10], [DDF00], [DDF07], [DFP04], [DFPS01], [DPS03], [FDD00], [FDPS05],

[MDU00]. La construcción de modelos de evaluación de calidad de software es uno de los proyectos en esta área. Esperamos que la misma sienta las bases para el desarrollo de nuevas tesis de grado y posgrado.

Otra línea que se está llevando adelante dentro de la misma área de investigación es la relativa a la creación de modelos de evaluación de técnicas de minería de datos, en donde se pretende el desarrollo de modelos específicos basados en la agregación de métricas para la evaluación de clasificadores.

Formación de Recursos Humanos

La evaluación de sistemas, métodos y herramientas es una de las áreas en la cual hemos estado trabajando desde hace varios años y que ha producido varias publicaciones como ya se ha mencionado más arriba. Este trabajo continuo nos ha conducido a la producción de tesis de licenciatura así como a la evaluación de sitios de gobierno electrónico, lo que ha dado como resultado tesis de maestría a la vez que otras se encuentran en preparación.

Los aspectos propios del trabajo aquí presentado son ambiciosos y se espera que las distintas tareas a desarrollar sirvan para la realización de nuevas tesis de posgrado así como de grado.

Referencias

- [DF10] Aristides Dasso, Ana Funes. "A Model for E-voting Systems Evaluation". 40 JAIIO/SIE 2011, August 29 to September 2, 2011. Córdoba, Argentina.
- [CDF09] 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.
- [DB97] 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.
- [DDF00] 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.
- [DDF07] Narayan Debnath, Aristides Dasso, Ana Funes, Roberto Uzal, José 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
- [DFP04] A. Dasso, A. Funes, M. Peralta, C. Salgado, "User Oriented Evaluation Models for DBMSs", 33 Jaiio (ASIS 04), Córdoba, Argentina, 20-24 de Septiembre, 2004.
- [DFPS01] A. 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, May 2001.
- [DPS03] 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.
- [Duj07] Jozo J. Dujmovic, "Continuous Preference Logic for System Evaluation", IEEE Transactions on Fuzzy Systems, Vol. 15, N° 6, December 2007.
- [Duj96] 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.

- [Duj97] 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.
- [FDD00] 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.
- [FDPS05] Ana Funes, Aristides Dasso, Carlos Salgado, Mario Peralta, "UML Tool Evaluation Requirements". Argentine Symposium on Information Systems ASIS 2005. Rosario, Argentina. September 29-30, 2005.
- [ISO9126] ISO/IEC TR 9126, Software engineering - Product quality, Parts 1, 2, 3, 4. ISO/IEC 200. URL: www.iso.org/
- [MDU00] 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.
- [Cas10] Castro, Marcelo; "Análisis de las propiedades y atributos propios de sitios de gobierno electrónico", Tesis para la Maestría en Ingeniería del Software. Directores: A. Dasso, A. Funes. Departamento de Informática, Universidad Nacional de San Luis, 2010.