Programación Matemática y Software (2013) 5 (2): 62-70. ISSN: 2007-3283

Herramienta para generar contratos de desarrollo de software en entornos de *outsourcing* basado en una ontología

Generating software development contracts in outsourcing environments based on an ontological tool

Sergio David Ixmatlahua-Díaz,[1]* Jezreel Mejía-Miranda,[2] Mirna Ariadna Muñoz-Mata,[2]
Roque Manuel Rueda-Anastacio,[1] Emmanuel Mendoza-Escobar,[1] Alejandro D. Velázquez Cruz[3]

[1] Departamento de Posgrado e Investigación, Instituto Tecnológico de Orizaba.
Av. Instituto Tecnológico 852, col. Emiliano Zapata, Orizaba, Veracruz. México
[2] Centro de Investigación en Matemáticas, Unidad Zacatecas.
Av. Universidad 222, col. Hidráulica, Zacatecas, Zacatecas, México.
[3] Sensa Control Digital, Av. Bravo Oriente 93, col. Centro, CP 27000, Torreón, Coahuila. México.
* Correo-e: davi3021@acm.org

PAIABRAS CIAVE:

outsourcing, desarrollo de software, web, modelo de contrato, arquitectura, ontología.

RESUMEN

Actualmente el *outsourcing* es una de las actividades principales de trabajo para las organizaciones de desarrollo de software. Sin embargo, las relaciones que se dan entre un cliente y un proveedor de servicios no siempre son lo suficientemente fuertes para cumplir las expectativas de los acuerdos. El contrato de *outsourcing* para proyectos de software es una alternativa a este tipo de relaciones. En este artículo se presenta la arquitectura de la herramienta Quetzalcóatl y se propone una ontología para representar el modelo de contrato, todo esto con el objetivo de generar y evaluar contratos para proyectos de desarrollo de software en entornos de tercerización.

KEYWORDS:

outsourcing, software development, web, contract model, ontological model.

ABSTRACT

Currently outsourcing is one of the main lines of work for software development organizations. However, the relationship that exists between a client and a service provider is not always strong enough to meet the expectations of the initial agreements. The outsourcing contract for software development projects is an alternative to such relationships. This paper presents the architecture of the tool Quetzalcoatl and an ontology is proposed to represent the contract model, with the objective of generating and evaluating contracts for software development projects in outsourcing environments.

1 INTRODUCCIÓN

La evolución del *outsourcing* de tecnologías de la información se ha realizado a pasos agigantados desde los años 60 y hasta el día de hoy, convirtiéndose así, en una estrategia empresarial rentable, que va desde los grandes corporativos mundiales hasta las pymes [1]. Sin duda alguna, esto ha sido el resultado de la evolución de las estrategias de negocio como un factor determinante para el desarrollo de las empresas.

Por un lado, algunos estudios realizados por el Instituto de Ingeniería de Software (SEI) [2], muestran que entre el 20% y el 25% de los grandes proyectos de *outsourcing* fracasan dentro de los primeros años y un 50% en los primeros 5 años. Una mala gestión del proyecto, la definición equivocada de los requerimientos, la inexistencia de procesos para la selección de proveedores, la falta de un contrato específico de TI (tecnologías de la información) fundamentado en la ingeniería de software y, también, la falta de herramientas especializadas en la generación de contratos en entornos de *outsourcing*, son algunos de los factores que ocasionan estos fracasos.

Por otra parte, en México, la International Data Corporation (IDC) [3] indicó que para el año 2011 el mercado del *outsourcing* de TI crecería unos 10 Billones de dólares que representan un crecimiento del 7% con respecto al año 2010. Sin embargo, pese a que este mercado cada vez es más demandado por todo tipo de clientes, ya sean empresas transnacionales, pymes o particulares, las cuales sufren problemas de cumplimiento de las actividades parciales o totales del proyecto de desarrollo de software en este tipo de entornos. Algunos autores, como Matthew K. O. [4], indican que la definición de un contrato, sin duda alguna, es la clave para que una relación exista entre clientes y proveedores de TI. Por lo tanto, en este contexto, el contrato es un mecanismo para asegurar que las expectativas sean cumplidas, pues los contratos definen los derechos, obligaciones, responsabilidades y las expectativas que a ambas partes corresponden.

Con base en lo anterior, el objetivo de este artículo es presentar la arquitectura de la herramienta Quetzalcóatl, la cual integra el modelo de contrato definido en [5], como un modelo conceptual de la ontología que representa al tipo de contrato; esto con la finalidad de ofrecer una herramienta que apoye a las grandes organizaciones, pymes o particulares, en general, que contraten proveedores para el desarrollo de software.

Este artículo está estructurado de la siguiente manera: en la sección 2 se presenta la arquitectura de la herramienta Quetzalcóatl, así como el modelo de contrato y método de evaluación que sustenta la herramienta, un análisis de herramientas de gestión de contratos, la funcionalidad y características de la herramienta Quetzalcóatl, y por último la vista de la arquitectura propuesta para la herramienta. En la sección 3 se presenta el modelo de la ontología a un nivel conceptual, el cual se desprende del análisis realizado al modelo de contrato de [5].

2 ARQUITECTURA DE LA HERRAMIENTA: QUETZALCÓATL

Es una herramienta que tiene como objetivo proveer de elementos que ayuden a las organizaciones –grandes, pymes o particulares– a conocer los componentes primordiales que un contrato de desarrollo de software en entornos de outsourcing debe tener. Para establecer la arquitectura de la herramienta, se analizó el modelo de contrato y el método de evaluación, además de un análisis de las herramientas para la generación de contratos.

En las siguientes secciones se describen el modelo de contrato y el método de evaluación definido en [5], el cual la herramienta Quetzalcóatl toma como referencia, así como, el análisis de las herramientas existentes que ha permitido analizar las características y funcionalidades que ofrecen otras herramientas con la finalidad de mejorar la arquitectura de la herramienta.

2.1 Modelo de contrato de desarrollo de software

Cabe mencionar que la herramienta se basa en un modelo de contrato y método de evaluación definido en [5]. Por lo que, en esta sección se describe de manera breve el modelo de contrato y método de evaluación. En la figura 1 se muestra la estructura general del modelo de contrato el cual se divide en 7 categorías. Cada una de estas categorías está formada por cláusulas, que se interrelacionan con el mismo tópico (una cláusula es un conjunto de componentes relacionados entre sí, y un componente es una

actividad o información básica que debe ser incluida dentro de un contrato).

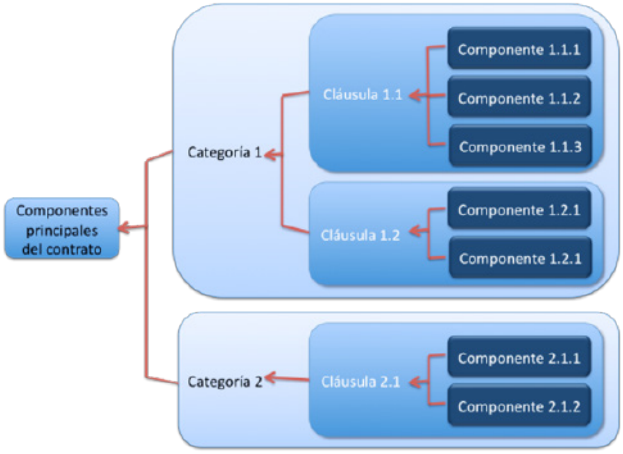


Figura 1. Estructura del modelo de contrato

A continuación se describen brevemente cada una de estas categorías, con la finalidad de conocer su objetivo particular.

- Categoría de servicios
  Las cláusulas dentro de esta categoría establecen el tipo de servicio, el alcance y la naturaleza de todos los servicios requeridos, el tiempo en que los servicios deberían estar listos y los niveles de rendimiento necesario.
- Categoría de gestión de servicios
  Las cláusulas agrupadas en esta categoría establecen y evalúan la relación entre las distintas partes implicadas para garantizar el mantenimiento de la relación.
- Categoría financiera
  En esta categoría se abordan los aspectos referentes a los acuerdos sobre precios y pagos, incluyendo incentivos y penalizaciones.
- Categoría de duración
  Las cláusulas de esta categoría ayudan a especificar las fechas acordadas durante la negociación definida al inicio del contrato, la finalización, las fechas de entrega de cada servicio y su duración.
- Categoría de transición
  Las cláusulas dentro de esta categoría hacen referencia a la elaboración de procedimientos que deben cumplirse con objeto de garantizar las condiciones actuales para la finalización o la transferencia de los activos relacionados con el servicio o producto.
- Categoría de seguridad
  El objetivo de las cláusulas agrupadas en esta categoría es dar garantía a los intereses del cliente en el caso de que se presente una finalización anticipada del contrato, proporcionándole al comprador una posición más fuerte ante una futura renegociación del contrato.
- Categoría de comunicación
  En esta categoría las cláusulas establecen las normas y procedimientos para futuras contingencias, las cuales permitirán mejores resultados.

2.2 Método de evaluación

Otra característica que la arquitectura de la herramienta Quetzalcóatl debe considerar es la capacidad de realizar evaluaciones de los contratos de acuerdo al método de evaluación que es definido en [4]. Esta evaluación puede realizarse sobre contratos creados directamente en la herramienta o contratos propiedad del usuario. El método de evaluación definido permite tener en consideración los aspectos que deberá cumplir el contrato del proveedor para satisfacer las necesidades del cliente. Este método se basa en dos criterios:

1. Tomar como base el modelo de contrato
2. Tomar como base los objetivos del negocio

Criterio 1: con base en el modelo de contrato, este método de evaluación tiene como objetivo identificar el porcentaje de cobertura de la cláusula, de la categoría y del contrato, a partir de la suma de los porcentajes alcanzados por cada componente en el Modelo de Contrato.

Criterio 2: con base en los objetivos del negocio, este método de evaluación tiene como objetivo identificar el porcentaje de cobertura de la cláusula o de la categoría, o del contrato a partir de la ponderación que se asigne a cada uno de los componentes, cláusulas y categorías del modelo, de acuerdo a los objetivos del negocio.

En otras palabras, se asigna un porcentaje que refleje el grado de importancia que tiene para el negocio dicho componente, cláusula o categoría.

2.3 Análisis de herramientas relacionadas

Con la finalidad de establecer las características y funcionalidades que la herramienta Quetzalcóatl debe incluir, en esta sección se describen, de manera breve,

las herramientas de generación de contratos que se han detectado durante la práctica de esta investigación.

En el mercado existen herramientas que ofrecen la gestión de contratos, éstas proporcionan la posibilidad de generar contratos a través de plantillas predefinidas; es decir, formatos de contratos con lineamientos muy comunes. Algunas de estas herramientas trabajan en ambientes Web, algunas otras en ambientes de escritorio. Además, por un lado, para poder hacer uso de las herramientas se debe cubrir un costo por licenciamiento de uso, lo que se ve como una desventaja para las pymes y particulares, ya que muchas veces los precios son altos. Por otro lado, después de hacer un análisis sobre las herramientas de este rubro, se encontró que estas herramientas no cuentan con un modelo de contrato basado en estándares internacionales de la ingeniería de software, como sí lo son ISO-12207, PMBOK, CMMI-ACQ [2, 6, 10].

Así, los contratos que ofrecen dichas herramientas, dentro del área del desarrollo de software, no cuentan con los elementos necesarios para el establecimiento de un contrato que garantice que las expectativas esperadas entre el cliente y el proveedor sean llevadas a cabo con éxito. Además, tampoco ofrecen mecanismos de evaluación para los contratos con el fin de medir el cumplimiento de los acuerdos establecidos en los mismos.

En la tabla 1, se presentan los aspectos tomados en cuenta para el análisis de algunas de las herramientas comerciales.

Tabla 1. Herramientas de gestión de contratos

Nombre / Características	Emptoris Contract Management	Business in a Box	Contract Eagle
Estándares	En su especificación no se menciona ningún estándar de proceso de adquisición	No contempla ningún estándar internacional dentro de sus plantillas	No especifica el uso de estándares para la definición de sus plantillas
Método de evaluación	En su especificación no se encontró la especificación de ningún método de evaluación	Dentro de sus características no se detecta la especificación de algún método de evaluación de contratos	No contempla mecanismos de evaluación de cumplimiento del contrato
Modelo de contrato de Ingeniería de Software	Sus plantillas se basan en formatos predefinidos	Se basa en plantillas predeterminadas	No especifica el uso de algún modelo de contrato
Gestión de contratos	Administración de los contratos, a través de un ciclo de vida de contrato establecido	Solo contempla la edición de plantillas	Ofrece la gestión de los documentos de contrato, creando un seguimiento del documento y recordatorios vía e-mail de los contrato a vencer
Generación de plantillas dinámicas	No especifica el tipo de plantillas que ofrece, ni como son generadas	Únicamente proporciona las plantillas y embebe un editor de texto para la edición	No proporciona ningún tipo de plantillas dinámicas
Tecnología de gestión y persistencia de conocimiento	Base de datos tradicional. Un repositorio central	No especifica la tecnología de persistencia	Utiliza una base de datos en SQL Server 2005 o posterior
Licencia	Licencia comercial con un costo aproximado de $1, 500.00 dólares	Su licencia tiene un costo de $ 299.85 dólares	Versión Estándar $1, 999.00 dólares Versión Empresarial $ 6, 439.00 dólares
Entorno de trabajo	Ambiente Web	Ambiente de escritorio, compatible con Windows y Mac OS	Servidor IIS y bajo un entorno Web para los clientes

Como se observa en la tabla anterior, las herramientas analizadas ofrecen prestaciones muy similares, sin embargo, no cubren aspectos tales como: la inclusión de un modelo de contrato de ingeniería de software y un método de evaluación del cumplimiento de las expectativas del contrato, el uso de tecnologías para la gestión y persistencia del conocimiento, software libre. Por lo que, la propuesta de innovación de la herramienta Quetzalcóatl se basa en que sea una herramienta de uso libre, que la gestión de los contratos sea a través de la creación de contratos personalizados haciendo uso de una ontología de acuerdo a los elementos específicos de las necesidades de cada usuario, con contratos sustentados en el modelo de contrato y con un método de evaluación para medir el porcentaje de cumplimiento de los elementos que integran dichos contratos.

2.4 Quetzalcóatl

Como se ha visto en las secciones anteriores, la problemática que se tiene a la hora de que las organizaciones desean contratar servicios de TI, así como las pymes que ofrecen servicios de TI, muchas veces no tienen la certeza de qué elementos debe contemplar un contrato para poder establecer la relación de trabajo con los clientes y/o proveedores.

Quetzalcóatl es una herramienta que integra un modelo de contrato de ingeniería de software basado en estándares internacionales como CMMI-ACQ [2], ISO 12207:2008 [6], PMBOK [10], dicho modelo de contrato se describió brevemente en la sección 2. Por lo que, la arquitectura de Quetzalcóatl busca ofrecer una herramienta que trabaje en un ambiente Web, que ofrezca la generación dinámica de contratos, plantillas de contratos, evaluación de los contratos generados, así como, la gestión del conocimiento por medio de la persistencia de los datos, la cual será gestionada a través de una ontología y, a su vez, para que cualquier persona con acceso a Internet tenga la posibilidad de hacer uso de ella.

A continuación se listan las funcionalidades principales de la herramienta Quetzalcóatl:

- Administración de perfil de usuario.
- Gestión de los contratos (creación, actualización, consulta de contratos).
- Evaluación de contratos externos a la herramienta.
- Patrones de contratos basados en el modelo de contrato a través de una ontología.
- Generación dinámica de contratos a través de una ontología.

2.5 Propuesta arquitectónica para la herramienta Quetzalcóatl

A lo largo de esta sección, se describe la propuesta arquitectónica, tomando en consideración el análisis de las características de las herramientas, descritas en la sección 2.3, así como el modelo de contrato y el método de evaluación que integrarán la herramienta Quetzalcóatl.

De acuerdo con I. Jacobson *et. al.* [11], la arquitectura de un sistema de información define no sólo la estructura y comportamiento, sino también, el uso, la funcionalidad, el rendimiento, la flexibilidad, la reutilización, la facilidad de comprensión, las restricciones y compromisos económicos y tecnológicos, y por supuesto, la vista.

Algunos investigadores definen a la arquitectura de un sistema de software como el conjunto de decisiones de diseño tomadas para un sistema [12]. Si se hace una analogía con la construcción de un edificio, se puede asumir que la arquitectura representa a los planos de construcción del software.

De acuerdo a la técnica de ingeniería de software 4 + 1 vista. De dicha técnica se toma la vista de desarrollo para el modelado de la arquitectura de la herramienta, basándose en el patrón arquitectónico Modelo-Vista-Controlador, y de acuerdo con las recomendaciones realizadas en [12], ya que en este patrón es notable la separación de los componentes relacionados con los datos de la aplicación de los componentes de la interfaz de usuario. Permitiendo tener, a nivel de desarrollo, un código más claro, flexible y reusable.

Cuando la aplicación sufre un cambio en su modelo de objetos, se envía un aviso del cambio de vista, de manera que los elementos afectados puedan ser modificados o actualizados. Estos avisos se emiten al controlador típicamente, de manera que el controlador puede modificar la vista si se diera el caso. La vista se comunica con el modelo para obtener los datos necesarios para mostrarlos en la interfaz. Al momento de realizar una entrada por parte del usuario, el controlador esta pendiente de la información de la vista para verificar qué medidas tomar de acuerdo con las solicitudes hechas.

Después, en caso de que el modelo de objetos cambie de valores, esto debe ser comunicado a la vista y al controlador con el objetivo de que la interfaz sea actualizada. Como se menciónó anteriormente la arquitectura de Quetzalcóatl está dividida en dos niveles: aplicación y datos.

En el nivel de aplicación se encapsulan los elementos de la arquitectura basados en el patrón arquitectónico MVC, en la cual se han definido los siguientes módulos:

2.5.1 Vista

Esta capa contiene las interfaces gráficas de usuario, que dan soporte a las pantallas gráficas de la herramienta Quetzalcóatl, estas interfaces (formularios, gráficos, imágenes, etc.) se componen de tres secciones correspondientes a: vista_contratos, vista_metodos, vista_usuarios. La sección de vista_contratos encapsula la interfaz referente a la administración de los contratos; la sección vista_metodos contiene las interfaces cor-

respondientes a los métodos de evaluación de los contratos; la sección vista_usuarios contempla las interfaces de la administración de cuentas de usuario, login, registro y perfiles de usuario.

2.5.2 Control

En esta capa se definen los módulos correspondientes a la capa de control, esta capa se encarda de las operaciones diversas que se realicen. Como se sabe de acuerdo al patrón MVC, la capa de control encapsula las clases correspondientes a cada una de las acciones o peticiones realizadas a través de la interfaz gráfica de usuario, por lo que se encuentran separadas de acuerdo a la funcionalidad que representan: gestión contratos, gestión métodos, control perfiles.

2.5.3 Modelo

En esta capa se define la parte correspondiente a la lógica del negocio, en términos coloquiales: "el motor de nuestra herramienta". Ésta es la capa más importante de la herramienta, puesto que encapsula todas las clases que mantienen la relación entre los datos y la información que se requiere mostrar en la interfaz gráfica.

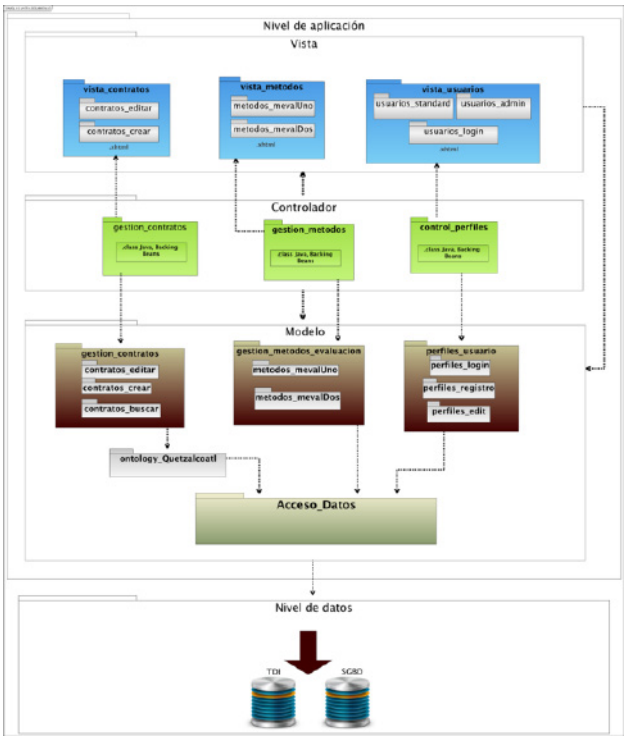


Figura 2. Arquitectura Quetzalcóatl

Como se puede apreciar en la figura 2, esta capa esta compuesta de cuatro paquetes encapsulando a las clases correspondientes de acuerdo con la funcionalidad que representan.

2.5.4 Acceso_Datos

En este módulo se encapsulan las clases de conexión con la base de datos que servirá para la persistencia de la información referente a los proyectos y cuentas de usuario, así como la conexión al gestor tripletas de la ontología. En la capa de datos, se agrupan los elementos de persistencia de la información y del conocimiento, tales como un sistema gestor de base de datos y un gestor de tripletas para la ontología.

3 MODELO ONTOLÓGICO PARA LA REPRESENTACIÓN DEL MODELO DE CONTRATO

En esta sección se presenta el modelo de la ontología que representa al modelo de contrato de [5]. Hoy en día el campo de la Web Semántica ha crecido a pasos agigantados, desde aquel artículo publicado por el creador de la web, Tim Berners-Lee [14], en el que describía una nueva web, en donde los contenidos fueran entendidos no solamente por humanos, sino que también fueran entendidos por agentes electrónicos, es decir, software que tuviera la capacidad de entender los contenidos diseñados por ellos. También ,se ha dado una evolución desde finales de la década de los 90, fecha en la que se lanzó la primera especificación, que se han convertido en estándares de facto por ser recomendaciones del Consorcio World Wide Web (W3C). Algunas de estas especificaciones son: RDF, que es un lenguaje para crear modelos semánticos; OWL: que es un lenguaje para la especificación de conceptos, relaciones y propiedades en una ontología; SPARCQL: que es uno de los lenguajes de consultas sobre datos semánticos [15]. La W3C recomienda el uso de OWL y RDF desde 2004 y en 2007 recomienda el uso de SPARQL como lenguaje de consulta.

No obstante, los lenguajes de construcción de ontologías siguen en un proceso de madurez hoy en día. Con lo anterior se ha visto que, gracias a la web semántica, las ontologías ofrecen mecanismos de representación semántica a través de la definición de conceptos, reglas y relaciones entre conceptos, propiedades de los conceptos considerando la capacidad del lenguaje de desarrollo como OWL.

De lo mencionado en el párrafo anterior, consideramos que el uso de una ontología es un buen mecanismo para la representación del modelo de contrato. Ya que mediante la ontología se puede establecer conceptos y relaciones entre ellos sobre un dominio en específico. Para entender esto primero veamos: ¿qué es una ontología? Una ontología [17] es una descripción explícita y formal de conceptos en un dominio o contexto, el cual está formado por clases (también llamados conceptos), propiedades de los conceptos (a veces llamados Slots) y restricciones sobre los Slots.

Lo que se busca con el modelado de la ontología, es la representación del modelo de contrato. Para lo cual se desarrolló de acuerdo con el proceso de desarrollo de ontologías que recomienda [18]. En la ontología se representa cada una de las siete categorías del modelo de contrato, con sus cláusulas, y componentes correspondientes. Ya que que el modelo de contrato obedece a una jerarquía: categoría – cáusula – componente, lo que hace más fácil el diseño de la ontología.

En la Figura 3, se muestra el modelo de contrato que se representa con la ontología, en la figura se puede observar que enfatiza tres aspectos: Representación, generar contratos y búsqueda sobre las taxonomías, estos puntos son los elementos sobre los que la ontología apoyará al modelo de contrato, es decir, con la ontología se forman patrones de contrato de diferentes áreas en proyectos de desarrollo de software, y sobre los cuales se buscan los patrones que al usuario más le convengan de acuerdo a ciertos parámetros de inferencia.

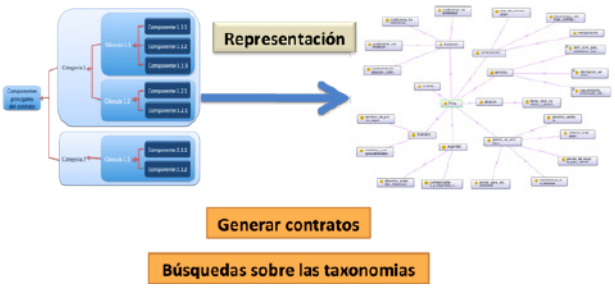


Figura 3. Representación del modelo de contrato

En la figura 4, se muestra un esquema de la representación de modelo de contrato, en el cual se definen las relaciones que da la semántica para realizar la inferencia en la ontología. La ontología básicamente representa al modelo de contrato, de esta manera, se hace el modelo de la ontología siguiendo la estructura del modelo de contrato. Como se mencionó en la

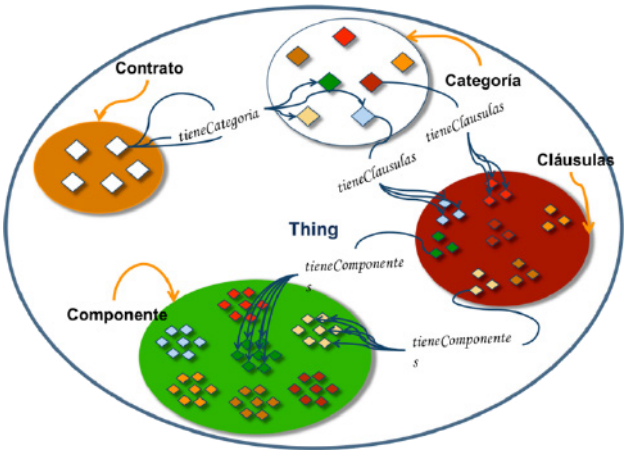


Figura 4. Modelo conceptual de la ontología de contrato

sección 2.1, el modelo de contrato se compone de categorías, cláusulas, y componentes, por lo que la ontología sigue este esquema.

Las clases principales que se definen son las siguientes: contrato, categoría, cláusulas, componente, cabe mencionar que el modelo no muestra el total de elementos que integran la ontología, ya que como todo desarrollo, se está en una etapa de refinamiento de la misma.

4 TRABAJO A FUTURO

Actualmente se está trabajado en la mejora tanto de la ontología y de la definición de las taxonomias de los contratos que la herramienta Quetzalcóatl ofrecerá. Para lograr esto, se sigue experimentando con la API de Jena para la manipulación de ontologías OWL (Web Ontology Language) y RDFS (Resource Description Framework Schema). Para llevar a cabo el mejoramiento, se están analizando contratos de compañías de diferente tamaño.

5 CONCLUSIÓN

La problemática que se tiene con respecto de los proyectos de desarrollo de software en entornos de *outsourcing*, es la falta de conocimientos de los elementos que debe contener un contrato en este tipo de entornos por parte de las organizaciones; además de la carencia de herramientas que permitan la generación y evaluación de contratos con manejo de los datos persistentes de manera dinámica mediante

el uso de ontologías. Otro aspecto importante es que las herramientas que apoyan la generación de contratos de este tipo, son comerciales lo cual limitan a las organizaciones para adquirirlas debido sus altos costos de venta. La propuesta de la arquitectura de la herramienta Quetzalcóatl, presentada en el artículo, muestra la estructura que se contempla para poder resolver la problemática planteada, tomando en consideración las características carentes de las herramientas analizadas.

REFERENCIAS

1. Jae-Nam, L., Minh Q. H., (2003) "IT outsourcing evolution: past, present, and future". Communications ACM, 46.
2. Gallagher, B. P., et al. (2007) "CMMI for Acquisition, Version 1.2" [ref. 2013]. Disponible en Web: <http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=8451>.
3. International Data Corporation (2010), "Visión de la Industria en el Segundo Semestre" .
4. Matthew K. O. L. (1996), "IT Outsourcing Contracts: Practical Issues for Management". Information Systems Department City University of Hong KongKowloon. Hong Kong.
5. Mejía Miranda, J. (2010), "Marco para el gobierno de la externalización del proceso de desarrollo del software a través de la Ingeniería del Contrato". Cap. 4. 2010, Universidad Politécnica de Madrid, Madrid, España.
6. IEEE Computer Society. "Systems and Software Engineering - Software life cycle process, ISO/IEC 12207-2008". Second Edition. 2008.
7. Emptoris Inc. "Emptoris Contract Management: Driving Competitive Advantage" (2011). [ref. 25 de octubre de 2011]. Disponible en Web: <http://www.emptoris.com/solutions/contract_management_software.asp>
8. Biztree Inc. "Business in-a-Box". [ref. 2 de septiembre de 2011]. Disponible en Web: <www.biztre.com> Contract Eagle Inc. "Contract Eagle" (2011), [ref. 20 de octubre de 2011]. Disponible en Web: <www.contracteagle.com>
9. Project Management Institute (2011), "A guide to the Project Management Body of Knowledge (PMBOK Guide)". Project Management Institute. 2011.
10. Jacobson et al., (2000) Jacobson, I., Booch, G., Rumbaugh, J. "El proceso unificado de desarrollo de software", Rational Software Corporation. Pearson Addison Wesley. Madrid, 2000.
11. Tahuiton Mora, J. (2011) "Arquitectura de software para aplicaciones Web", Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional. México, D.F.
12. González, D. L. (2007) "La industria Mexicana del software. Un estudio en cifras", [ref. 2013]. Disponible en Web: <http://www.sg.com.mx/content/view/486>.
13. Berners-Lee, et al., (2001), T. Berners-Lee, J. Hendler, and O. Lassila. "The semantic web". Scientific American, 284(5):28–37.
14. Prud'hommeaux et al., (2007), E. Prud'hommeaux and A. Seaborne. Sparql query language for rdf. Technical report, W3C Working Draft 26 March 2007.
15. Yalan Yan, Jinlong Zhang, Mi Yan. "Ontology Modeling for Contract: Using OWL to Express Semantic Relations". EDOC '06 Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference, School of Management, Huazhong University of Science and Technology, Wuhan, P.R.China.
16. F. Natalya, Noy and L. Deborah., (2005) "Desarrollo de Ontologías-101: Guía para crear tu primera Ontología". Stanford University. Stanford, CA.
17. Muñoz et al., (2009) Muñoz, E., Espuña, A., Puigjaner, L. Towards an Ontological Infraestructure for Chemicals Batch Process Management. Department of Chemical Engineering, Universidat Politécnica de Catalunya. Barcelona, España.

Acerca de los autores



Sergio David Ixmatlahua Díaz es Maestro en Sistemas Computacionales por el Instituto Tecnológico de Orizaba. Durante sus estudios de maestría realizó una estancia profesional en la Facultad de Informática de la Universidad Politécnica de Madrid, España. Ha participado en congresos internacionales y nacionales como ponente, presentando artículos relacionados con la ingeniería de software y desarrollo web. Actualmente se desempeña como profesor en el Instituto Tecnológico de Iztapalapa, en la ciudad de México. Participa ampliamente en el análisis y diseño de proyectos software con el uso de metodologías ágiles y UML, así como el uso de estándares de calidad de software.



Jezreel Mejía Miranda es Doctor en Informática con mención de Doctorado Europeo por la Facultad de Informática de la Universidad Politécnica de Madrid, España con obtención de nota Cum laude, dentro del programa de Lenguajes y Sistemas Informáticos e Ingeniería de Software. Actualmente es miembro de la Cátedra de Mejora de Procesos Software en Espacio Iberoamericano, y se encuentra afiliado al Centro de Investigación en Matemáticas, A. C., Unidad Zacatecas, en donde establece proyectos relacionados con la mejora de procesos software.



Mirna Ariadna Muñoz Mata es Doctora en Informática con mención de Doctorado Europeo por la Facultad de Informática de la Universidad Politécnica de Madrid, con obtención de nota Cum laude, dentro del programa de Lenguajes y Sistemas Informáticos e Ingeniería de Software. Actualmente es miembro de la Cátedra de Mejora de Procesos Software en Espacio Iberoamericano. Ha participando activamente en proyectos realizados en el área de mejora de procesos software enmarcada dentro de la ingeniería de software vinculados con la empresa Everis Consulting. Como resultado ha realizado diferentes publicaciones en congresos internacionales y nacionales, así como artículos en revistas de alto impacto y presentación de pósters. Asimismo participó en la traducción oficial al español de CMMI DEV v1.2 y actualmente participa en la traducción de la nueva versión v1.3. Además, se desempeña como investigadora en el Centro de Investigación en Matemáticas, Unidad Zacatecas.



Roque Manuel Rueda Anastacio es Licenciado en Sistemas Computacionales Administrativos por la universidad Veracruzana. Actualmente se desempeña como Senior System Engineer en Infosys en la ciudad de México. Participa activamente en proyectos de desarrollo y mantenimiento de software enfocado en la mejora de procesos software. Ha realizado trabajo en el análisis de metodologías de desarrollo de software y mejores prácticas, asimismo en diseño arquitectónico de sistemas e implementación de nuevas tecnologías. Actualmente se desempeña en proyectos de mantenimiento de software. Sus áreas de Conocimiento son principalmente desarrollo móvil, android, IOS, metodologías ágiles, desarrollo y mantenimiento de software, arquitectura de software.



Alejandro D. Velázquez Cruz es Maestro en Sistemas Computacionales por el Instituto Tecnológico de Orizaba. Actualmente se desempeña como docente en el Instituto Tecnológico de Iztapalapa, en la Ciudad de México. Ha participado en diversos proyectos de gestión y desarrollo de software de calidad dirigiendo tesis de la licenciatura en informática en el Tecnológico de Laguna e Iztapalapa. Tiene especial interés en el desarrollo de software en C# y Java utilizando las técnicas de análisis y diseño de sistemas con UML. Es docente en el departamento de Ciencias Computacionales de la University of People, donde además imparte cursos de fundamentos de programación. Ha participado en diversos congresos nacionales e internacionales como ponente.