

Evaluación cuantitativa y sistemática de Software Educativo Libre

Estela Fritz¹; María Eva Ascheri¹; Alejandra Zangara²

¹Departamento 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.

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. 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. El modelo presentado en este trabajo está basado en el método LSP.

Palabras clave: Método LSP, evaluación de software, software educativo, software libre, métodos cuantitativos.

Contexto

Por Resolución N° 160/18 del Consejo Directivo de la Facultad de Ciencias Exactas y Naturales de la Universidad Nacional de La Pampa se acredita el Proyecto de

Investigación: "*Propuesta de Clasificación de software libre utilizado en la enseñanza de la programación*". La Directora es la Mag. María Eva Ascheri de la Facultad de Ciencias Exactas y Naturales de la Universidad Nacional de La Pampa (UNLPam) y la Co-Directora es la Dra. Alejandra Zangara de la Facultad de Informática de la Universidad Nacional de La Plata (UNLP). El mencionado proyecto incluye a la Profesora Estela Marisa Fritz como investigadora. Constituye su Trabajo Final para alcanzar el grado de Especialista en Tecnología Informática Aplicada en Educación. Además, será la base para desarrollar su futura tesis de maestría a llevarse a cabo en la UNLP.

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.

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.

Los criterios para la selección del software deben ser establecidos previamente. Por ello 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 a la importancia o relevancia que posea dicha característica en el sistema a seleccionar.

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 [1]. Otros enfoques se rigen por los principios propios de la Ingeniería del Software [2]. Se busca en otros casos la calidad del software según los estándares internacionales [3] y [4]. 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 [5], 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, D. y Morata, R. [6] 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.

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.

Por ello, se hará una referencia breve a los diferentes conceptos que constituyen los ejes de este objetivo: software libre- modelo cuantitativo.

1.1 Software 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 [7].

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. Podríamos resumirlo del siguiente modo: *libertad uno* es ayudarse a uno mismo, *libertad dos* es ayudar a otros (distribuir copias) y *libertad tres* es la libertad de construir la comunidad de usuarios. 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.

1.2 Modelo Cuantitativo

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: adecuación a los objetivos de aprendizaje, enfoque pedagógico (constructivista o cognitivista), orientación hacia los alumnos, grado de intervención docente, inclusión de módulos de evaluación y seguimiento, entre otros.

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, por nombrar algunos.

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, ya que el proceso de evaluación tiene como finalidad la selección del software más adecuado y que reúna las características buscadas o se aproxime a ello, lo más posible. 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.

1.2.1 Construcción del modelo

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 (Logic Scoring Preferences)* [8].

LSP es un método cuantitativo basado en el empleo de la lógica continua que posibilita la creación de funciones complejas que intervienen en la evaluación, comparación y posterior selección de sistemas de software de propósito general [9].

Una primera etapa en la construcción de esta función compleja consiste en establecer las características deseables para el software a evaluar. Esas características se transforman según LSP en *variables de performance* (X_1, \dots, X_n). En el caso del software educativo, algunas variables de performance que se pueden describir son las siguientes: en qué medida es tenido en cuenta el aprendizaje individual, cuantificar fases de aprendizaje inmediato y de largo alcance, gradualidad del software para el aprendizaje (¿Es posible identificar esa gradualidad?), en qué medida el software está basado en el conocimiento de cómo aprende el ser humano, en qué grado contribuye el software a los principios de contigüidad, repetición y refuerzo, qué contenidos de la memoria o capacidades aprendidas el software contribuye a recuperar, qué categorías (en cuanto a resultados del aprendizaje) el software tiende a desarrollar, cuál es el rol del docente en el proceso, en qué

medida el sistema guía al sujeto hacia el autoaprendizaje. Esto por mencionar algunas.

Para cada variable de performance es necesario definir un rango aceptable de valores. Los valores de dichas variables, se transforman, por la aplicación de funciones llamadas *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, $0 \leq E_i \leq 100$ ó $0 \leq E_i \leq 1$ y expresan el grado de cumplimiento con un requisito del sistema que está siendo evaluado. Estas preferencias elementales van siendo agregadas mediante operadores de la lógica continua en estructuras de agregación más complejas que permiten obtener un único valor final denominado *Preferencia global* E . Existe la posibilidad de ponderar las preferencias elementales dentro de la estructura de agregación, mediante pesos W_i , que son coeficientes que multiplican a las preferencias elementales y les otorgan mayor o menor relevancia en la E . La preferencia global se interpreta como el grado global de satisfacción de todos los requerimientos especificados.

2. Líneas de Investigación y Desarrollo

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.

3. Resultados

Obtenidos/Esperados

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 [10]. *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 [11]. 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.

Por otro lado, 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.

4. Formación de Recursos Humanos

En el marco del presente proyecto, como ya se mencionó anteriormente, se espera que la Prof. Fritz alcance el grado de Especialista en Tecnología Informática Aplicada en Educación y que, en un futuro, pueda desarrollar su tesis de maestría en la UNLP.

Actualmente, la Prof. Fritz ha completado la etapa de búsqueda y análisis de métodos de evaluación del software, principalmente métodos cuantitativos, tema en el que no era experta. Durante 2020, realizará todos los aportes relacionados con la definición de criterios para evaluar software educativo libre para la enseñanza de la programación, principalmente desde el punto de vista pedagógico, y tomando como base alguna teoría del aprendizaje en particular. A partir de ello, se espera construir el modelo de Preferencia Global para la posterior clasificación y selección de software. Esto constituye un insumo necesario para los nuevos avances en el proyecto.

5. Bibliografía

- [1] **Zaragoza, J., Casado, A.** (1992) “Aspectos técnicos y pedagógicos del ordenador en la escuela”. Editorial Bruno, Madrid, España.
- [2] **Galvis, A.** (1995). “Fundamentos de tecnología educativa”. Editorial de la Universidad Estatal a Distancia. San José, Costa Rica.
- [3] **International Standard. ISO/IEC 9126** (1991) Information Technology – Software Product Evaluation – Quality characteristics and guidelines for their use.
- [4] **International Standard ISO/IEC 9001** (1991) Quality Systems – Model for quality assurance in design/development production, installation and servicing.
- [5] **Squires, D., McDougall, A.** (1997) “Cómo elegir y utilizar software educativo”. Traducción Pablo Manzano. Ediciones Morata y Fundación Pandeia, Madrid, 1^o edición.
- [6] **Insa, D., Morata, R.** (1998) Multimedia e Internet: las nuevas tecnologías aplicadas en la educación.
- [7] **Stallman, R.** (2004) “Software libre para una Sociedad libre”. Traficantes de sueños, Madrid.
- [8] **Dujmovic, J.J. and Elnicki, R.** (1982) “A DMS Cost/Benefit Decision Model: Mathematical Models for Data Management System Evaluation, Comparison, and Selection”. National Bureau of Standards, Washington. DC., No NBS-GCR, NTIS No PB82- 170150 (155 pages).
- [9] **Dujmovic, J.J.** (1996) “A Method for Evaluation and Selection of Complex Hardware and Software Systems”. Proceedings of the 22nd International Conference for the Resource Management and Performance Evaluation of Enterprise CS, CMG 96, 1, 368-378.
- [10] **Daso, A. et al** (2013) “Desarrollo de Modelos de Evaluación Usando Operadores de una Lógica Continua”. XV WICC (Workshop de Investigadores en Ciencias de la Computación), págs. 420-424.
- [11] **Dujmović, J. J., Hajime Nagashima.** (2006) “LSP Method and Its Use for Evaluation of Java IDEs.” International Journal of Approximate Reasoning 41.1: 3–22. Web.