Conciencia Tecnológica No. 39, Enero-Junio 2010

Modelo para Almacenar y Recuperar Métricas de Software

Investigación

Dr. Enrique Luna Ramírez
Inst. Tec. El Llano Ags.
El Llano, Ags., México
elunaram@hotmail.com

Dr. Francisco J. Álvarez Rdz.
Univ. Autónoma de Ags.
Aguascalientes, México
fjalvar@correo.uaa.mx

Dr. J. Mauricio Espinoza Mejía
Universidad de Cuenca
Quito, Ecuador
mespinoz@ucuenca.edu.ec

M.A.T.I. Humberto Ambriz D.
Inst. Tec. El Llano Ags.
El Llano, Ags., México
humbertoambriz@hotmail.com

M.T.I. J. Antonio Nungaray O.
Inst. Tec. El Llano Ags.
El Llano, Ags., México
aenr25@hotmail.com

Resumen

La Ingeniería de Software y la Minería de Datos son *a priori* dos campos totalmente independientes entre sí, sin embargo, su combinación abre un campo de investigación de gran interés para la mejora del proceso de desarrollo de software, siendo posible obtener, a partir de un conjunto de datos, métricas que describen cómo evoluciona dicho proceso en función de resultados esperados: *tiempo, costo y calidad.* Con base en esto, se presenta una aproximación para extraer métricas de interés de un proyecto software y almacenarlas en un repositorio que permite a los usuarios finales (líderes de proyecto o ingenieros de software) consultar tales métricas mediante alias de identificadores de métricas, con la finalidad de dar una mayor flexibilidad y eficiencia al proceso de consulta.

Palabras clave: métricas software, minería de datos.

Abstract

Though Software Engineering and Data Mining are *a priori* two independent fields, its combination opens a research field of great interest for the improvement of the software development process, being possible to obtain, from a data set, metrics that describe the evolution of such a process in terms of expected results: *time, cost and quality.* Based on this, an approach for extracting specific metrics from a software project and storing them in a repository is presented. The approach allows the final users (project leaders or software engineers) to query such metrics through alias of metric´s identifiers, with the purpose of giving a major flexibility and efficiency to the query process.

Key words: software metrics, data mining.

Introducción

En la mayoría de los proyectos ingenieriles, para crear un producto, las métricas ayudan a mejorar el proceso utilizado para desarrollar el producto y por ende a elevar la calidad del mismo. Desafortunadamente, en el caso de la Ingeniería de Software, la medición se aleja de lo común, existiendo dificultades en ponerse de acuerdo sobre qué medir y cómo medir [1]. Algunas directrices que dan la pauta para definir métricas útiles y confiables son las siguientes: detectar, analizar y corregir defectos, establecer las mejores prácticas, determinar la complejidad de un proyecto, evaluar la productividad de un desarrollador, construir una base histórica para las estimaciones, evaluar los beneficios del uso de nuevos métodos y/o herramientas, detectar cuando se ha logrado la calidad esperada en el producto y en el proceso, etc.

Así, desde hace mucho tiempo se ha discutido el tema de la medición en el proceso de desarrollo de software y sus beneficios en cuanto a la mejora de procesos y calidad de productos, por lo que la recolección, el almacenamiento y el análisis de métricas se han convertido en un proceso crítico para el éxito de un proyecto de desarrollo de software. En este sentido, el objetivo de este trabajo es diseñar un modelo para extraer, almacenar y recuperar métricas de software haciendo uso de herramientas de minería de datos; el diseño del modelo incluye definir la arquitectura para estructuras y procesos, diseñar un modelo conceptual, diseñar un modelo relacional y construir un prototipo para evaluar los resultados.

A lo largo de las siguientes secciones se describirá el trabajo realizado, iniciando con los fundamentos teóricos sobre los temas de métricas de software y minería de datos, continuando con la metodología y la recopilación del estado del arte del tema en cuestión, para terminar con los resultados obtenidos hasta el momento, las conclusiones correspondientes y un planteamiento de trabajo futuro.

Fundamentos teóricos

Las primeras etapas del desarrollo de software son cruciales en la consecución de productos de calidad dentro de los límites de tiempo y costo establecidos para un proyecto. Los errores introducidos en estas primeras etapas son causa frecuente de dificultades en el mantenimiento, baja reutilización y comportamiento defectuoso de los programas. Igualmente, las malas estimaciones realizadas al comienzo del proyecto tienen consecuencias desastrosas en cuanto a costos y plazos de entrega. Por estas razones, la medición del software en el ámbito de especificación de requisitos es de suma importancia. Estudios relacionados se han centrado fundamentalmente en el desarrollo de métricas para determinar el tamaño y la funcionalidad del software, siendo comunes las métricas *de puntos de función* [2], las métricas *Bang* [3] y los *puntos objeto* [4].

La medición de atributos que definen la calidad de un producto software en el ámbito de la especificación de requisitos ha sido también objeto de algunos trabajos que van desde la medición de especificaciones formales [5] hasta la aplicación de métricas para evaluar la calidad de especificaciones expresadas informalmente en lenguaje natural [6,7], pasando por técnicas orientadas a determinar el cumplimiento de estándares, especificaciones y procedimientos [8,9,10]. Recientemente se han propuesto métricas para la evaluación de la calidad a partir de modelos producidos en etapas iniciales del ciclo de vida, como son las métricas de calidad y complejidad en modelos OMT [11], métricas de calidad de los diagramas de clases en UML [11,12] y técnicas de medición de modelos conceptuales basados en eventos [13].

La construcción de modelos para medir diferentes aspectos del proceso de desarrollo de software requiere la recolección de numerosos datos procedentes de observaciones empíricas. Los avances tecnológicos actuales posibilitan la rápida obtención de grandes cantidades de datos de fuentes muy diversas, así como el almacenamiento eficiente de los mismos. Tales datos encierran información muy valiosa que puede tratarse mediante los métodos tradicionales de análisis de datos; sin embargo, estos métodos no son capaces de encontrar toda la información útil latente en la gran masa de datos que se maneja.

Las técnicas de minería de datos surgen como las mejores herramientas para realizar exploraciones más profundas y extraer información nueva, útil y no trivial que se encuentra oculta en los grandes volúmenes de datos. *Estas técnicas han dado lugar a una paulatina sustitución de la verificación de datos por un enfoque de análisis de datos,* siendo la principal diferencia la posibilidad de descubrir información sin necesidad de formular previamente una hipótesis. Así, la aplicación de algoritmos de minería de datos permite detectar fácilmente patrones en los datos, razón por la cual estas técnicas son mucho más eficientes que el análisis dirigido a la verificación cuando se intenta explorar datos procedentes de repositorios de gran tamaño y complejidad elevada. Cabe señalar que estas técnicas emergentes se encuentran en continua evolución como resultado de la colaboración entre diferentes campos tales como bases de datos, reconocimiento de patrones, inteligencia artificial, sistemas expertos, estadística, visualización y recuperación de información.

No obstante, la aplicación de técnicas de minería de datos requiere realizar previamente una serie de actividades encaminadas a preparar los datos de entrada debido a que comúnmente tales datos proceden de diversas fuentes, por lo que no tienen el formato adecuado o contienen ruido. El proceso completo consta de las siguientes etapas [14]:

- Determinación de objetivos.
- Preparación de datos:
  - o Selección: Identificación de las fuentes de información externas e internas y selección de los datos necesarios.
  - o Pre-procesamiento: estudio de la calidad de los datos y determinación de las operaciones de minería que se pueden realizar.
- Transformación de datos: conversión de datos en un modelo analítico.
- Minería de datos: tratamiento de datos mediante una combinación apropiada de algoritmos.
- Análisis de resultados: interpretación de los resultados de la etapa anterior, generalmente con la ayuda de una técnica de visualización.
- Asimilación de conocimiento: aplicación del conocimiento descubierto.

Aunque los pasos anteriores se realizan en el orden en que aparecen, el proceso es altamente iterativo, estableciéndose retroalimentación entre los mismos.

Así, las técnicas de minería de datos están siendo utilizadas desde hace varios años para la obtención de patrones en los datos y la extracción de información valiosa en el campo de la Ingeniería de Software. Entre las aplicaciones relevantes se puede citar la utilización de árboles de decisión en la construcción de modelos de clasificación de diferentes características del desarrollo de software [15,16], el uso de técnicas de *"clustering"* en la planificación del mantenimiento [17] y en la estimación de la fiabilidad del software [18], y el uso de redes neuronales en la predicción de riesgos de mantenimiento en módulos de programa [19]. La mayoría de los trabajos realizados están orientados a la construcción de modelos para la estimación de esfuerzo de desarrollo [20] y modelos para la predicción de diferentes aspectos de la calidad del software [21].

Metodología

Dado que este trabajo consiste básicamente en diseñar un modelo e implementarlo, la metodología se reduce a la ejecución secuencial de las siguientes actividades:

- Recopilación del estado del arte sobre la aplicación de la Minería de Datos en la Ingeniería de Software.
- Diseño de la arquitectura del modelo para extraer, almacenar y recuperar métricas de una base de datos de proyectos de desarrollo de software.
- Diseño de un modelo conceptual a partir del diseño de la arquitectura.
- Diseño de un modelo relacional derivado del modelo conceptual.
- Implementación del modelo como una aplicación Web, que permita tener acceso al repositorio de métricas de manera remota.

Estado del arte

A pesar de que las técnicas de minería de datos en el campo de la Ingeniería de Software aún están en desarrollo, en la actualidad existen esfuerzos importantes que abordan diversos aspectos en torno a este tema. A continuación se describen brevemente algunos de los trabajos más relevantes.

Xie et al. [22] discuten la aplicación de algunos algoritmos de minería de datos a diversas tareas involucradas en el proceso de ingeniería de software, esto para mejorar la productividad y calidad del software. Los autores presentan varios algoritmos para minar secuencias, gráficos y texto de los datos generados durante dicho proceso.

Menzies y Boetticher [23] discuten algunos enfoques prácticos para minar datos que incluyen redes neuronales, algoritmos genéticos, árboles de decisión, reglas de asociación, técnicas Bayesianas, programación lógica inductiva y planeación temprana del ciclo de vida de un proyecto.

Eno y Thompson [24] discuten la aplicación de algunas técnicas de minería de datos, particularmente *árboles de decisión,* para descubrir patrones que pueden ser utilizados para transformar datos originales en conjuntos de datos sintéticos, útiles en la evaluación de nuevo software. De esta manera, los autores proponen un enfoque para la evaluación de software basada en datos derivados más que en datos originales, manteniendo así su integridad. Su trabajo se basa en dos tecnologías: un lenguaje de definición de datos sintéticos y un lenguaje de modelado predictivo.

Auer et al. [25] proponen un método para preparar métricas de software basado en la aplicación de técnicas para visualizar datos multidimensionales provenientes de diversas fuentes. Su enfoque permite, según los autores, el análisis exploratorio de información histórica.

Wang et al. [26] presentan un modelo basado en redes neuronales para predecir la calidad del software. Los autores proponen el entrenamiento de una red neuronal utilizando datos coleccionados de un sistema de software para telecomunicaciones, extrayendo luego reglas de la red neuronal entrenada por medio de algoritmos genéticos. Así, las reglas extraídas son utilizadas para detectar posibles módulos con fallas o propensos a fallas a la hora de su liberación.

Gousios y Spinellis [27] presentan una herramienta orientada a minar repositorios de software. La presentan como una plataforma para el análisis de la calidad del software, específicamente diseñada para apoyar la investigación en el campo de la Ingeniería de Software, integrando grandes volúmenes de datos provenientes de diversas fuentes.

Alcalá-Fernández et al. [28] presentan una herramienta, denominada KEEL, concebida para evaluar los algoritmos evolutivos que se utilizan en problemas de minería de datos. Los autores incluyen en su trabajo técnicas de regresión, clasificación, clustering, minería de patrones, etc., basándose en diferentes enfoques tales como Pittsburgh, Michigan, IRL and GCCL para lograr su propósito.

Shen y Liu [29] presentan un enfoque para detectar defectos en el software basándose en el análisis de reglas de conexión, realizado mediante técnicas de minería de datos. Su propósito es decrementar los defectos en el software y por ende lograr una mayor confianza en el mismo.

Mertik et al. [30] presentan una herramienta para minar datos, Multimethod, basada en un modelo de predicción de fallas en la construcción de software. La herramienta combina diferentes aspectos de métodos de aprendizaje supervisado en ambientes dinámicos.

Resultados

Hasta el momento se han logrado todos los objetivos específicos planteados para la primera fase de este proyecto, a decir, la recopilación del estado del arte, el diseño del modelo y la construcción de un prototipo inicial que permita evaluar la operabilidad del repositorio como una aplicación Web, habiéndose utilizado el lenguaje de programación PHP [31] y el gestor de bases de datos MySQL [32] para el desarrollo de la aplicación.

Esta sección comenzará con una descripción completa del modelo propuesto y posteriormente se hará una breve descripción del prototipo. En principio, en la figura 1 se muestra de manera genérica nuestro enfoque para proveer al usuario final de un repositorio de métricas de software.

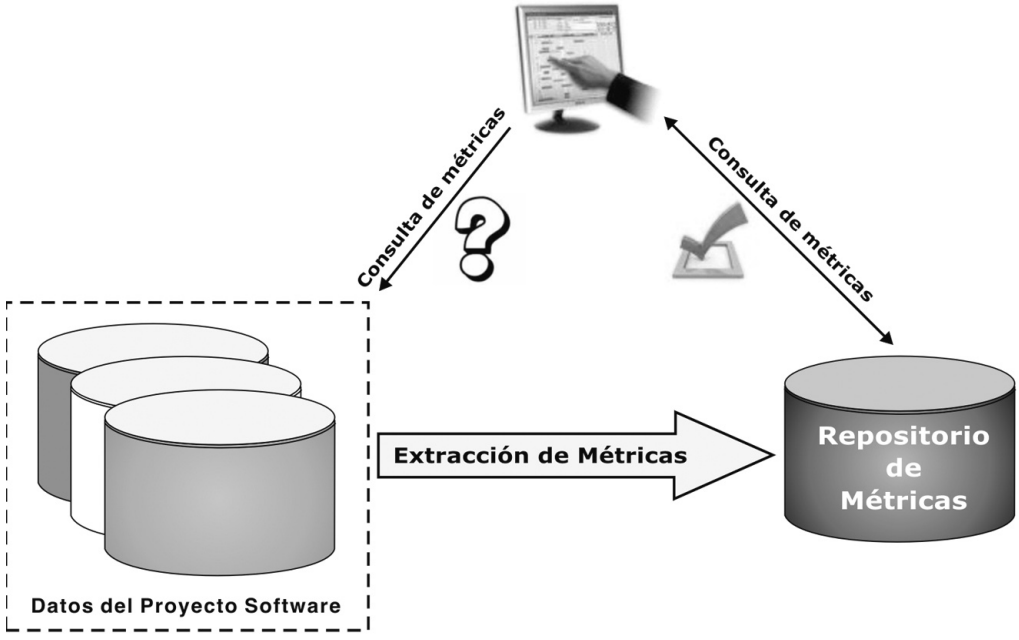

Figura 1. Esquema genérico para construir un repositorio de métricas de software.

Se pretende dotar al repositorio de una estructura que permita consultar métricas de interés en lenguaje natural (mediante el uso de alias), a la vez que las consultas puedan ser ejecutadas directamente sobre el repositorio, sin necesidad de acudir a las bases de datos originales. De esta manera, el repositorio contendrá las métricas que se vayan extrayendo de las bases de datos originales mediante herramientas de minería de datos. Cabe mencionar que se utilizarán tanto herramientas comerciales, como de libre distribución, esto debido a las capacidades que ofrecen unas y otras para propósitos de nuestro proyecto.

Para proveer al repositorio de una estructura que permita consultas en lenguaje natural, se propone un modelo compuesto por una parte estática para almacenar las métricas y una parte dinámica para recuperarlas. Es decir, la estructura del repositorio consistirá precisamente en estas dos partes que interactuarán entre sí y con las herramientas de minería de datos para lograr el propósito de ofrecerle al usuario final una herramienta que realmente le apoye en su labor. En la figura 2 se muestra un esquema de dicho modelo, en el cual se pueden observar dos *actores* principales: el usuario final y el administrador del repositorio de métricas.

El proceso para la gestión de métricas es llevado a cabo por el administrador e inicia con su extracción de las bases de datos de un proyecto de desarrollo de software mediante la ejecución de herramientas para minar datos; las métricas extraídas son clasificadas de acuerdo a su función y posteriormente almacenadas. Es decir, las métricas pueden ser clasificadas como *métricas técnicas, de calidad, de productividad, orientadas al tamaño, orientadas a la función, orientadas a los desarrolladores,* etc., permitiéndose la posibilidad de realizar subclasificaciones a cualquier nivel. De esta manera, la información extraída es ubicada en un conjunto de estructuras que permitirán identificarla y almacenarla en forma de metadatos. Estas estructuras pueden ser identificadas en la figura 2 con el nombre de MÉTRICAS, METADATOS, PALABRAS y SINÓNIMOS. Cabe señalar que estas dos últimas estructuras están pensadas precisamente para poder identificar métricas por medio de alias.

Por su parte, el proceso de consulta de métricas es llevado a cabo por el usuario final, quien echa a andar dicho proceso al realizar una consulta específica, que en primera instancia es comparada con los identificadores almacenados en la estructura MÉTRICAS; si la consulta coincide con alguno de ellos, simplemente se extraen los metadatos correspondientes (de la estructura METADATOS). En caso contrario, la frase consultada es descompuesta en las palabras que la componen, identificándose después los sinónimos de cada palabra; esto permitirá generar diversos identificadores, alguno de los cuales necesariamente estará almacenado en la estructura MÉTRICAS. De esta manera, nuestro propósito será diseñar el modelo antes mencionado e implementarlo en un prototipo que permitirá evaluar la aplicabilidad del modelo.

Dr. Enrique Luna Ramírez, Dr. Francisco J. Álvarez Rdz., Dr. J. Mauricio Espinoza Mejía
M.A.T.I. Humberto Ambriz D., M.T.I. J. Antonio Nungaray O.

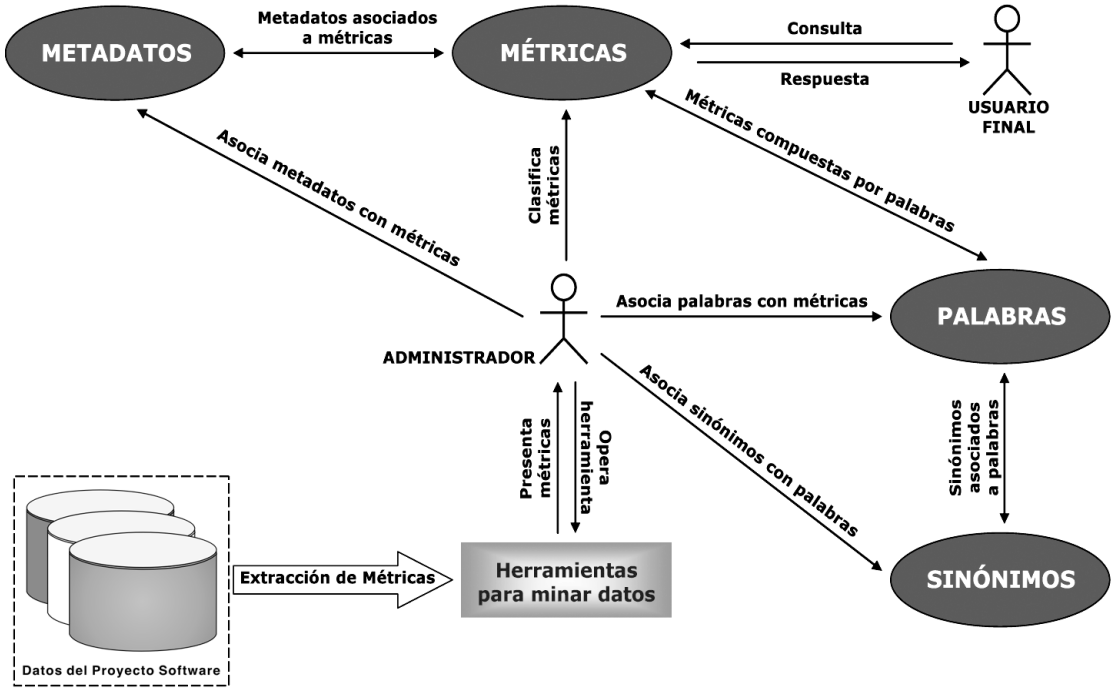

Figura 2. Modelo preliminar para realizar consultas en lenguaje natural sobre el repositorio de métricas.

En la figura 3 se muestra un prototipo inicial del repositorio de métricas propuesto, que es básicamente una fuente de consulta (de métricas) con la capacidad de operar mediante alias de identificadores. Es decir, su importancia radica en contener el conocimiento extraído de las bases de datos de proyectos de desarrollo de software y ofrecer al usuario final la posibilidad de recuperar tal conocimiento de manera eficaz y flexible.

A manera de ejemplo, se realizó una consulta bajo el identificador "Todas las métricas", que se trata de una consulta natural en el contexto de métricas. Como se observa en la figura, este identificador produjo los resultados esperados, gracias al uso de alias de identificadores. Es decir, el resultado habría sido el mismo si la consulta se hubiese realizado con algún otro identificador (frase) similar tal como "Métricas" o "Medidas", o incluso cualquier identificador que incluya estás palabras. Además, como parte de las capacidades del repositorio, es posible realizar consultas sobre métricas más específicas, de tal suerte que, por ejemplo, "Métricas de calidad" o cualquier métrica referente a calidad puede ser consultada directamente, siempre que haya sido extraída previamente de las bases de datos del proyecto.

Figura 3. Prototipo inicial del repositorio de métricas.

Para una mejor comprensión de la extracción de métricas mediante minería de datos, se presenta el siguiente ejemplo sobre un tipo particular de métricas. Las *métricas predictivas de usabilidad,* también llamadas *métricas de diseño,* son fundamentales para desarrollar sistemas interactivos de calidad. Este tipo de métricas permiten comparar y evaluar diseños de forma rápida y económica sin necesidad de construir previamente un sistema real, siendo suficiente un prototipo [33]. Para entender mejor esto, considérese el caso del desarrollo de un sitio Web, diseñado para contener información relevante sobre un tema en específico. El sitio incluye, a nivel de prototipo, un foro de discusión, diversos recursos sobre el tema (artículos, bibliografía, noticias, presentaciones, etc.) y la posibilidad de que un usuario se registre.

A partir de un conjunto de visitas al sitio, se generó una base de datos de 100 registros, cuyo formato se adecuó a la plataforma WEKA [34], herramienta utilizada para minar los datos. Para visualizar los resultados, se utilizaron gráficos provistos por esta herramienta basados en el sistema de ejes cartesianos, como se ilustra en la Figura 4.

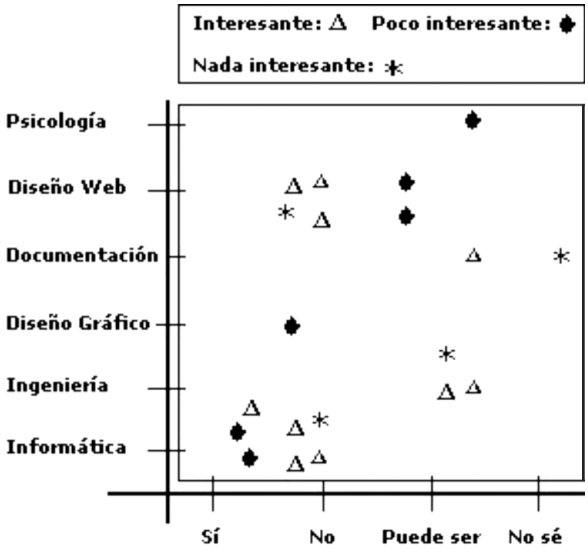

Figura 4. Gráfico obtenido para patrones interesantes.
Eje x: atributo "registrarse como usuario".
Eje y: atributo "área de conocimiento".
Símbolos: atributo "espacio de discusión".

Las reglas de asociación obtenidas permitieron evidenciar la existencia de relaciones entre los datos aportados por los usuarios que no habían sido consideradas *a priori* por los desarrolladores. Como muestra de esto, considérese la siguiente relación:

sexo= varón 13 ⇨ artículos=si 13 conf: (1)

Esta relación explicita formalmente un patrón de comportamiento observado en el 43% de los casos (13/30) que establece una relación de causa-efecto entre el sexo y los recursos preferidos por los usuarios con un 100% de certeza.

Conclusiones y trabajo futuro

La proliferación actual de métricas y el gran volumen de datos que se generan en un proyecto de desarrollo de software han puesto de manifiesto que las técnicas clásicas de análisis de datos son insuficientes para lograr los objetivos perseguidos en un determinado proyecto, por lo que nuevos enfoques como la minería de datos están cambiando el análisis tradicional de datos *dirigido a la verificación,* por un análisis de datos *dirigido al descubrimiento de conocimiento.*

Las técnicas de minería de datos, a pesar de su gran difusión en los últimos años, no han sido muy utilizadas en el campo de la Ingeniería de Software, no obstante, se presentan como una alternativa que puede ofrecer grandes beneficios en esta área, pudiendo ser aplicadas a bases de datos que almacenan información sobre procesos de desarrollo de software.

Con base en lo anterior, en este artículo se presentó un modelo para extraer métricas de interés de un proyecto software y almacenarlas en un repositorio estructurado de manera que los usuarios finales puedan consultar tales métricas mediante alias de métricas, dándole así una mayor flexibilidad y eficiencia al proceso de consulta.

Es importante señalar que ninguno de los trabajos en el estado del arte considera el uso de un repositorio con estructura propia para almacenar métricas, ni tampoco considera el uso de alias para recuperarlas, por lo que se la contribución es original.

Como trabajo futuro, en la segunda fase del proyecto se tiene considerado construir un repositorio que se alimente de información extraída de bases de datos de proyectos desarrollo de software en el área de Tecnologías de la Información de la Secretaría de Gestión e Innovación del Gobierno del Estado de Aguascalientes, para lo cual existe un convenio.

Referencias

[1] Ebert, C., Dumke, R., Bundschuh, M. & Schmietendorf, A., (2004), *Best Practices in Software Measurement,* Springer, 1st edition.

[2] Albrecht, A.J., (1979), "Measuring application development", *Proceedings of IBM Applications Development Joint SHARE/GUIDE Symposium,* Monterey, CA, pp. 83-92.

[3] DeMarco, T., (1982), *Controlling Software Projects,* Yourdon Press, USA.

[4] Boehm, B.W. & Kaspar, J.R., (1978), *Charracteristics of Software Quality,* TRW Series of Software Technology, USA.
[5] Samson, W.B. & Nevill, D.G., (1990), "Predictive software metrics based on a formal specification", *Software Engineering Journal,* 5:1, pp. 116-121.
[6] Lehner, F., (1993), "Quality Control in Software Doc.: Measurement of Text Comprehensibility", *Information and Management,* 25, pp. 133-146.
[7] Arthur, J.D. & Stevens, K.T., (1989), "Assessing the adequacy of documentation through document quality indicators", *Proceedings of the IEEE Conference of Software Maintenance,* pp. 40-49.
[8] Davis, A., (1993), "Identifying and Measuring Quality in a Software Req. Specification", *Proceedings of the First International Software Metrics Symposium,* pp. 141-152.
[9] Brykczynskki, B., (1999), "A survey of software inspection checklist", *ACM Software Engineering Notes,* 24:1, pp. 82-89.
[10] Farbey, B., (1990), "Software quality metrics: considerations on requirements specification", *Information and Software Tech.,* 32:1, pp. 60-64.
[11] Genero, M., Manso, M.E., Piattini, M. & García, F.J., (1999), "Assessing the Quality and the Complexity of OMT Models", *2nd European Software Measurements Conference,* pp. 99-109.
[12] Genero, M., Piattini, M. y Calero, C., (2000), "Una propuesta para medir la calidad de los diagramas de clases en UML", *IDEAS´2000,* Cancún, México, pp. 373-384.
[13] Poels, G., (2000), "On the measurements of event-based object-oriented conceptual models", *4th Int. ECOOP Workshop on Quantitative Approaches in Object Oriented Software Engineering,* pp. 37-42.
[14] Cabena, P., Hadjinian, P., Stadler, R., Verhees, J. & Zanasi, A., (1998), *Discovering Data Mining: From Concept to Implementation,* Prentice Hall.
[15] Khoshgoftaar, T.M. & Allen, E.B., (1999), *Modeling Software Quality with Classif. Trees,* Hoang Pham Editor - World Scientific, Singapore.
[16] Tian, J. & Palma J., (1998), "Analyzing and Improving Reliability: A Tree-based Approach", *IEEE Software,* 15:2, pp. 97-104.
[17] Krohn, U., (1999), "Application of Cluster Algorithms for Batching of Proposed Software Changes", J. *Soft. Maintenance,* 11, pp. 151-165.
[18] Podgurski, A., Masri, W., McCleese Y. & Wolff, F.G., (1999), "Estimation of Software Reliability by Stratified Sampling", *ACM Transactions on Software Engineering and Methodology,* 8:3, pp.263-283.
[19] Khoshgoftaar, T.M., (1995), "A Neural Network Approach for Early Detection of Program Modules Having High Risk in the Maintenance Phase" *Systems Software,* 29:1, pp. 85-91.
[20] Srinivasan, K., (1995), "Machine Learning Approaches to Estimating Software Development Effort", *IEEE Transactions on Software Engineering,* 21:2, pp.126-137.
[21] Khoshgoftaar, T.M., Allen, E.B., Hudepohl, J.P. & Aud, S.J., (1997), "Neural Networks for Software Quality Modeling of a Very Large Telecomunications System", *IEEE Transactions on Neural Networks,* 8:4, pp. 902-909.
[22] Xie, T., Thummalapenta, S., Lo, D. & Liu, C., (2009), "Data Mining for Software Engineering", *Computer,* 42:8, pp. 55-62.
[23] Menzies, T. & Boetticher, G.D., (2002), "Smarter software eng.: practical data mining approaches", Proc. on Software *Engineering Workshop – 27th Annual NASA Goddard,* pp. 1-38.
[24] Eno, J. & Thompson, C.W., (2008), "Generating Synthetic Data to Match Data Mining Patterns", *IEEE Internet Computing,* 12:3, pp. 78-82.
[25] Auer, M., Graser, B. & Biffl, S., (2003), "An approach to visualizing empirical software project portfolio data using multidimensional scaling", *IEEE International Conference on Information Reuse and Integration,* pp. 504-512.
[26] Wang, Q., Yu, B. & Zhu, J., (2004), "Extract rules from software quality prediction model based on neural network", *16th IEEE Int. Conference on Tools with Artificial Intelligence,* pp. 191-195.
[27] Gousios, G. & Spinellis, D., (2009), "A platform for software engineering research", *6th IEEE International Working Conference on Mining Software Repositories,* pp. 31-40.
[28] Alcala-Fernández, J., García, S., Berlanga, F.J., Fernandez, A., Sanchez, L. & Herrera, F., (2008), "KEEL: A data mining software tool integrating genetic fuzzy systems", *3rd Int. Workshop on Genetic and Evolving Systems,* pp. 83-88.
[29] Shen, Y. & Liu, J., (2009), "Research on the Application of Data Mining in Software Testing and Defects Analysis", *2nd Int. Conference on Intelligent Computation Technology and Automation,* pp. 619 – 621.
[30] Mertik, M., Lenic, M., Stiglic, G. & Kokol, P., (2006), "Estimating Soft. Quality with Advanced Data Mining Techniques", *Int. Conference on Software Engineering Advances,* p. 19.
[31] PHP 5.3.2, http://www.php.net/, visitado el 6 de mayo de 2010.
[32] MySQL Query Analyzer, http://www.mysql.com/, visitado el 6 de mayo de 2010.
[33] Constantine, L. & Lockwood, L., (1999), Software for Use: *A practical Guide to the Models and Methods of Usage-Centered Design,* ACM Press and Addison-Wesley.
[34] Weka 3: Data Mining Software in Java, http://www.cs.waikato.ac.nz/ml/weka/, visitado el 6 de mayo de 2010.

Agradecimientos: Al Cuerpo Académico "Ingeniería del Conocimiento" del I.T. El Llano Aguascalientes, al Cuerpo Académico "Objetos de Aprendizaje e Ing. de Software" de la U. Autónoma de Aguascalientes y a la Dirección Gral. de Educación Superior Tecnológica por su apoyo a través del Proyecto de Investigación Científica registrado con la clave 3527.10-P.