ReCIBE, Año 2 No.3, Diciembre 2013

Herramienta de gestión de Modelos Ontológicos aplicados a la Mejora de Procesos Software

José Eduardo Guadalupe Gaytán Solís
Centro de Investigación en Matemáticas, Unidad Zacatecas
jose.gaytan@cimat.mx

Ricardo González Saldívar
Centro de Investigación en Matemáticas, Unidad Zacatecas
ricardo.gonzalez@cimat.mx

Edrisi Muñoz Mata
Centro de Investigación en Matemáticas, Unidad Zacatecas
emunoz@cimat.mx

Resumen: Actualmente el software representa un componente principal en el desarrollo de las actividades dentro de las organizaciones. Este hecho hace que la calidad del software sea uno de los principales factores claves a los que se dedica mucho esfuerzo. La calidad del software permite de manera implícita la creación de valor agregado al software durante su desarrollo, resultando en un factor de competitividad en estas organizaciones. Éste trabajo propone una solución inteligente mediante el uso de gestión del conocimiento, con el fin de hacer que la aplicación de teorías de ingeniería de software, modelos y estándares de mejora de procesos sea de fácil

implementación y a un menor costo. Específicamente, en este trabajo se presenta el desarrollo de un modelo ontológico basado el Cuerpo de Conocimientos de la Ingeniería de Software (SWEBOK por las siglas en inglés de Software Engineering Body of Knowledge) así como su aplicación práctica mediante un sistema software. Finalmente se presentan los resultados de la implementación de técnicas, métricas y actividades correspondientes al área de conocimiento de pruebas de software de SWEBOK. Así mismo se presentan resultados de la integración del marco base de datos de lenguaje ontológico web (OWLDB por las siglas en ingles ontology web language data base), con el objetivo de almacenar datos de manera persistente durante la aplicación.

Palabras clave: **Ontología, Cuerpo de Conocimientos de la Ingeniería de Software, SWEBOK, OWLDB, Bases de Datos, Mejora de procesos de software.**

Ontologic Models management tool applied to Software Process Improvement

Abstract: **Nowadays, software is a major component in the development of activities within organizations. This fact makes software quality is one of the main key factors which demands much effort. In this way, software quality allows the creation of added value to the software during development, resulting in a competitive factor for the organization. This paper proposes an intelligent solution for a lower cost implementation of software engineering theories, models and process improvement standards by the use of knowledge management. Specifically, this paper presents the development of an ontological model based on the Body of Knowledge Software Engineering (SWEBOK) and its practical application through a software system. Finally this work presents the results of the implementation of techniques, metrics and activities for the area of software testing knowledge of SWEBOK. It also presents results of the ontology web language data base (OWLDB) frame integration, aiming to store persistent data in the application.**

Keywords: **Ontology, Software Engineering Body of Knowledge, SWEBOK, OWLDB, Data Bases, Software Process Improvement.**

1. Introducción

En la actualidad existen numerosos modelos y estándares dentro de la industria que pretenden regular y asegurar la calidad como por ejemplo CMMI, Six Sigma y SWEBOK (Pyzdek, 2003; Abran, Bourque, Dupuis, Moore & Tripp,

2004; CMMI Product Team, 2010). Estas guías metodológicas plantean procesos que establecen una manera estructurada y orientada para mantener una eficiente preparación, operación y mantenimiento de los productos que generan. Por otra parte, los datos significativos para la operación de las empresas son una fuente de conocimiento necesario para el desarrollo de las mismas y por su importancia es necesario que se le dé un trato eficiente de conservación, reutilización y gestión para mantener competitividad. Una manera de gestionar este conocimiento puede generarse a través de la utilización de los modelos ontológicos. Una ontología es una especificación formal de la conceptualización de un área o dominio en particular. Mediante el desarrollo de una ontología se establece un modelo formal, que proporciona heterogeneidad semántica. Este tipo de modelos proveen una comprensión compartida que es procesable por humanos y por aplicaciones informáticas (Cullot, Ghawi & Yétongnon, 2007; Quintanilla, 2005; Barchini & Herrera, 2010; Romá, 2009).

Por otra parte una nueva tendencia en cuanto a la administración del conocimiento es la utilización de un sistema de almacenamiento persistente a través de bases de datos. Este tipo de administración de datos aporta una extensibilidad y amplía el campo de aplicación de los modelos ontológicos (Cullot, Ghawi & Yétongnon, 2007; Trinkunas & Vasilecas, 2007; Aksoy, Alparslan, Bozdağ, Çulhacı, 2011; Heymans et al., 2008). Aún más, mediante la integración de este almacenamiento y los modelos semánticos se crean nuevos modelos y estructuras de información que dan como resultado las conocidas bases de datos inteligentes que explotan “conocimiento” y una vista semántica a la información.

El presente trabajo presenta la elaboración de una ontología basada en guía de cuerpo de conocimientos de la ingeniería de software (SWEBOK por las siglas en inglés de Software Engineering Body of Knowledge) desarrollado por el Instituto de Ingeniería de Software (Abran, Bourque, Dupuis, Moore & Tripp, 2004). A continuación se describe de manera breve el contenido de este

trabajo. La sección 2 muestra la propuesta y a la audiencia a la que está destinado el trabajo. En la sección 3 se da a conocer la metodología de desarrollo de la ontología y de la herramienta de software para la administración de la ontología. La sección 4 describe de manera breve el proceso de desarrollo de la ontología y del sistema y, de manera general, el contenido de dicha ontología, además del desarrollo de la herramienta y la implementación con OWLDB . En la sección 5 se muestran las conclusiones del trabajo realizado. Finalmente la sección 6 presenta el trabajo futuro basado en las conclusiones obtenidas.

2. Propuesta

Los modelos de calidad existentes en la actualidad están formados por una gran cantidad de conceptos, guías específicas y reglas que deben seguirse para poder ser implementados (Pyzdek, 2003; Abran, Bourque, Dupuis, Moore & Tripp, 2004; CMMI Product Team, 2010). El manejo de esta información, por parte de las empresas, puede representar una inversión relativamente considerable de recursos que pudiera resultar en un impedimento o retraso en la puesta en marcha de determinado modelo de calidad (Muñoz, Mejía, Muñoz, 2013). Esta dificultad puede llegar a ser manejable de una manera menos costosa a través del uso de la propuesta descrita en este documento. Para ello, se divide este trabajo, en dos partes principales de acción:

1. Desarrollo de la Ontología de SWEBOK. Se trata de una ontología basada en el conocimiento contenido en la guía de SWEBOK. El objetivo es dar fácil acceso al conocimiento de la guía de SWEBOK. Con ese conocimiento es posible ayudar a la definición y mejora de procesos dentro de las empresas.
2. Desarrollo del sistema software. Este sistema es derivado al trabajo de la ontología. Se trata de un sistema que permite la lectura y razonamiento de modelos ontológicos. El sistema está diseñado para leer cualquier ontología. Pero su funcionalidad está orientada al uso de la ontología de SWEBOK. Este sistema software permite la utilización del conocimiento contenido dentro de la ontología. Además tiene la capacidad de almacenar de manera

persistente la información relacionada a las operaciones de la empresa. El almacenamiento se hace dentro de una base de datos relacional (RDB, del inglés Relational Data Base). Además, el sistema software permite a usuarios sin conocimientos avanzados en el uso de herramientas ontológicas poder crear y manejar tres aspectos claves para la gestión del conocimiento como son: ontologías, inteligencia y datos dinámicos.

En resumen, el sistema está basado en la ontología de SWEBOK (aunque puede leer cualquier modelo ontológico, las excepciones de manejo y lectura de ontologías son mostradas en la sección 4.2) y almacena los datos de la ontología en una RDB.

2.1. Usuarios Finales

La estructura de la guía de SWEBOK está dividida en 10 áreas de conocimiento; la ontología aquí presentada, mantiene esa modularidad. Esto permite que la ontología pueda adaptarse a distintos requerimientos. De igual manera el uso de la ontología se puede adaptar a través del sistema software. Todo esto mediante los aspectos que puedan ser requeridos en diferentes tamaños y complejidades de proyectos.

1. PyMEs hasta empresas de gran tamaño que realizan proyectos de software de una complejidad considerable. La ontología y el sistema brindan el conocimiento necesario para poder definir y/o mejorar su definición de proceso software, definición de roles, desarrollo de políticas de evaluación y desempeño, especificar tareas de desarrollo software, implementar las prácticas establecidas por el SWEBOK y brindar fácil acceso a la información contenida en el SWEBOK.
2. Grupos de trabajo de universidades que requieran de la utilización de SWEBOK, puede ayudar a definir requerimientos de educación, formación y desarrollo de políticas de evaluación de desempeño.
3. Usuarios sin conocimientos avanzados en SWEBOK ni en el uso de herramientas ontológicas. Las propuestas contenidas en el presente trabajo ayudan a este tipo de usuarios al leer una ontología y a hacer uso de la información contenida dentro de esta (ontología), sin la necesidad de tener conocimientos avanzados sobre ontologías .

4. En general cualquier organización y usuarios que en la actualidad implementa o desee implementar las prácticas establecidas por el SWEBOK.

3. Metodología

El desarrollo de la ontología está basado en una metodología sistemática. Esta metodología, a su vez, está basada en las metodologías Methontology (Corcho, Fernández, Gómez & López, 2005) y On-To-Knowledge (Sure, Staab & Studer, 2002), ambas estructuradas dentro del ciclo de mejora continua (Muñoz, Espuña & Puigjaner 2010) o ciclo PDCA. El ciclo PDCA (Plan, Do, Check and Act), es un modelo de gestión de calidad clásica (Ning, Cheng & Liu, 2010; Zhichun, 2011). La descripción de esta metodología puede verse más a detalle en la sección 4.2 y en la Figura 1. Por otro lado, dentro del desarrollo del prototipo del sistema software, la metodología se basó en el desarrollo ágil planteado por SCRUM .

4. Desarrollo

En la actualidad, existen numerosos estándares para la ingeniería de software y procesos de producción. Este tipo de estándares, generalmente, tratan de cubrir todos los aspectos que están involucrados dentro de cierta área de conocimiento o de aplicación práctica en la industria. La cantidad de información y conceptos que contienen estos documentos puede representar una dificultad en su manejo; además, se le puede añadir a esta dificultad, la cantidad de información que es propia de la organización. Con respecto a esto, como una solución, las ontologías ayudan a la mejora de procesos de desarrollo de software. Así, ayudan en el entendimiento común de la información entre miembros del equipo de desarrollo de software (Mendes & Abran, 2004; Willie, Abran, Desharnais & Dumke 2004; Bermejo, 2006), así

como en facilitar el análisis del estado del proceso que actualmente ejecutan, identificar oportunidades de mejora y plantear metas de mejora. En este contexto, existen ontologías que han ayudado en la mejora de procesos software como lo son la ontología de Capability Maturity Model Integration for Development (CMMI-DEV) (Soydan, 2012). Existen además ontologías híbridas: la ontología que integra Personal Software Process (PSP) y herramientas utilizadas por Six Sigma (Kim et al., 2009), la ontología que integra CMMI con el estándar ISO 9001:2000 (Ferchichi, Bigand & Lefebvre, 2008) y la que une Six Sigma con Information Technology Infrastructure Library (ITIL) (Dash, 2012).

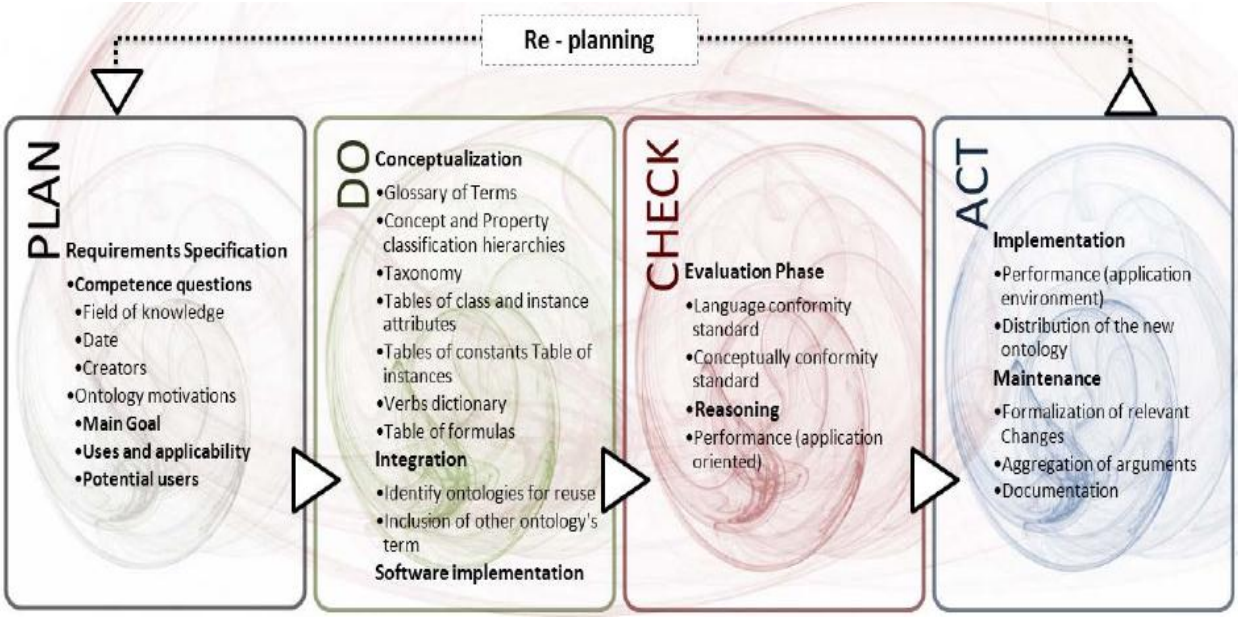

Figura 1. *Metodología propuesta para el desarrollo de la ontología SWEBOK (Muñoz, Mejía, Muñoz, 2013).*

4.1 Cuerpo de Conocimiento de la Ingeniería de Software (SWEBOK)

La guía de SWEBOK está compuesta de 10 áreas de conocimiento (Mendes & Abran, 2004), como se puede ver en la Figura 2 y tiene como objetivos:

caracterizar los contenidos de la disciplina de la ingeniería de software (IS), proveer un acceso típico al cuerpo de conocimiento de la IS, proveer una vista consistente mundial de la IS, clarificar el lugar de la IS y fijar sus límites y proveer fundamentos para desarrollo de plan de estudios y de material de certificación individual (Willie, Abran, Desharnais & Dumke., 2003, 2004; Mendes & Abran, 2004; Abran et al., 2006).

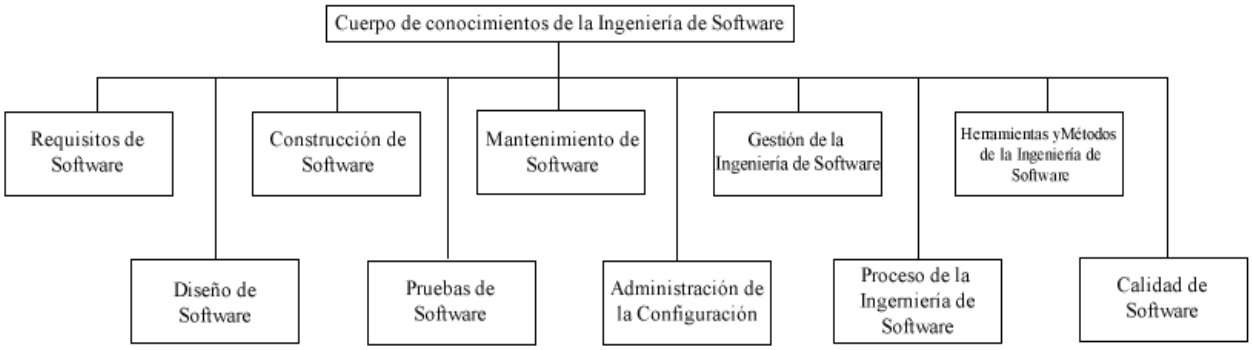

Figura 2. *Áreas del Conocimientos contenidas dentro del Cuerpo de Conocimientos de la Ingeniería de Software.*

4.2 Desarrollo de la Ontología de SWEBOK

Para la construcción de la ontología se siguen los pasos de la metodología propuesta:

Fase de Planeación. Investigación de la situación actual del dominio mediante la lectura de la guía de SWEBOK en (Abran, Bourque, Dupuis, Moore & Tripp, 2004). Derivado a esto, se hizo:

1. Una estrategia de desarrollo. Se plantean 6 actividades principales:
    1. Revisión literaria. Esta actividad significa la búsqueda de literatura relacionada con el dominio.
    2. Estudio del dominio. Se refiere al siguiente paso después de revisar la literatura con el fin de comprender plenamente el dominio.

3. Revisión de la metodología. Para el desarrollo de la ontología, la metodología debe ser revisada. Esta revisión permitirá seguir los pasos adecuados para la correcta interpretación de la ontología.
4. Implementación de la metodología. Después de revisar minuciosamente la metodología, el siguiente paso es seguir de una manera secuencial los pasos de esa metodología para el desarrollo de la ontología.
5. Entregables del proyecto. Esto se refiere a los documentos y archivos entregables (de acuerdo a la calendarización del proyecto) relacionados con la ontología.
6. Repetir las actividades desde la actividad 1 a la 6. Este es un proceso incremental e iterativo. Con el fin de lograr los resultados deseados es necesario el perfeccionamiento de las actividades.

2. Identificación del propósito. El propósito es construir una ontología para ayudar en el uso del SWEBOK y procesos de IS.
3. Usuarios finales. Los usuarios finales son mostrados en la sección 2.1.
4. Definición de dominio. La definición de dominio se encuentra en la sección 4.1.
5. Definición de alcance mediante 11 preguntas de competencia (las preguntas de competencia, son preguntas que la ontología ya construida debe de ser capaz de responder) y son las siguientes:
    1. ¿Qué es la Ingeniería de Software?
    2. ¿Cuáles son las áreas de conocimiento de SWEBOK?
    3. ¿Qué disciplinas están relacionadas con la Ingeniería de Software?
    4. ¿Qué herramientas y métodos se utilizan en Ingeniería de Software?
    5. ¿Cuáles son las características de la profesión de ingeniería del software?
    6. ¿Cuáles son los fundamentos en la construcción de software?
    7. ¿Cuáles son los fundamentos de diseño de software?
    8. ¿Cuáles son los fundamentos de requerimientos de software?
    9. ¿Cuáles son los fundamentos de pruebas de software?
    10. ¿Cuáles son los fundamentos del mantenimiento de software?
    11. ¿Cuáles son los fundamentos de calidad de software?

Fase de Desarrollo. Creación del glosario de términos. Se identifican 12 clases principales, 403 conceptos, tabla de verbos y 471 propiedades de objeto. Con el fin de la creación de la taxonomía se engloban cada una de las 403 clases dentro de una clase principal. La taxonomía es mostrada en la Figura 3.

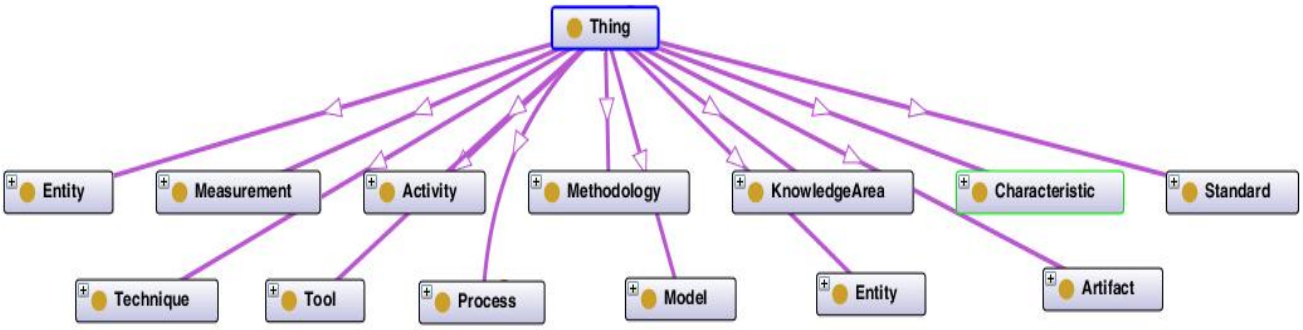

Figura 3. *Taxonomía de la Ontología de SWEBOK.*

Fase de revisión. La conceptualización de la ontología se ha verificado mediante el uso de axiomas experimentales y la introducción de instancias a través de la herramienta Protégé 4.1 . Se encontraron algunas inconsistencias, 11 subclases de la taxonomía se encontraban englobadas en la clase principal incorrecta.

Fase de Actuación. Se tomaron las acciones correctivas sugeridas en la fase anterior (fase de revisión).

Fase de Re planeación. Con el fin de ayudar en el ambiente empresarial, otras preguntas de competencia, axiomas y reglas necesitan ser planteadas para satisfacer las necesidades de mejora de definición de procesos de la compañía. En esta etapa, el uso del sistema software, 4 preguntas de competencia, 2 propiedades de objeto, 1 clase y 32 reglas son propuestas.

4.2 Desarrollo del sistema para el manejo de modelos ontológicos

El desarrollo del sistema software está enfocado a la lectura y razonamiento de una ontología modelada en lenguaje OWL (Web Ontology Language). La construcción del sistema software se basó en el desarrollo ágil guiado por la técnica SCRUM. Por otra parte Java fue empleado como lenguaje de programación, en donde se permitía la implementación y el uso de las librerías OWLAPI 3.4.3 , Pellet 2.3.0 y HermiT 2.0 , todas ellas desarrolladas en Java. Además, adicionalmente en este trabajo se han probado de manera experimental potenciales herramientas y librerías que pudiesen ser de gran ayuda para el manejo de información y razonamiento de los modelos ontológicos. Por otro lado los formatos que el sistema software soporta son RDF, OWL y Turtle. Por el contrario, los archivos en formato KRSS2 y Manchester OWL no son garantizados que se carguen correctamente dentro del sistema software propuesto (de igual manera la herramienta Protégé en sus versiones 4.1, 4.2 y 4.3 tampoco puede leer estos formatos).

El sistema software ha sido probado mediante un caso de estudio (mostrado en la sección 4.3). Posteriormente, al caso de estudio, se hizo la integración de la herramienta con el marco o framework OWLDB. OWLDB utiliza como principal medio de almacenamiento a Hybernate. Esta aplicación permite crear automáticamente una base de datos siguiendo la estructura ontológica. Durante la integración de OWLDB se tuvo que hacer un degradado con las librerías OWLAPI de 3.4.3 a 3.2.3 empleadas por el sistema, ya que OWLDB utiliza librerías OWLAPI 3.2.3, sacrificando así valiosas capacidades en inferencia y razonamiento.

4.3 Caso de Estudio

Con el fin de probar la funcionalidad de la ontología y el sistema software se brinda una solución práctica mediante un caso de estudio en una compañía.

4.3.1 Entorno de aplicación

La compañía consiste de 40 empleados compuestos por administradores, diseñadores y personal de apoyo. El estatus actual de la compañía fue definido por medio de entrevistas con el personal que trabaja en tres diferentes niveles de la compañía.

4.3.2 Hallazgos

Por medio de las entrevistas fueron encontrados varios problemas, estos incluyen: falta de definición de procesos de desarrollo software, falta de estándares y modelos de calidad. La etapa de pruebas de la compañía no está bien formalizada, donde cuatro tareas principales son identificadas:

1. Revisión inicial.
2. Planeación.
3. Revisión de ciclos.
4. Revisión de los resultados.

De las cuatro tareas, únicamente la tarea de planeación puede ser descompuesta en cinco acciones (las demás tareas no pueden ser descompuestas debido a la falta de definición del proceso):

1. Revisión de requerimientos.
2. Revisión con el cliente.
3. Análisis de riesgos.
4. Definición de casos de prueba.

5. Planeación de pruebas.

Además tiene los roles: equipo de pruebas, usuario final y equipo de desarrollo.

4.3.3 Desarrollo

El caso de estudio ha sido probado específicamente para el proceso de pruebas de la empresa. Para probar el proceso se empleó el área de conocimiento de pruebas de software de la Ontología de SWEBOK en conjunto con el sistema software. Con el fin de alcanzar la mejora de proceso de pruebas en la compañía, es necesario plantear preguntas de competencia específicas a este caso. Los conceptos contenidos en la definición del proceso de pruebas de la compañía (etapas, tareas, acciones, roles, herramientas y recursos mostrados en la sección 4.3.2) son vistos como instancias dentro de la ontología. Estas instancias fueron ingresadas de manera manual dentro de la ontología identificando la clase (o concepto) a la cual pertenece. Por ejemplo, la acción de revisión de requerimientos (dentro del proceso de la compañía) se refiere al concepto Requirements Reviews contenido en la guía de SWEBOK. Por ende, la acción de revisión de resultados es una instancia de la clase RequirementsReviews dentro de la ontología.

4.3.4 Resultados del Caso de Estudio

La introducción de la ontología de SWEBOK ayudó a la compañía a empezar una definición formal de este proceso y su mejora continua. La ontología SWEBOK brinda información acerca de en qué punto se encuentra la definición del proceso. La mejora de definición de proceso es logrado por medio de satisfacer toda el área de conocimiento de pruebas de software de SWEBOK. Esto ayuda en el análisis de brechas, al proveer aquellos conceptos cubiertos actualmente por el proceso, y aquellos cuales el proceso no tiene y necesita

cubrir. Así, el sistema muestra el porcentaje cubierto por el proceso respecto al SWEBOK y las clases de SWEBOK que el proceso no cubre en su definición. El proceso de la compañía en el actual caso de estudio cubre el 12.5% del área de conocimiento de pruebas de SWEBOK, además se hace la verificación de si el proceso contiene entradas y salidas. Brindando así, nuevas oportunidades de mejora. Los resultados se muestran en la Figura 4.

5. Conclusiones

En el presente trabajo se brindó la construcción de la ontología de SWEBOK y el desarrollo del sistema software. El caso de estudio ha sido probado específicamente para el proceso de pruebas de la empresa. Para probar el proceso se empleó el área de conocimiento de pruebas de software de la Ontología de SWEBOK en conjunto con el sistema software (propuestos en el presente documento). La ontología de SWEBOK fue probada mediante un caso de estudio dentro de una compañía de consultoría en tecnologías de información, principalmente enfocada en el desarrollo de proyectos web, multimedia y aplicaciones móviles. La situación actual de la compañía fue recabada mediante entrevistas.

La introducción de la ontología de SWEBOK ayudó a la compañía a empezar una definición formal de este proceso y su mejora continua. Con el fin de alcanzar la mejora de proceso de pruebas en la compañía, fue necesario plantear preguntas de competencia específicas a este caso. Como se mencionó anteriormente, la ontología SWEBOK brinda información acerca de en qué punto se encuentra la definición del proceso. La mejora de definición de proceso es logrado por medio de satisfacer toda el área de conocimiento de pruebas de software de SWEBOK. El proceso de la compañía en el actual caso de estudio cubre el 12.5% del área de conocimiento de pruebas de SWEBOK,

además se hace la verificación de si el proceso contiene entradas y salidas. Brindando así, nuevas oportunidades de mejora.

Como se adelantó en la sección anterior con la integración de la herramienta y OWLDB, se logró almacenar la ontología de SWEBOK de manera persistente dentro de una base de datos relacional MySQL. OWLDB permite el almacenamiento de todo el esquema del modelo ontológico. Desde el manejador de MySQL, phpMyAdmin, se constató que se ha guardado todo el esquema dentro de 65 tablas, algunas de las tablas son mostradas en la Figura 5. Sin embargo los resultados no fueron del todo satisfactorios. Se logró realizar el degradado con las librerías, pero se sacrificaron funcionalidades dentro de la herramienta, por lo que significa un retroceso al trabajo. OWLDB emplea Hibernate para construir y poblar las tablas automáticamente a la base de datos. Sin embargo este es un proceso que toma bastante tiempo. En pruebas realizadas (por medio de NetBeans 7.1.2 en una PC con 3.48 GB de RAM, Intel Core i5-2310 a 2.90 Ghz y Sistema Operativo de 64 bits) el almacenamiento de la ontología SWEBOK tardó 6 min. 35 seg. y 6 seg. de carga desde la BD a el sistema.

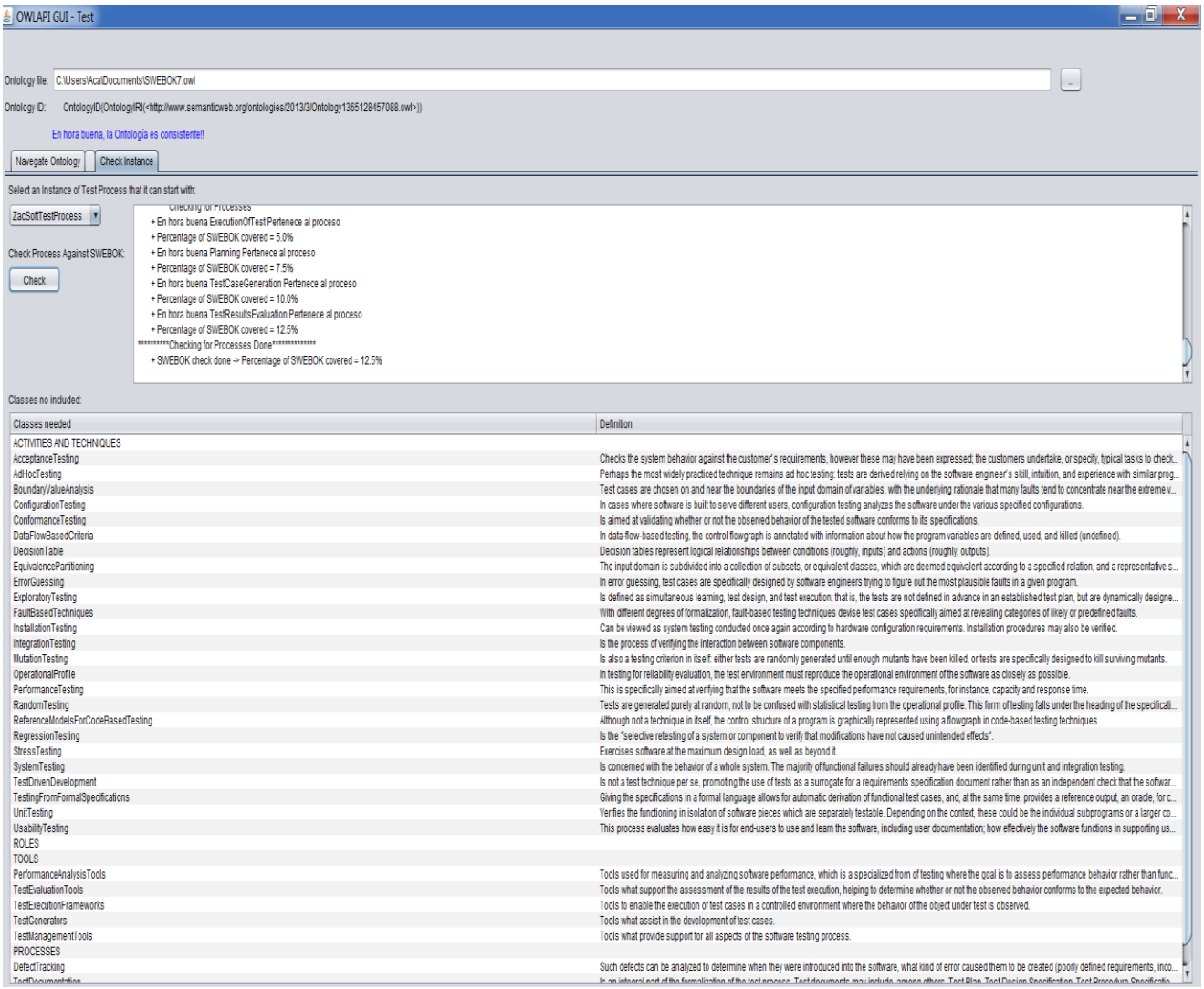

Figura 4. *Sistema con la Ontología de SWEBOK mostrando los resultados del caso de estudio.*

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
annotationassertion	Examinar Estructura Buscar Insertar Vaciar Eliminar	409	InnoDB	latin1_swedish_ci	112 KB	-
annotationiri	Examinar Estructura Buscar Insertar Vaciar Eliminar	408	InnoDB	latin1_swedish_ci	96 KB	-
annotationproperty	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	latin1_swedish_ci	32 KB	-
annotationpropertydomain	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
annotationpropertyrange	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
annotation_annotations	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
classassertionaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	20	InnoDB	latin1_swedish_ci	64 KB	-
class_expressions	Examinar Estructura Buscar Insertar Vaciar Eliminar	454	InnoDB	latin1_swedish_ci	80 KB	-
class_operands	Examinar Estructura Buscar Insertar Vaciar Eliminar	441	InnoDB	latin1_swedish_ci	48 KB	-
dataoneof_values	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
datapropertycharacteristicaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
datarange_facets	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
declarationaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	941	InnoDB	latin1_swedish_ci	176 KB	-
disjointunionaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
facetrestriction	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
haskeyaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
importsdeclaration	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	16 KB	-
individualrelationshipaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	42	InnoDB	latin1_swedish_ci	80 KB	-
indi_parts	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	latin1_swedish_ci	48 KB	-
inverseobjectpropertiesaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	latin1_swedish_ci	64 KB	-
key_prop_exprs	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
objectoneof_values	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
objectpropertychainsubpropertyaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
objectpropertycharacteristicaxiom	Examinar Estructura Buscar Insertar Vaciar Eliminar	411	InnoDB	latin1_swedish_ci	48 KB	-
obpropchainsub_parts	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
ontologyiri	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	latin1_swedish_ci	16 KB	-
owlannotation	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	64 KB	-
owlanonymousindividual	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
owlcardinalityrestriction	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	latin1_swedish_ci	48 KB	-
owlclass	Examinar Estructura Buscar Insertar Vaciar Eliminar	418	InnoDB	latin1_swedish_ci	176 KB	-
owlconstant	Examinar Estructura Buscar Insertar Vaciar Eliminar	401	InnoDB	latin1_swedish_ci	128 KB	-
owldatacomplementof	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
owldataproperty	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	64 KB	-
owldatatype	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	latin1_swedish_ci	48 KB	-
owldatatypedefinition	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	64 KB	-
owldatatyperestriction	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48 KB	-
owlnamedindividual	Examinar Estructura Buscar Insertar Vaciar Eliminar	20	InnoDB	latin1_swedish_ci	48 KB	-

Figura 5. *Base de Datos de la Ontología de SWEBOK.*

6. Trabajo futuro

Actualmente se encuentra en proceso la investigación de un marco o framework llamado Ontop que permite mitigar los problemas presentados por OWLDB en almacenamiento en bases de datos. Al contrario de OWLDB, Ontop es un framework actual y permite consultas ágiles en la base de datos. Las pruebas con este framework han sido exitosas; de igual manera las pruebas con Jena y almacenamiento persistente también han sido exitosas. Por otra parte, también se encuentra en actual desarrollo la unión entre ontologías de SWEBOK, CMMI-DEV y Six Sigma. En un futuro se planea adaptar las librerías de Ontop y la unión de ontologías para adecuarlas a las necesidades de mejora de procesos de software por medio del sistema mostrado en el presente documento.

Referencias

Abran, A., Bourque, P., Dupuis, R., Moore, J., Tripp, L. (2004). Guide to the Software Engineering Body of Knowledge (SWEBOK). IEEE Press, Piscataway, NJ.

Abran, A., Cuadrado, J., García, E., Mendes, O., Sánchez, S. & Sicilia, M. (2006). Engineering the Ontology for Software Engineering Body of Knowledge: Issues and Techinques. En Ontologies for Software Engineering and Software Technology (pp. 103-121).

Aksoy, C., Alparslan, E., Bozdağ, S., Çulhacı., Í. (2011). OSDBQ: Ontology Supported RDBMS Querying. Metadata and Semantic Research. Communications in Computer and Information Science, Vol. 240, (pp. 47 – 55). Berlin, Alemania: Springer-Verlag. doi: 10.1007/978-3642-24731-6_5.

Barchini, G., Álvarez, M. (2010). Dimensiones e Indicadores de la Calidad de una Ontología. Revista Avances en Sistemas e Informática, 7 (1). Consultada el 3 de Octubre de 2012, de http://www.redalyc.org/articulo.oa?id=133115523004

Bermejo, A. (2006). Ontology-based Software Engineering, Engineering Support for Autonomous Systems. Integrating Cognition+Emotion+Autonomy, European Integrated Project IST-027819.

CMMI Product Team. (2010). CMMI® for Development, Version 1.3 (CMU/SEI-2010-TR-033). Consultado de http://resources.sei.cmu.edu/asset_files/TechnicalReport/2010_005_001_15287.pdf

Corcho, O., Fernández, M., Gómez, A. & López, A. (2005). Building legal ontologies with METHONTOLOGY and WebODE. Law and the Semantic Web (pp. 142-157). doi: 10.1007/978-3-540-322553-5_9.

Cullot, N., Ghawi, R., Yétongnon, K. (2007). DB2OWL : A Tool for Automatic Database-to-Ontology Mapping [Versión electrónica]. SEBD. P. 491-494. Consultada el 6 de Octubre de 2012, de http://www.citeulike.org/user/stephane-jean/article/2424618

Dash, S. (2012). Ontology Driven Benchmarking in ITIL to achieve Six Sigma. International Journal of Computer Applications, (0975-8887), 42 (19). Consultada el 16 de Octubre de 2012, de http://research.ijcaonline.org/volume42/number19/pxc3877826.pdf

Ferchici, A., Bigand, M., Lefebvre, H. (2008). An Ontology for Quality Standards Integration in Software Collaborative Projects. En Proceedings of the First International Workshop on Model Driven Interoperability for Sustainable Information Systems. MDISIS'08, p. 17-30. Montpellier, France. Consultada el 16 de Octubre de 2012 de http://ceur-ws.org/Vol-340/paper02.pdf

Heymans S., Ma, L., Anicic, D., Ma, Z., Steinmetz, N., Pan, Y., Mei, J., Fokoue, A., Kalyanpur, A., Kershenbaum, A., Schonberg, E., Srinivas, K., Feier, C., Hench, G., Wetzstein, B., Keller, U. (2008). Ontology Reasoning with Large Data Repositories. Ontology Management. Computing for Human Experience, Vol. 7, (pp. 89-128). Berlin, Alemania: Springer-Verlag. doi: 10.1007/978-0-387-69900-4_4.

Kim, N., Andrews P., Asselbergs, F., Frost, H., Williams, S., Harris, B., Read, C., Askland, K. & Moore, J. (2012). Gene Ontology analysis of pairwise genetic associations in two genome-wide studies of sporadic ALS. BioData Mining 9(5).

Mendes, O. & Abran, A. (2004). Software Engineering Ontology: A Development Methodology. Metrics News 9(1) p. 64-71.

Muñoz, E., Espuña, A. & Puigjanier, L. (2010). Towards an Ontological Infraestructure for Chemical Batch Process Management. Computers & Chemical Engineering, 34 (5), p. 668-682.

Muñoz, M, Mejía, J. & Muñoz, E. (2013). Knowledge Management to Support using Multi-model Environments in Software Process Improvement. En 20th EuroSPI Conference. Dundalk Institute of Technology, Ireland.

Ning, J., Chen, Z. & Liu, G. (2010). PDCA Process Application in the Continuous Improvement of Software Quality. College of Computer Science and Engineering, Changchun University of Technology, p. 61–65, 201.

Pyzdek, T. (2003). The Six Sigma Handbook. A Complete Guide for Green Belts, Black Belts, and Managers at All Levels. [Adobe Digital Editions version]. doi: 10.1036/0071415963.

Quintanilla, A. (2005). Tecnología: In enfoque Filosófico y Otros Ensayos de Filosofía de la Tecnología (1ª Edición). México, D.F. Fondo de Cultura Económica, p. 65.

Romá, M. (2009). Tesis Doctoral. OntoFIS: Tecnología Ontología en el dominio farmacoterapéutico [Versión Electrónica]. Tesis Doctoral. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante. Consultada el 6 de Octubre de 2012, de http://gplsi.dlsi.ua.es/gplsi11/sites/default/files/tesis_Roma_OntoFIS.pdf

Soydan, G. & Kokar, M. (2012). A Partial Formalization of the CMMI-DEV – a Capability Maturity Model for Development. Journal of Software Engineering and Applications, 5 (10), p. 1-25.

Sure, Y., Staab, S. & Studer, R. (2002). Methodology for development and employment of ontology based knowledge management application. ACM Sigmod Record 31 (4) p. 18-23. doi: 10.1145/637411.637414.

Trinkunas, J. & Vasilecas, O. (2007). Building Ontologies from Relational Databases using Reverse Engineering Methods. En Rachev B., Smrikarov, A. & Dimov, D. (Eds.), Proceedings CompSysTech '07 Proceedings of the 2007 international conference on Computer systems and technologies. Bulgaria. doi: 10.1145/1330598.1330614.

Wille, C., Abran, A., Desharnais, J. & Dumke, R. (2003). The Quality concepts and sub-concepts in SWEBOK: An Ontology challenge. En International Workshop on Software Measurement (IWSM), Montreal, Canadá, p. 18.

Wille, C., Abran, A., Desharnais, J. & Dumke, R. (2004). E-Learning Infrastructure for Software Engineering Education: Steps in Ontology Modeling for SWEBOK. En Proceedings of the IASTED International Conference SOFTWARE ENGINEERING, Innsbruck, Austria.

Zhichun, Q. (2011). Quality improvement of wall energy conservation project based on PDCA cycle, 2011 International Conference on Electric Technology and Civil Engineering (ICETCE), p. 1416–1419.

Notas biográficas:

José Eduardo Guadalupe Gaytán Solís es actual estudiante de la maestría en ingeniería de software en el Centro de Investigación en Matemáticas A.C. (CIMAT), unidad Zacatecas, en Zacatecas, México. Obtuvo el grado de Ingeniero en Computación en el año 2012 en la Universidad Autónoma de Zacatecas. Cuenta con las certificaciones: Microsoft MCTS .NET Framework 2.0, Windows Development Foundation, Microsoft MCTS .NET Framework 3.5, ASP .NET Applications, Oracle Java Standard Edition 6 Programmer, Novel Certified Linux Desktop Administrator SUSE Linux Enterprise Desktop 10 y SEI-Certified PSP Developer. Sus intereses son programación de aplicaciones web y de escritorio, inteligencia artificial y arquitectura de software. En deportes: Fútbol Soccer, Fútbol Americano, Atletismo.

Ricardo González Saldívar es actual estudiante de la Maestría en Ingeniería de Software en el Centro de Investigación en Matemáticas A.C. (CIMAT), unidad Zacatecas, en Zacatecas, México. Obtuvo el grado de Ingeniero en Sistemas Computacionales en el año 2010 en el Instituto Tecnológico Superior de Nochistlán. Cuenta con las certificaciones: Sun Certified Java SE Developer y SEI-Certified PSP Developer. Sus intereses son Arquitectura y Diseño de Software, Procesos de Desarrollo de Software, Ciencias de la Computación y Gestión del Conocimiento. Le gusta practicar ajedrez, ilustración digital y la lectura.

Edrisi Muñoz Mata Ingeniero industrial con especialidad en manufactura y Maestro en ciencias en ingeniaría industrial con especialidad en calidad por el Instituto tecnológico de Orizaba (ITO) de México. Doctor en filosofía en ingeniaría de procesos químicos por la Universidad Politécnica (UPC) de Cataluña de España. Su área de investigación principal es la gestión del conocimiento mediante el desarrollo de modelos ontológicos, sistemas de soporte a las decisiones en distintas áreas de proceso y optimización de procesos mediante el uso de modelos analíticos rigurosos. Actualmente es investigador asociado del Centro de Investigación en Matemáticas A.C. (CIMAT) de México, así como investigador invitado en el Centro de procesos y medio ambiente en la UPC. Su participación compete diferentes proyectos de investigación mexicanos y europeos. Participa en la publicación de diferentes artículos en revistas internacionales indexadas, así como en distintos congresos internacionales de renombre. Última publicación: Edrisi Muñoz, Elisabet Capón-García, José Miguel Laínez, Antonio Espuña, Luis Puigjaner, Integration of enterprise levels based on an ontological framework, Chemical Engineering Research and Design, Volume 91, Issue 8, August 2013, Pages 1542-1556, ISSN 0263-8762.