

Tipo de artículo: Artículo de revisión
Temática: Ingeniería y gestión de software
Recibido: 20/09/19 | Aceptado: 04/01/2020 | Publicado: 06/01/2020

Propuesta para el desarrollo de una ontología en la realización de pruebas de software.

Proposal for the development of an ontology in the realization of software tests.

Amanda Valdivieso Martínez^{1*}

¹ Dirección de Calidad de Software. Universidad de Ciencias Informáticas, Carretera a San Antonio de los Baños, Km 2 ½, Torrens, Boyeros. La Habana. Cuba. avaldivieso@estudiantes.uci.cu

* Autor para correspondencia: avaldivieso@estudiantes.uci.cu

Resumen

Para desarrollar un software con la calidad requerida, se tienen en cuenta una serie de etapas por las que debe transitar antes de ser entregado al cliente. Para velar por el correcto desarrollo del sistema en cada etapa en la que se esté trabajando es esencial la realización de las pruebas de software. En la Dirección de Calidad de la Universidad de las Ciencias Informáticas se ejecutan pruebas a nivel de gerencia y pruebas de liberación del producto, donde la cantidad de personal involucrado de diferentes entidades tiene distintos niveles de conocimiento y se generan grandes volúmenes de documentación. Todo ello, provoca dificultades en el entendimiento y comprensión de varios de los términos y conceptos utilizados. No se gestiona el conocimiento generado durante la ejecución de las pruebas y hay principios y métodos que aún están siendo definidos y consolidados. El objetivo del presente trabajo es realizar una investigación sobre las ontologías relacionadas con la calidad de software más específicamente sobre las Pruebas de software.

Palabras clave: Calidad, conocimiento, ontología, pruebas.

Abstract

To develop a software with the required quality, a series of stages through which it must go before being delivered to the client are taken into account. To ensure the correct development of the system at each stage in which it is working, it is essential to perform the software tests. In the Quality Directorate of the University of Computer Science, tests are carried out at the management level and product release tests, where the number of personnel involved from different entities has different levels of knowledge and large volumes of documentation are generated. All this causes difficulties in understanding and understanding several of the terms and concepts used. The knowledge generated during the execution of the tests is not managed and there are principles and methods that are still being defined and consolidated. The objective of this work is to develop an Ontology that allows organizing and representing the knowledge generated in the quality laboratory, thus achieving an effective recovery of information, based on the structuring of knowledge.

Keywords: Information, knowledge, Quality, ontology, tests.

Introducción

El análisis realizado a varias investigaciones relacionadas con el desarrollo de ontologías en el área de calidad (Duarte, et al., 2000, Ferreira, 2006, Echeverría Pérez, et al., 2011, Echeverría Pérez, et al., 2012, Souza, 2013) arrojó un conjunto de ontologías aplicadas al dominio de calidad de software. Las mismas, son incompletas, generalizadoras de este dominio, no abordan todos los términos o las relaciones surgidas en la Dirección de Calidad, específicamente las que abarcan las pruebas de software.

La ejecución de las pruebas a nivel de proyectos, de centro y de gerencia, son complejas por la cantidad de personal involucrado de diferentes áreas de la Universidad con diversos niveles de conocimiento, además influye también el volumen de documentación generada. Todo ello, provoca dificultades en el entendimiento y comprensión total de los subprocesos, fundamentalmente de los términos y conceptos utilizados. No existe consenso sobre la terminología utilizada y no se gestiona el conocimiento generado durante la ejecución de los subprocesos. Hay principios y métodos que aún están siendo definidos, consolidados y mejorados. Por otra parte, constantemente se identifican e incorporan estándares, normas, métodos y modelos internacionalmente reconocidos y no siempre se registran y se tienen al alcance de los usuarios de los procesos para su comprensión y uso. Así mismo estas propuestas que guían y orientan los aspectos fundamentales de estos subprocesos, en algunos casos son ambiguos y solapados, y presentan

disímiles significados para un mismo término. Pueden encontrarse inconsistencias y conflictos de terminología entre estándares de diferentes organizaciones e incluso en los de una misma organización (ISO/IEC/IEEE. 2017). Lo mismo sucede con los modelos de calidad. La NC ISO/IEC 25000 (NC ISO/IEC 25000 2011), contiene una visión general sobre la calidad de productos de software, pero hay aspectos, como la medición, que se complementan con otros estándares y ofrecen vistas parciales de este dominio. Las pruebas de software, son escasamente tratadas. Todo lo anterior, conlleva a la búsqueda de técnicas y herramientas que proporcionen un vocabulario común para resolver el problema de integridad e inconsistencia, identificado en los diversos estándares y en el personal involucrado en la Dirección de Calidad de la UCI, con el objetivo de reducir la pérdida o desaprovechamiento del conocimiento.

Después de analizar diferentes variantes, se constató las potencialidades para apoyar la toma de decisiones en el subproceso de pruebas de software, teniendo como principal objetivo organizar y comunicar el conocimiento acumulado. Por tanto, en aras de mitigar estas dificultades surge la presente investigación, que tiene como objetivo desarrollar una ontología que apoye el subproceso de pruebas de software en la UCI, permitiendo la consistencia, organización y comunicación del conocimiento existente y generado.

Materiales y métodos o Metodología computacional

Para desarrollar la ontología se pretende utilizar como metodología METONTHOLGY, pues permite llevar un orden de las actividades a desarrollar y conceptualiza la ontología modelando sus componentes, como son la taxonomía de conceptos, las relaciones entre ellos, las clases. Como lenguaje Ontology Web Language (OWL) este, permite publicar y compartir datos mediante el uso de ontologías, usa Resource Description Framework (RDF). Es un lenguaje diseñado para representar conocimiento complejo acerca de cosas, grupos de cosas y relaciones entre las cosas. Está basado en la lógica matemática y por tanto es procesable automáticamente por las computadoras. Como herramienta de modelado para la construcción y creación de las clases, propiedades, instancias, axiomas de la ontología se utilizó Protégé en su versión 5.5.0-beta-4, debido a la transparencia de su trabajo con OWL. A pesar de tener su propio lenguaje interno para definir ontologías, cuenta con un entorno abierto y fácil de entender, que ha generado en torno suyo toda una comunidad que contribuye activamente a ampliar el entorno con todo tipo de contribuciones en forma de plugins y está haciendo de esta herramienta un entorno sumamente potente. Además, es la herramienta más empleada para el trabajo con la metodología Methontology y permite integrarse con otras herramientas que permiten visualizar el contenido de las ontologías y realizar validaciones a estas.

Tipos de Ontología

Existen diferentes criterios para clasificar ontologías en dependencia de la intención de uso o tarea en particular, la formalidad del lenguaje de construcción utilizado, la generalidad, entre otros (Ramos, et al., 2007)

Las ontologías se clasifican por su dependencia del contexto en:

- Ontologías de Nivel Superior: Describen conceptos muy generales que son independientes de un problema particular o dominio. Por ejemplo, ontologías sobre el tiempo, el espacio, la materia, el objeto, el acontecimiento, la acción, entre otros.
- Ontologías de Dominio: Proporcionan el vocabulario necesario para describir un dominio dado. Incluyen términos relacionados con los objetos del dominio y sus componentes, un conjunto de verbos o frases que dan nombre a actividades y procesos que tienen lugar en ese dominio, y conceptos primitivos que aparecen en teorías, relaciones y fórmulas que regulan o rigen el dominio.
- Ontologías de Tareas: Proveen un vocabulario sistemático de los términos usados para resolver problemas asociados con tareas particulares, ya sean dependientes o no del dominio.
- Ontologías de Aplicación: Contienen las definiciones necesarias para modelar el conocimiento requerido para una aplicación particular en un dominio dado. Describen conceptos que dependen tanto del dominio particular como de las tareas. Estas ontologías son especializaciones de las ontologías de dominio y de tareas.

Metodologías para el desarrollo de Ontologías

En el proceso de construcción de ontologías es conveniente emplear una adecuada metodología, siguiendo los procesos de desarrollo y las actividades y técnicas indicadas. Las metodologías proporcionan un conjunto de directrices que sugieren cómo hay que llevar a cabo las actividades identificadas en el proceso de desarrollo, qué técnicas son las más oportunas en cada actividad y qué produce cada una de ellas. Algunas de estas metodologías (Ramos, et al., 2007) son:

Methontology

Methontology es una metodología recomendada por Fundación de Agentes Físicos Inteligentes (FIPA) usada en el proceso de desarrollo de una ontología, la cual tiene sus raíces en las actividades identificadas por la Instituto de

Ingenieros Eléctricos y Electrónicos (IEEE) para el proceso de desarrollo de software. Esta metodología está basada en prototipos evolutivos, habilitando añadir, cambiar y eliminar términos en cada nueva versión de la ontología

El ciclo de vida de Methontology está compuesto por varias actividades, entre ellas, las actividades de control, aseguramiento de calidad, adquisición de conocimiento, integración, evaluación, documentación y manejo de configuración. Estas actividades se realizan simultáneamente con las actividades de desarrollo, como son: especificación, conceptualización, formalización, implementación y mantenimiento.

Pasos de los prototipos evolutivos propuestos por Methontology:

- Especificación: en este paso se delimitan los objetivos para la creación de la ontología (ej. compartir información entre personas o por agentes software, permitir la reutilización del conocimiento de un dominio, hacer explícitas las suposiciones que se efectúan en un dominio, separar el conocimiento del dominio del conocimiento sobre su fabricación o forma de operación o simplemente analizar el conocimiento del dominio), se decide el dominio de actuación de la misma (para no modelar objetos poco relevantes en perjuicio de otros más importantes), quién la usará y para qué, las preguntas a las que deberá responder (ayuda al establecimiento de las dos anteriores) y quién se encargará de su mantenimiento (decidiendo si se limitará a introducir nuevas instancias, se permitirá la modificación de conceptos o atributos).
- Conceptualización: consiste en crear un glosario de términos que pertenecen al dominio, definirlos y crear una taxonomía (estableciendo una clasificación o jerarquía entre los conceptos, sus niveles, las relaciones entre ellos, sus instancias, sus propiedades o atributos, e igualmente los axiomas o reglas).
- Formalización: proceso que consiste en convertir el modelo anterior en un modelo formal o semi-computable.
- Implementación: convierte el modelo formalizado en un modelo computable mediante un lenguaje para construcción de ontologías.
- Mantenimiento: labor que puede acarrear desde el borrado de instancias ya inútiles o la incorporación de nuevas instancias que se han ido produciendo con el tiempo, hasta las tareas de introducción de cambios en el contenido de la información, ya sea redefiniendo atributos, relaciones o incluso conceptos.

On-To-Knowledge

Aparece alrededor de 1999, esta metodología está orientada a procesos, se basa en el análisis de escenarios de uso y considera la identificación de los objetivos y metas que deberían alcanzar las aplicaciones. On-To-Knowledge

gestiona el conocimiento de grandes organizaciones. Las fases de esta metodología son estudio de factibilidad, comienzo, refinamiento, evaluación y mantenimiento.

Terminae

Terminae: para el 2002, aparece esta metodología, que aporta una herramienta para la construcción de ontologías a partir de textos. Se basa en un análisis lingüístico de los textos, el cual se realiza mediante la aplicación de diferentes herramientas para el procesamiento del lenguaje natural. En particular se usan dos herramientas: Syntex para identificar términos y relaciones; y Caméléon para identificar roles o relaciones. La metodología funciona como sigue. Mediante la aplicación de Syntex obtenemos una lista de posibles palabras y frases del texto y algunas dependencias sintácticas y gramaticales entre ellas. Estos datos se usan como entrada para el proceso de modelado junto con el texto original.

Competency Questions

Competency Questions: preguntas relevantes o preguntas de verificación; es una de las metodologías más sencillas de aplicar que existe. Determina el dominio y alcance de la ontología mediante lista de preguntas que el sistema debería ser capaz de contestar. La respuesta a estas preguntas sugiere lo que podría ser las instancias de la ontología

De las metodologías estudiadas es Methontology la cual cuenta con una serie de características que la hacen superior a las otras estudias.

- No está inspirada en experiencias concretas, ha sido empleada para modelar ontologías de diversos dominios.
- Propone un ciclo de vida iterativo y basado en prototipos evolutivos.
- Propone actividades específicas a desarrollar.
- Se explican bien los pasos a seguir para el desarrollo de una ontología.
- Se proponen elementos para el cómo desarrollarla y no sólo el qué hacer.
- Varios softwares existentes para la implementación de ontologías le dan soporte a la misma.
- Cuenta con una amplia documentación que posibilita conocer mejor los procesos que en ella ocurren.
- Independiente de las aplicaciones en las que se vayan a utilizar.

Lenguajes para representar ontologías

Existen numerosos lenguajes para representar ontologías aquí explicaremos algunos de ellos como:

LOOM

Lenguaje basado en lógica de descripciones y reglas de producción. Con el auge de Internet surgen lenguajes de Ontologías para la web con sintaxis basada en HTML o XML.

SHOE

SHOE: Simple HTML Ontology Extensions. Fue el primer lenguaje de etiquetado para diseñar ontologías en la Web. Este lenguaje nació antes de que se ideara la Web Semántica. Las ontologías y las etiquetas se incrustaban en archivos HTML. Este lenguaje permite definir clases y reglas de inferencia, pero no negaciones o disyunciones.

RDF

Más tarde aparece ResourceDescription Framework (RDF) como sucesor del DAML+OIL basado en una tripleta que define (sujeto, predicado, objeto) donde el sujeto es el recurso en sí, el predicado representa las características del sujeto, y los objetos son los valores que se le asignan a las propiedades y mediante el cual se establecen relaciones.

RDF Schema: proporciona un vocabulario para el modelado de datos en RDF. Útil para la creación de ontologías ligeras.

XML

En principio, se trabajó con XML (extensible Markup Language) que es independiente de la plataforma y proporciona una estructura sintáctica que las computadoras pueden interpretar. Es un lenguaje para definir lenguajes. Fue diseñado para proporcionar un formato para intercambiar definiciones de ontologías y con la semántica de los sistemas de representación del conocimiento orientados a objetos. Por lo tanto, no está pensado para ser utilizado para el desarrollo de ontologías, sino como un lenguaje intermedio para la transferencia entre diversos sistemas.

Luego se utilizó XMLS (XML Schema) permitiendo la definición de gramáticas y etiquetas significativas a los documentos, con este aparece una semántica intuitiva para los humanos, pero no para las computadoras. Extensión de XML usado como lenguaje de definición tipos de datos.

OWL

En la actualidad el lenguaje que predomina en el mundo es el Ontology Web Language (OWL) que permite publicar y compartir datos mediante el uso de ontologías, usa RDF. Es un lenguaje diseñado para representar conocimiento complejo acerca de cosas, grupos de cosas y relaciones entre las cosas. Está basado en la lógica matemática y por tanto es procesable automáticamente por las computadoras. Posee tres variantes:

- OWL Lite que solo utiliza restricciones simples y una cardinalidad de 0 a 1.
- OWL DL permite una máxima expresividad sin perder el uso de algoritmos de razonamiento.
- OWL Full que mantiene su compatibilidad con RDF y posibilita el aumento del vocabulario definido anteriormente.

OWL 2: es la segunda versión de OWL que incorpora mejoras sustanciales con respecto a su predecesora OWL.

Posee tres perfiles:

- OWL 2 EL: grandes volúmenes de propiedades y clases.
- OWL 2 QL: grandes volúmenes de instancias.
- OWL 2 RL: aplicaciones que requieren razonamiento escalable sin sacrificar mucho poder de expresividad.

OIL

OIL: Ontology Inference Layer. Este lenguaje, derivado en parte de SHOE, fue impulsado también por el proyecto de la Unión Europea On-To-Knowledge. Utiliza ya la sintaxis del lenguaje XML y está definido como una extensión de RDFS. Se basa tanto en la lógica descriptiva (declaración de axiomas) y en los sistemas basados en frames (taxonomías de clases y atributos). OIL posee varias capas de sub-lenguajes, entre ellas destaca la capa base que es RDFS, a la que cada una de las capas subsiguientes añade alguna funcionalidad y mayor complejidad. La principal carencia de este lenguaje es la falta de expresividad para declarar axiomas.

TURTLE

TURTLE: (Terse RDF Triple Language) es un formato para la expresión de los datos en el Resource Description Framework (RDF). Fue la primera notación aceptada por la W3C, en agosto de 2011. Hoy en día es probablemente la notación RDF más extendida en la web semántica. Utiliza definición de prefijos para aumentar la legibilidad.

Herramientas para la creación de ontologías.

Las herramientas son imprescindibles al momento de crear una ontología y escoger la correcta dependiendo de qué tipo sea la que vas a desarrollar facilita el proceso de creación. Aquí le explicaremos algunas de ellas:

KAON

Es una infraestructura de gestión de ontología de código abierto orientada a aplicaciones de negocios. Incluye una completa suite de herramientas que permite la creación de la ontología y administrar con facilidad, proporciona un marco para la creación de aplicaciones basadas en ontologías. Un aspecto importante de KAON es su razonamiento

escalable y eficiente. KAON3 consiste en una serie de módulos diferentes que proporcionan una amplia gama de funcionalidades entorno a la creación, almacenamiento, recuperación, mantenimiento y aplicación de ontologías. Fue desarrollada en un esfuerzo conjunto, principalmente por miembros de Instituto de la Universidad de Karlsruhe (AIFB) y el Centro de Investigación en Tecnologías de la Información.

OntoEdit

Es una herramienta de edición de ontologías que apoya el desarrollo y mantenimiento de las mismas utilizando medios gráficos en un entorno web. Permite la representación semántica de lenguajes conceptuales y estructuras mediante conceptos, jerarquías de conceptos, relaciones y axiomas.

La interfaz es abierta y permite a los usuarios ajustarlo a sus necesidades. Además, tiene disponibles una gran variedad de funcionalidades fáciles de usar y adaptables a los diferentes escenarios.

Ofrece una herramienta para representar gráficamente ontologías y que, además, pueda almacenarlas y posteriormente manipularlas en una base de datos relacional.

Otros editores de ontologías se centran en ofrecer interfaces gráficas ricas en funcionalidades para editar, integrar y traducir ontologías; pero como estas herramientas no poseen integración en una base de datos, las ontologías no pueden ser guardadas, gestionadas y manipuladas directamente por el editor como si fuera un Sistema Gestor de Base de Datos

Aunque OntoEdit está pensado para un entorno web, también puede aplicarse a una Intranet. De esta forma se preserva la seguridad de la información que contiene.

Protégé

La primera herramienta Protégé fue creada con el principal objetivo de simplificar el proceso de adquisición de conocimiento para sistemas expertos. Para lograr este objetivo, se utilizan los conocimientos adquiridos en etapas anteriores del proceso para generar formularios personalizados para la adquisición de más conocimiento. Desde entonces, Protégé ha pasado por varias versiones y se ha centrado en diferentes aspectos de la adquisición de conocimiento (bases de conocimiento, los métodos de resolución de problemas, ontologías).

Protégé se basa en los marcos y la lógica de primer orden. Es compatible con Open Knowledge Base Connectivity (OKBC), lo que significa que los componentes principales de modelado son las clases, los slots, las facetas e instancias. Las clases están organizadas en jerarquías de clases donde se permite la herencia múltiple y los slots también pueden ser organizados en jerarquías.

Protégé reconoce marcos, XML Schema, RDF Schema y OWL, que son lenguajes semánticos utilizados en la Web, en contraposición a la rigidez del HTML.

Conclusiones

Podemos concluir que con el uso de una ontología la información queda estructurada a través de conceptos y relaciones semánticas, de forma tal que pueda ser accedida y manipulada por personas o software. Analizar los términos y relaciones que lo configuran, ya sea formalmente o no. Compartir el entendimiento común de la estructura de información entre personas o agentes de software y permitir la reutilización de conocimiento de un dominio. Para esta investigación se desea desarrollar una ontología de dominio. La misma debe ser capaz de representar el dominio relacionado con el proceso de prueba de software que se desarrolla en la Dirección de Calidad de La Universidad de Ciencias Informáticas.

Se utilizó como metodología para su creación Methontology. Esta metodología es la más usada en la construcción de ontologías, pues permite llevar un orden de las actividades a desarrollar y conceptualiza la ontología modelando sus componentes. El lenguaje que utiliza es el OWL, este permite publicar y compartir datos mediante el uso de ontologías, usa RDF. Es un lenguaje diseñado para representar conocimiento complejo acerca de cosas, grupos de cosas y relaciones entre las cosas. Está basado en la lógica matemática y por tanto es procesable automáticamente por las computadoras.

Dentro de las herramientas que dan soporte a la creación de ontologías la más utilizada y que más se ajusta a las necesidades de la investigación es Protégé debido a la transparencia de su trabajo con OWL. A pesar de tener su propio lenguaje interno para definir ontologías, cuenta con un entorno abierto y fácil de entender, que ha generado en torno suyo toda una comunidad que contribuye activamente a ampliar el entorno con todo tipo de contribuciones en forma de plugins y está haciendo de esta herramienta un entorno sumamente potente. Además, es la herramienta más empleada para el trabajo con la metodología Methontology y permite integrarse con otras herramientas que permiten visualizar el contenido de las ontologías y realizar validaciones a estas. Estas serían las bases para la creación de la ontología para las pruebas de software. Podemos afirmar que las ontologías son un arma importante a la hora de gestionar la información de una empresa o de un proyecto determinado.

Referencias

Duarte, K.C y Almeida Falbo, R. 2000. . *Uma Ontologia de Qualidade de Software.* . Vitória- Espírito Santo : Repositorio Institucional de Universidade Federal do Espírito Santo., 2000.

Grupo Editorial “Ediciones Futuro”
Universidad de las Ciencias Informáticas. La Habana, Cuba
seriecienfifica@uci.cu

Echeverría Pérez, D y Fernández Pérez, Y. 2011. *Desarrollo de una ontología de apoyo al procedimiento del Departamento de Pruebas de Software. UCI. La Habana : s.n., 2011.*

Echeverría Pérez, D, Fernández Pérez, Y y Zulueta Pozo, D. 2012. *Ontología de apoyo al procedimiento de pruebas del Departamento de Pruebas de Software de Calisoft. In Tenth LACCEI Latin American and Caribbean Conference (LACCEI'2012). 2012.*

Ferreira, M. 2006. *Medición del Software Ontología y Metamodelo. La Mancha. : Informe Técnico Universidad de Castilla, 2006.*

Ramos, E y Núñez, H. 2007. *ONTOLOGÍAS: componentes, metodologías, lenguajes, herramientas y aplicaciones. Caracas. : s.n., 2007.*

Souza, E.F. 2013. *Using Ontology Patterns for Building a Reference Software Testing Ontology. In 17th IEEE International Enterprise Distributed Object Computing Conference Workshops. pp. 21–30. 2013.*