Un Modelo Cuantitativo para Evaluar Software Educativo Libre basado en LSP

Estela Fritz[1]; María Eva Ascheri[1]; Alejandra Zangara[2]

[1]Departamento de Matemática
Universidad Nacional de La Pampa
Av. Uruguay 151 – (6300) Santa Rosa – La Pampa – Argentina
Tel.: +54-2954-245220 – Int. 7125
[fritzem, mavacheri]@exactas.unlpam.edu.ar

[2] Facultad de Informática
Universidad Nacional de La Plata
50 y 120 – (1900) La Plata – Argentina
Tel.: +54-221-4277270 / 4277271
alejandra.zangara@gmail.com

Resumen

El presente trabajo se enmarca en el Proyecto de Investigación: *"Propuesta de Clasificación de software libre utilizado en la enseñanza de la programación"*, que se desarrolla en el ámbito de la Facultad de Ciencias Exactas y Naturales de la Universidad Nacional de La Pampa. Fue aprobado, con evaluación externa, por Resolución N° 160/18 del Consejo Directivo de dicha Facultad.

El principal objetivo de este trabajo es construir un modelo para evaluar cuantitativamente software educativo y poder incluir en ese modelo las características deseables de un software libre. Dicho modelo, está basado en el método *LSP (Logic Scoring Preferences*). Por otra parte, se pretende una evaluación desde el punto de vista pedagógico más que desde el punto de vista técnico o desde la Ingeniería del software. La evaluación a través del modelo propuesto arroja una valoración cuantitativa, que permite comparar y posteriormente, seleccionar software educativo libre.

Palabras clave: LSP, software educativo, software libre, evaluación cuantitativa

1. Introducción

La rápida evolución de las Tecnologías de la Información y la Comunicación (TICs), ha creado un escenario para la enseñanza y el aprendizaje donde ya es impensado trabajar sin los recursos que brinda el uso del software.

Los recursos de software son abundantes y variados por lo que su selección se vuelve un proceso complejo si no se tienen en claro, primeramente, los objetivos de dicho proceso.

Los criterios para la selección del software educativo deben ser considerados en una instancia previa al proceso de evaluación y selección. Identificar apropiadamente esos criterios es la clave para la evaluación exitosa y posterior selección del software.

En la bibliografía disponible, no hay una lista genérica de criterios que puedan ser usados para evaluar cualquier software [1]. Los investigadores usualmente proveen una lista de criterios relacionados con la correctitud, eficiencia, usabilidad [2] y proponen métodos de evaluación basados en encuestas al usuario, cuyas respuestas organizan características según una jerarquía: buena, muy buena, aceptable, no tan buena, entre otras.

Por otro lado, el creciente número de repositorios digitales ha posibilitado que los

docentes puedan compartir sus materiales educativos, sin límite de fronteras. La mayor parte de estos materiales se comparte bajo la filosofía del software libre.

Por todo lo expuesto, frente a la gran cantidad de material disponible, es necesario que el docente pueda seleccionarlo según los objetivos que se plantea en el proceso de enseñanza- aprendizaje, los destinatarios y sus características particulares, el contexto en el cual enseña, las actividades que planifica, la evaluación y la metodología subyacente en el software, entre otros aspectos relevantes.

Como criterio para la selección del software establecido previamente, este trabajo propone la construcción de un modelo cuantitativo elaborado a través de un método sistemático: *el método LSP*.

En este modelo, el docente puede ponderar cada una de las características deseables para el software de acuerdo con la importancia o relevancia que posea dicha característica en el sistema a seleccionar.

2. Modelos para evaluar

Existen en la bibliografía numerosos modelos para evaluar software educativo. Algunos con énfasis en el proceso de diseño y desarrollo, tal como se propone en [3]. Otros enfoques se rigen por los principios propios de la Ingeniería del Software [4]. En otros casos, se busca la calidad del software según los estándares internacionales [5] y [6]. Muchas publicaciones presentan un modelo basado en listas de criterios que se formulan en forma de preguntas. Son ampliamente difundidas en este sentido las listas de control (*check-lists*), aunque contrariamente a lo que propone el método del presente trabajo, la técnica de las listas de control no permite ponderar por relevancia los diferentes ítems. Squires y Mc Dougall [7], proponen un modelo basado en interacciones entre los actores intervinientes desde la etapa de diseño del software hasta su utilización: interacciones entre diseñadores, estudiantes y profesores tomados de dos en dos; algunas de esas interacciones tienen sentido bidireccional. Este modelo propone un enfoque cualitativo y aporta un modo de pensar acerca de la selección del software.

En cuanto a los criterios a tener en cuenta para la evaluación y posterior selección del software, Insa y Morata [8] proponen tres categorías: 1) criterios técnicos que hacen referencia al software en lo concerniente a recursos audiovisuales, a la interacción y la documentación, 2) criterios didácticos, y 3) criterios de integración curricular.

Los enfoques cuantitativos buscan, fundamentalmente, objetividad y se basan en criterios preestablecidos.

3. Aspectos del modelo propuesto

El objetivo principal de esta línea de investigación y del presente trabajo, puede resumirse del siguiente modo:

- Proponer un modelo para evaluar cuantitativamente software educativo libre.

Cualquier trabajo relacionado con el uso del software libre plantea inevitablemente la cuestión de establecer decisiones relacionadas con dicha utilización.

En este trabajo se plantea un modelo para evaluar software educativo que contempla varios criterios e incluye entre los mismos el de pertenecer a la categoría de software libre. Para precisar el concepto de software libre, se exponen algunas consideraciones realizadas por el creador del proyecto GNU [9].

3.1 Software libre

Un software puede considerarse libre si el usuario posee con relación a aquél las siguientes libertades:

Libertad cero: libertad para ejecutar el software con cualquier propósito. *Libertad uno*: libertad que posee el usuario de modificar el software para ajustarlo a sus necesidades (esto podría incluir arreglar “bugs”) o para añadirle nuevas características o para migrarlo a un sistema informático diferente. *Libertad dos*: libertad de un usuario de distribuir copias del programa. Es una forma de compartir conocimiento. Pueden compartirse las

modificaciones realizadas al software. *Libertad tres*: es la libertad o posibilidad que tiene un usuario de publicar una versión mejorada de un software determinado. En la medida en que muchos usuarios realizan contribuciones, se crea una comunidad de software. Esta libertad refuerza la idea de trabajar juntos por el avance del conocimiento. Así las libertades *uno, dos y tres* son las que diferencian el software libre del software típico. Es necesario aclarar, por último, que software libre no significa que no sea comercial. La libertad tres, fundamentalmente, es lo que justifica plenamente el uso del software libre en la enseñanza ya que se relaciona con la construcción del conocimiento.

3.2 Modelo Cuantitativo

Se describe a continuación, en forma muy general, el método en el que se basa el modelo cuantitativo propuesto en este trabajo: el método *LSP*.

El método *LSP (Logic Scoring Preferences)* [10], es una generalización y extensión de varias técnicas de scoring. Es un método cuantitativo basado en el empleo de la lógica continua que permite la creación de funciones complejas de evaluación y su aplicación en la evaluación, optimización, comparación y selección de sistemas de software de propósito general.

Una vez identificadas las características que intervienen en la evaluación del software, los datos obtenidos de las distintas características del software, de acuerdo con *LSP*, serían las *variables de performance* (X_1, …, X_n) del sistema. Este proceso termina cuando derivamos componentes que ya no admiten una descomposición más simple y que pueden ser medidos. En el caso del software educativo, algunas variables de performance que se pueden describir se muestran en la fig.1.

También, deben considerarse criterios relativos a las posibilidades de un software de incluir procedimientos de evaluación.

Para cada variable de performance es necesario definir un rango aceptable de valores.

1. Grado de participación del estudiante
   - 1.1 Software tutorial
   - 1.2 Una herramienta para el aprendizaje
   - 1.3 Sistema cerrado
2. En qué medida permite el aprendizaje significativo
   - 2.1 Actividades cognitivas que puede promover
     - 2.1.1 Memorización
     - 2.1.2 Interpretación
     - 2.1.3 Comprensión
     - 2.1.4 Relaciones con conocimientos previos
     - 2.1.5 Razonamiento
       - 2.1.5.1 Razonamiento deductivo
       - 2.1.5.2 Razonamiento inductivo
       - 2.1.5.3 Razonamiento crítico
     - 2.1.6 Pensamiento divergente
     - 2.1.7 Resolución de problemas
3. Gradualidad
4. Teoría del aprendizaje subyacente
   - 4.1 Aprendizaje conductista
     - 4.1.2 Cierto grado de conductismo
     - 4.1.3 Fuerte aprendizaje conductista
   - 4.2 Aprendizaje constructivista
     - 4.2.1 Cierto grado de constructivismo
     - 4.2.1 Fuerte aprendizaje constructivista
5. Estrategias educativas que el software puede desarrollar.
   - 5.1 Entrenamiento
   - 5.2 Información
   - 5.3 Exploración
   - 5.4 Experimentación
   - 5.5 Entretenimiento
6. Rol del docente
   - 6.1 El software promueve el auto- aprendizaje
   - 6.2 El software está centrado en el aprendizaje
   - 6.3 El software está centrado en la enseñanza
   - 6.4 El software puede evaluar al estudiante

Figura 1. Variables de performance

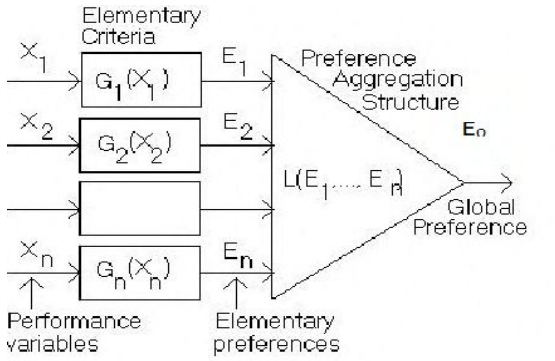
Los valores de dichas variables son mapeados por medio de funciones denominadas *criterios elementales* G_i , $1 \leq i \leq n$, en *preferencias elementales* E_i , $1 \leq i \leq n$.

Esos criterios elementales transforman el valor de una variable de performance en un valor perteneciente al intervalo [0,100] o [0,1], es decir, $0 \leq E_i \leq 1$ ó $0 \leq E_i \leq 100\%$, y expresan el grado de cumplimiento con un requisito del sistema que está siendo evaluado.

Son estas preferencias elementales las que van siendo agregadas mediante operadores de la lógica continua en estructuras de agregación

que nos permiten obtener un único valor final (*preferencia global final* E_0 en la fig. 2)

Figura 2. El proceso de evaluación de LSP



Si todos los componentes E_i no son igualmente importantes, entonces introducimos los pesos normalizados W_1, ..., W_n que son coeficientes que multiplican a las preferencias elementales y les otorgan mayor o menor relevancia en la **E**.

Generalmente,

$0 < W_i < 1$, i = 1,2,…,n y $W_1 + ... + W_n = 1$

La preferencia global se define como una media aritmética ponderada:

$E = W_1.E_1 + W_2.E_2 + … + W_n.E_n \quad 0 \leq E \leq 1$

La preferencia global se interpreta como el grado global de satisfacción de todos los requerimientos especificados.

En casos simples, la aproximación de **E** por la media aritmética ponderada puede ser adecuada. No así en los sistemas más complejos que pueden tener más de 100 variables de performance.

Como se expresó anteriormente, los modelos cuantitativos buscan objetividad en la evaluación. Para ello utilizan un conjunto de criterios preestablecidos relevantes y coherentes en contraste con evaluaciones subjetivas basadas en puntos de vista particulares (el del docente, el del estudiante, el del experto). Estos criterios preestablecidos valoran diferentes componentes del software, características deseables relacionadas con aspectos pedagógicos y didácticos.

También es posible incluir en el modelo cuantitativo, variables asociadas con las cuatro libertades de un software para ser considerado software libre. Asimismo, sería posible incluir variables (o criterios) relacionados con aspectos técnicos de diseño de software: accesibilidad, adaptabilidad, interoperabilidad, reusabilidad, entre otros.

Este modelo supone una evaluación previa del software y los criterios seleccionados no están relacionados con la utilización de éste por parte de los estudiantes. Sin embargo, el modelo es muy flexible y podrían incluirse variables (criterios) relacionadas con el uso del software una vez que el estudiante interactúa con el mismo.

4. Resultados Obtenidos/Esperados

El proyecto de investigación antes citado aborda los siguientes aspectos, relacionados a:

- Relevamiento de software libre utilizado en la enseñanza de la programación.
- Descripción de un método cuantitativo para la evaluación y selección del software libre basándose en criterios, fundamentalmente pedagógicos.
- Evaluación del software relevado con el fin de clasificarlo según los criterios preestablecidos.

El presente trabajo corresponde al segundo de los aspectos mencionados.

Después de considerar los tópicos explicados anteriormente, como resultado, puede decirse que el método *LSP* propuesto constituye un muy buen método para construir a partir de él, un modelo cuantitativo de evaluación que incrementa la eficiencia de ésta y la posterior selección de sistemas de software en general y software educativo en particular con filosofía GNU. Representa un método más ventajoso cuando se lo compara con otros métodos cuantitativos [11]. *LSP* tiene principalmente las siguientes ventajas relativas a los métodos cuantitativos para evaluación de sistemas:

(1) La especificación de requerimientos es sistemática, flexible y completa.

(2) El proceso de evaluación es sistemático y racional, el cual refleja explícita y cuantitativamente el nivel global de satisfacción de los requerimientos del usuario.

(3) Es un modelo que permite objetividad en la evaluación.

El método *LSP* puede ser usado para la evaluación y comparación de una amplia variedad de sistemas complejos [12]. El principal poder de la estrategia de *LSP* es la capacidad de construir un modelo versátil de un conjunto de preferencias. Mediante la combinación apropiada de operadores (lógicos) de agregación con el conjunto de preferencias, es posible derivar criterios complejos que tienen poder expresivo y flexibilidad.

El software educativo debe satisfacer una variedad de requerimientos. El nivel global de satisfacción de esos requerimientos (**E**) es usado para su evaluación y comparación.

Además, se espera que esta investigación brinde a los docentes herramientas cuantitativas para la evaluación y selección de software, en función de los objetivos que desean alcanzar.

Por otro lado, el proyecto de investigación antes mencionado, dirigido por la Mg. María Eva Ascheri de la Facultad de Ciencias Exactas y Naturales de la Universidad Nacional de La Pampa (UNLPam) y co-dirigido por la Dra. Alejandra Zangara de la Facultad de Informática de la Universidad Nacional de La Plata (UNLP), incluye a la Prof. Estela Fritz como investigadora. Dicho proyecto constituye su Trabajo Final para alcanzar el grado de Especialista en Tecnología Informática Aplicada en Educación. Por ello, además, como resultado se espera que la Prof. Fritz alcance el grado de Especialista en Tecnología Informática Aplicada en Educación y que, en un futuro, sea la base de su tesis de maestría.

Referencias

[1] Abohamad, W., Arisha, A. *Evaluating and Selecting Optimization Software Packages: A Framework for Business Application* World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering. Vol.4https://waset.org/publications/4426/evaluating-and-selecting-optimization-software-packages-a-framework-for-business-applications, 2010.

[2] Gorga, G., Madoz, M., Pesado P. *Hacia una propuesta de métrica para la evaluación de Software Educativo,* CACIC, 2000.
En línea: http://hdl.handle.net/10915/23514

[3] Zaragoza, J., Casado, A**.** *Aspectos técnicos y pedagógicos del ordenador en la escuela*. Editorial Bruno, Madrid, España, 2010.

[4] Galvis, A. *Fundamentos de tecnología educativa.* Editorial de la Universidad Estatal a Distancia. Costa Rica, 1995.

[5] International Standard. ISO/IEC 9126 Information Technology-Software Product Evaluation - Quality characteristics and guidelines for their use, 1991.

[6] International Standard ISO/IEC 9001 Quality Systems - Model for quality assurance in design/development production, installation and servicing, 1991.

[7] Squires, D., McDougall, A**.** *Cómo elegir y utilizar software educativo*. Traducción Pablo Manzano. Ediciones Morata y Fundación Pandeia, Madrid, 1° ed, 1997.

[8] Insa, D., Morata, R**.** *Multimedia e Internet: las nuevas tecnologías aplicadas en la educación*. 1998.

[9] Stallman, R**.** *Software libre para una Sociedad libre*. Traficantes de sueños, Madrid, 2004.

[10] Dujmovic, J.J**.** *A Method for Evaluation and Selection of Complex Hardware and SoftwareSystems*. Proceedings of the 22nd Inter. Conf. for the Resource Management and Performance Evaluation of Enterprise CS, CMG 96. 368-378, 1, 1996.

[11] Daso, A. et al. *Desarrollo de Modelos de Evaluación Usando Operadores.* de una *Lógica Continua*. XV WICC (Workshop de Investigadores en Ciencias de la Computación), 420-424, 2013.

[12] Dujmović, J. J., Hajime N. *LSP Method and Its Use for Evaluation of Java IDEs.* International Journal of Approximate Reasoning 41.1: 3–22. Web, 2006.