

INGENIERÍA ONTOLÓGICA APLICADA EN EL DISEÑO DE UN SISTEMA DE ONTOLOGÍAS PARA LA GESTIÓN DE HORARIOS

Maricela Claudia Bravo Contreras

Universidad Autónoma Metropolitana
mcbc@correo.azc.uam.mx

Francisco Pavón Gutiérrez

Universidad Autónoma Metropolitana
teorema06@gmail.com

José Alejandro Reyes Ortiz

Universidad Autónoma Metropolitana
jaro@correo.azc.uam.mx

Roberto Alfonso Alcántara Ramírez

Universidad Autónoma Metropolitana
raar@correo.azc.uam.mx

Resumen

En este artículo se describe el proceso de ingeniería ontológica aplicado en el diseño y construcción de un sistema de ontologías para la gestión de horarios. Se presenta la metodología detallada que se implementó, la cual consistió de las siguientes etapas: especificación de las preguntas de competencia, diseño modular y basado en dominios del sistema de ontologías, diseño y evaluación de cada ontología individualmente, integración de las ontologías y evaluación general del sistema de ontologías mediante preguntas de competencia. Como resultado se obtuvo un sistema de ontologías que cumple con los principios de diseño de claridad, coherencia, modularidad, y usabilidad. Finalmente se evaluó la competencia de la ontología ejecutando las preguntas de competencia a través de la definición de reglas de inferencia lógica.

Palabras Claves: Ingeniería ontológica, principios de diseño de ontologías, sistema de ontologías.

Abstract

In this paper we describe the ontological engineering process applied in the design and construction of an ontology system for academic schedule management. We present the detailed methodology that was implemented, which consisted of the following stages: specification of competency questions, modular and domain-oriented design of the ontology system, design and evaluation of each ontology, integration of ontologies and general evaluation of the ontology system. As a result, a system of ontologies was obtained that complies with the design principles of clarity, coherence, modularity, etc. Finally, the competence of the ontology was evaluated by executing the competency questions through the definition of inference and query rules.

Keywords: *Ontology engineering, ontology design principles, ontology system.*

1. Introducción

La ingeniería ontológica es el proceso mediante el cual se diseñan y aplican métodos, técnicas y principios de diseño con el objetivo de producir una ontología o sistema de ontologías. Una pregunta frecuente durante el diseño de una ontología es decidir sobre la incorporación de otras ontologías en una sola. En este artículo nos referimos al concepto de **sistema de ontologías** como una ontología que incluye o incorpora otras ontologías, dentro de la cual se definen relaciones semánticas que existen tanto entre conceptos como entre individuos de las diferentes ontologías importadas. El objetivo de la ingeniería ontológica es apoyar todas las etapas del diseño y desarrollo de manera eficiente para lograr ontologías de calidad. Un factor clave para lograr la eficiencia durante el diseño de ontologías es la incorporación de principios de diseño.

Los principios de diseño son criterios de calidad que guían y orientan el diseño y construcción de las ontologías con el objetivo principal de lograr ontologías reutilizables, fáciles de mantener y actualizar a lo largo de su vida útil. Asimismo, a

partir de los principios de diseño es posible proponer mecanismos para evaluar la calidad del diseño de una ontología. En esta sección se presenta una revisión de los principios de diseño que han sido propuestos y discutidos por renombrados autores en la literatura especializada.

Thomas Gruber [Gruber, 1993] fue uno de los primeros autores que describió principios de diseño aplicables a las ontologías. Gruber propuso un conjunto de criterios de diseño cuyo propósito era compartir el conocimiento y facilitar la interoperabilidad entre programas mediante una conceptualización compartida. Los criterios de diseño de ontologías descritos por Gruber son los siguientes:

- Claridad. Especifica que una ontología debe comunicar efectivamente el significado intencional de los términos definidos. Las definiciones deben ser objetivas e independientes del contexto social o computacional. Siempre que sea posible, una definición debe estar completa con las condiciones necesarias y suficientes, en lugar de ser parcialmente definida. Todas las definiciones deben documentarse en lenguaje natural.
- Coherencia. Establece que una ontología debe generar inferencias que sean consistentes con las definiciones. Esto implica que todos los axiomas definidos en la ontología deben ser consistentes lógicamente. La coherencia también debe aplicarse a los conceptos descritos en lenguaje natural en la documentación. Si un enunciado que puede inferirse de los axiomas contradice una definición dada informalmente, entonces la ontología es incoherente.
- Extensibilidad. Establece que una ontología debe diseñarse anticipando los posibles usos del vocabulario. Debe ofrecer el fundamento conceptual para un rango de tareas anticipadas, y la representación debe de construirse de tal forma que se pueda extender y especializar la ontología monotónicamente. En otras palabras, debe permitir la definición de nuevos términos para usos especiales basados en el vocabulario existente.
- Tendencia de codificación mínima. La conceptualización debe ser especificada a nivel de conocimiento sin depender de una codificación particular a nivel de símbolos.

- Compromiso ontológico mínimo. Una ontología debe poseer el compromiso ontológico mínimo y suficiente para soportar las actividades requeridas de conocimiento compartido. Esto significa que una ontología debe hacer tan pocas aclamaciones como sea posible sobre el espacio del mundo que está modelando, permitiendo que las partes adheridas a la ontología tengan la libertad de especializar e instanciar la ontología como sea necesario.

En [Fox, 1998] los autores propusieron seis características para evaluar un Modelo Empresarial. Estas características fueron propuestas para responder a la pregunta de ¿Cómo se puede determinar cuál ontología es la correcta para determinada tarea? Para dar una pauta sobre la operatividad de estas características, los autores definen el concepto de competencia del modelo de la siguiente forma: dado un modelo apropiadamente instanciado y un demostrador de teoremas, la competencia de un modelo es el conjunto de preguntas que el modelo puede responder. Otro significado de competencia se refiere a la expresividad requerida de la ontología para representar las preguntas de competencia y para caracterizar sus soluciones. Con base a este concepto de competencia, explican las características que se listan a continuación:

- Completitud funcional. Esta característica es determinada por la competencia de la ontología. Esto es, el conjunto de preguntas que puede responder con un modelo instanciado apropiadamente.
- Generalidad. La generalidad de una ontología puede determinarse verificando si la unión de preguntas de un amplio conjunto de funciones, inclusive provenientes de diferentes sectores, es reducible al conjunto de preguntas de competencia de la ontología en cuestión.
- Eficiencia. La eficiencia de una ontología puede medirse calculando el número de inferencias lógicas por segundo requeridas para responder una pregunta. Sin embargo, los autores toman en cuenta que existe más de una forma para representar el mismo conocimiento y que cada representación no tiene la misma complejidad al responder a una clase específica de

preguntas. Por lo tanto proponen calcular la complejidad promedio de las preguntas de competencia para estimar la eficiencia.

- Claridad. La claridad de una ontología se logra mediante la axiomatización. Esto es, proporcionar definiciones formales de los objetos, sus relaciones y sus atributos.
- Precisión. Esta característica se refiere al grado en el que las definiciones de los conceptos se declaran distintos entre sí. Por ejemplo: si un concepto está incluido en otro, o qué conceptos se encuentran en la intersección o en la unión de dos o más conceptos.
- Granularidad. Esta característica se refiere a la capacidad de una ontología de representar conceptos en diferentes niveles de abstracción.
- Minimalista. Esta característica se determina utilizando los axiomas para determinar que por cada objeto o concepto en la ontología, no existe otro objeto que sea lógicamente equivalente.
- En [Morbach, 2009] los autores establecieron que una ontología debe cumplir con dos objetivos prioritarios: ser usable y reutilizable:
 - a) La reutilización la definen como el grado en el que un módulo de software u otro producto puede ser utilizado en más de un programa de cómputo o sistema de software. En particular, la reutilización de ontologías puede definirse como la capacidad de adaptación de una ontología a contextos de aplicación arbitrarios, incluyendo aquellos contextos que no fueron previstos al momento de la creación de la ontología.
 - b) La usabilidad, por otro lado denota el grado en el que un componente de software es útil para una tarea o aplicación específica. El término también tiene la connotación de facilidad de uso, refiriéndose al esfuerzo requerido por un usuario para utilizar un sistema de software dado. El objetivo de la usabilidad de una ontología es minimizar el esfuerzo requerido para adecuar la ontología de tal forma que pueda ser usada por humanos o máquinas en un contexto

de aplicación dado. En este sistema de ontologías se atendieron los principios de Modularidad, Coherencia, Claridad y Usabilidad.

En relación con las metodologías para el diseño y construcción de ontologías, [Gómez, 3] establece que una metodología se compone de métodos, técnicas, procesos y actividades. En esta sección se presenta una revisión de las metodologías de diseño de ontologías.

La metodología para construir la ontología CyC fue presentada por [Lenat, 1990]. Los autores proponen tres métodos para la construcción de una ontología: extracción manual y codificación del conocimiento, extracción semi-automática y codificación del conocimiento, y extracción completamente automática y representación del conocimiento.

[Uschold, 1995] presentaron una metodología para desarrollar y evaluar ontologías, la cual fue refinada y mejorada después por [Uschold, 1996]. Esta metodología incluye las siguientes etapas: identificar el propósito, construir la ontología – captura de la ontología, codificación de la ontología, integración de ontologías existentes -, evaluación y documentación.

En [Grüninger, 1995] los autores describen el proceso de construir una ontología para el proyecto TOVE (Toronto Virtual Enterprise). La metodología consiste de las siguientes etapas: describir el escenario de motivación, definir las preguntas de competencia, definir la terminología de la ontología (objetos, atributos y relaciones), y especificar las definiciones y restricciones, finalmente la evaluación de la ontología se realiza por el teorema de completitud.

En [Berner et al., 1996] los autores presentaron una metodología para construir ontologías para el proyecto KACTUS. El objetivo de la metodología era reutilizar las ontologías existentes y evaluar la factibilidad de reutilizar conocimiento en sistemas complejos. Esta metodología estableció los siguientes procesos generales: desarrollo de la lista de requerimientos (especificación de la aplicación), identificación de los términos generales para generar un modelo preliminar, estructuración y refinamiento de la ontología, y búsqueda de ontologías pre-existentes para reutilizarlas.

METHONTOLOGY fue presentada por primera vez por [Gómez, 1996]. Posteriormente refinada y detallada por [Fernández et al, 1997], [Fernández, 1981], y [Fernández, 1999]. El objetivo de esta metodología es facilitar la construcción de ontologías. METHONTOLOGY define un proceso de desarrollo y un ciclo de vida el cual consiste de las siguientes actividades: especificación, conceptualización, formalización, implementación, y mantenimiento. La conceptualización consiste en: construir el glosario de términos, construir la taxonomía de conceptos, construir el diagrama de relaciones binarias, y construir el diccionario de conceptos.

CommonKADS [Schreiber, 2000] es una metodología que reconoce la necesidad de reutilizar elementos del modelo de conocimiento o una combinación de éstos. La metodología CommonKADS incluye las siguientes etapas: identificación del conocimiento, especificación del conocimiento, y refinamiento del conocimiento.

La metodología NEON [Suárez, 2010] se basa en el uso de patrones de diseño. Esta metodología propone la reutilización de ontologías de un repositorio público de ontologías y un conjunto de patrones de diseño conocidos para integrarlos mediante procesos de re-ingeniería. Las etapas generales de esta metodología son: identificación de requerimientos, identificación de los patrones de diseño disponibles, dividir y transformar el problema en problemas parciales, encontrar las coincidencias de los problemas parciales con los patrones de diseño, seleccionar el patrón de diseño, aplicar el patrón de diseño para hacer una composición, evaluar las soluciones parciales, e integrar las soluciones parciales.

A pesar de que se han presentado varias propuestas metodológicas para la construcción de ontologías, éstas no consideran la incorporación de principios de diseño desde la especificación de la competencia de la ontología, hasta su evaluación final.

En este artículo se presenta una metodología integral para el diseño y construcción de un sistema de ontologías para la gestión de horarios. La metodología descrita se basa en los principios de diseño de coherencia, inteligibilidad y modularidad. La evaluación de la ontología se realiza con base a su competencia y atendiendo una lista inicial de preguntas de competencia.

2. Métodos

En esta sección, se describe detalladamente la metodología integral que se implementó para el diseño, construcción y evaluación del sistema de ontologías para la gestión de horarios. Uno de los aspectos más importantes de esta metodología es que se aplica para el diseño y construcción de ontologías de dominio de aplicación.

De acuerdo con [Morbach, 2009] una ontología de dominio tiene como objetivo capturar el conocimiento de todo un dominio de aplicación como la física, la química o la ingeniería.

Las principales características que fueron incorporadas en esta metodología son las siguientes:

- Diseño del sistema de ontologías con orientación a dominios.
- Centrado en la reutilización de las ontologías mediante el diseño modular.
- Lograr que las ontologías resultantes sean coherentes.
- Metodología iterativa e incremental.
- Evaluación del sistema de ontologías basada en principios de diseño de calidad.
- Especificación y evaluación de la competencia del sistema de ontologías por medio de preguntas de competencia.

Esta metodología está prevista para realizarse con un equipo de trabajo integrado por un grupo de expertos en el dominio de aplicación, un grupo de programadores con experiencia en el desarrollo de aplicaciones que explotan ontologías, y un grupo de ingenieros ontológicos. La figura 1 presenta la metodología dividida en cuatro etapas.



Figura 1 Etapas de la metodología para la construcción del sistema de ontologías.

Etapas 1. Diseño General del Sistema de Ontologías

Durante esta etapa se realiza el diseño general del sistema de ontologías, identificando primero los conceptos clave, a partir de estos conceptos definir cuáles son las clases candidatas y separando por dominios los módulos del sistema de ontologías:

- Elicitación de términos. El diseño general de sistema de ontologías para la gestión de horarios se inició con la elicitación de términos; para ello se implementó una técnica de Preguntas de Competencia, la cual consiste en la definición de una lista de preguntas que la ontología debe ser capaz de contestar con todos los conceptos (clases), propiedades entre conceptos, propiedades entre conceptos y datos, los axiomas y reglas definidas en la ontología. Se solicitó a los participantes en el equipo de diseño de la ontología que indicaran las preguntas que les interesaba que la ontología fuera capaz de contestar con respecto al escenario que se planteó al inicio del ejercicio. De esta actividad la lista de preguntas de competencia resultante es la siguiente:

¿En qué salón se encuentra profesor “Pérez”?

¿Cuáles son los cursos que se imparten en el salón F305 los días martes?

¿Qué horario tiene el profesor “Suárez”?

¿Cuáles son los cursos que tienen en su nombre la palabra “programación”?

¿Cuáles son los cursos que se imparten los martes de 10:00 a 12:00?

¿Cuáles son los cursos que pertenecen al Tronco General?

¿Cuáles son los cursos que pertenecen al Tronco Básico Profesional?

¿Cuántos créditos tiene el curso “Teoría de la Computación”?

¿Cuáles son los cursos que pueden llevar corrección?

¿Qué profesores imparten la materia de “Compiladores”?

¿Cuántos créditos suman todos los cursos del área de Sistemas de Información?

¿Cuáles son las claves de los cursos del área de Sistemas Embebidos?

¿Cuáles son las horas y los días en que se imparten los cursos del tronco Inter y Multidisciplinar?

¿Cuáles son los cursos que requieren autorización para inscribirse a ellas?

¿Quiénes son todos los profesores que pertenecen a la División de Ciencias Básicas?

Como resultado de la elicitación de términos, se identificaron los siguientes conceptos como clases candidatas: institución, profesor, estudiante, curso, salón. Antes de finalizar la lista de conceptos, es necesario determinar si se requiere más de una ontología, para ello se deben responder a las siguientes preguntas:

¿La lista de términos identificados podría especializarse de tal forma que su manejo se facilite en una ontología separada?

¿Es posible clasificar los términos en más de una ontología con un dominio particular claramente diferenciado en cada ontología?

¿De qué tamaño será la población de cada ontología y qué tan dinámica será dicha población?

- Identificación de módulos del sistema de ontologías. Esta actividad consiste en realizar un análisis de la lista de términos para identificar los conceptos principales del dominio de la aplicación, los cuales serán representados por ontologías separadas. El objetivo de esta actividad es aplicar el principio de diseño de Modularidad, para obtener ontologías individuales, auto contenidas, reutilizables y más fáciles de actualizar. En particular, para este sistema se identificaron las siguientes ontologías: Persona, UEA (curso), Salón, y Grupo. Las cuales se integrarían en un sistema de ontologías para la gestión de horarios.

Etapas 2. Diseño y Evaluación de las Ontologías Individuales

Durante esta etapa se realiza el diseño y evaluación de cada una de las ontologías que se definieron en la etapa anterior. Como resultado de la identificación de módulos, se decidió que los siguientes conceptos fueran implementados como ontologías individuales: Persona, UEA (curso), Salón, y

Grupo. Por cada una de estas ontologías se debe realizar el proceso iterativo que se muestra en la figura 2, el cual señala las actividades de: definir y verificar las relaciones taxonómicas; definir las relaciones entre objetos y las relaciones entre tipos de datos; realizar la axiomatización de clases, de propiedades y de individuos; y finalmente evaluar cada una de las ontologías individuales.



Figura 2 Proceso del diseño y evaluación de las ontologías individuales.

Descripción de la Ontología Persona:

- Paso 1. En este paso se realizan dos actividades: la definición de la jerarquía de clases y la validación de las relaciones taxonómicas. Como resultado del proceso de diseño de la ontología Persona, se definió la jerarquía de clases que se muestra en la figura 3. La jerarquía de clases consiste de tres clases principales llamadas Persona, GradoAcademico y Departamento. La clase Persona se sub-clasifica en Alumno, Empleado y Visitante, la Clase Empleado se sub-clasifica en Academico y Administraivo, y la clase Academico se sub-clasifica en Ayudante y Profesor. Todas estas clases se definieron para representar a los diferentes tipos de personas que pueden existir en un entorno académico. Para validar que la jerarquía de clases es “semánticamente” correcta se deben enunciar las relaciones taxonómicas; por ejemplo: un Profesor es un Academico, es un Empleado, y es una Persona. Si el enunciado suena congruente para el ser humano y experto del dominio del tema modelado, entonces la jerarquía es correcta.

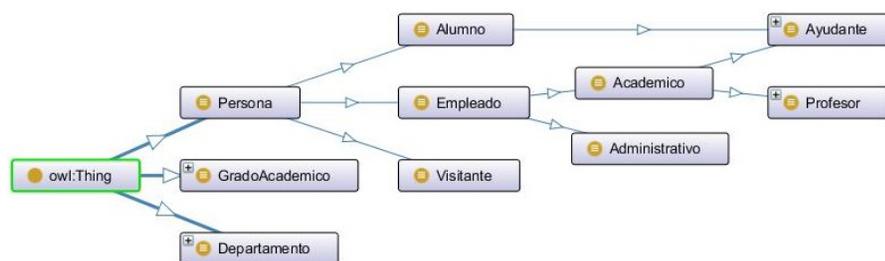


Figura 3 Jerarquía de clases de la ontología *Persona*.

- Paso 2. En este paso se realizan las siguientes actividades: definición de las relaciones entre clases y tipos de datos (dataProperties), así como las relaciones entre clases (objectProperties). Las propiedades de datos representan atributos o características de cada concepto. En una jerarquía de clases se definen primero todos los atributos que son comunes para todos los conceptos; las características comunes deben especificarse en el nivel de clase correcto. Así mismo, es muy relevante especificar una característica distintiva para cada concepto (sub-clase) de la jerarquía de clases para facilitar la tarea de clasificación de individuos del razonador. Las propiedades entre clases que se definieron en la ontología *Persona* son *tieneDepartamento* y *tieneGradoAcademico*, que se utilizaron específicamente con la clase de *Profesor*. Para cada relación entre objetos se debe decidir si la relación es de alguno de los siguientes tipos: funcional, inversa funcional, transitiva, simétrica, reflexiva, etc.
- Paso 3. Definición de axiomas de clases, propiedades e individuos. Durante este este paso se realiza la axiomatización de clases, la cual requiere que por cada clase se determine el conjunto de propiedades (dataProperties, objectProperties) necesarias y suficientes que cada individuo debe cumplir para pertenecer a la clase. También se definen otros tipos de axiomas como: clases disjuntas, cierre axiomático sobre propiedades, y axiomas de cobertura sobre clases. Por cada propiedad definida se deben identificar el tipo de restricciones a utilizar: restricciones de cardinalidad, restricciones existencial o universal, o restricciones de valor. La axiomatización o conceptualización de ontologías puede definirse como el proceso mediante

el cual todos los conceptos (o clases) de una ontología se definen explícitamente estableciendo las condiciones (o restricciones) necesarias y suficientes sobre el conjunto de propiedades de datos o de objetos. En una jerarquía de clases todas las propiedades comunes se heredan en las sub-clases, por lo tanto, se deben definir un conjunto mínimo de propiedades específicas y distinguibles en cada sub-clase para poder identificar de forma inequívoca a cada concepto dentro de la ontología.

En la figura 4 se muestran varios ejemplos de la implementación de axiomas sobre la clase Profesor. Por ejemplo: se observa que un Profesor por el hecho de ser Persona hereda las restricciones sobre las propiedades de tieneGenero y tieneNombrePersona. Así mismo se observa que por ser Empleado hereda la restricción sobre la propiedad de tieneNumEconomico y que por ser Academico hereda las restricciones de tieneCategoriaNivel, tieneCorreoElectronico y tieneDepartamento. Adicionalmente se estableció una restricción más sobre la propiedad tieneGradoAcademico. Por todas las restricciones anteriores, un individuo de la clase Profesor, para pertenecer a esta clase, debe cumplir con todo el conjunto de restricciones, tanto las heredadas como las propias.

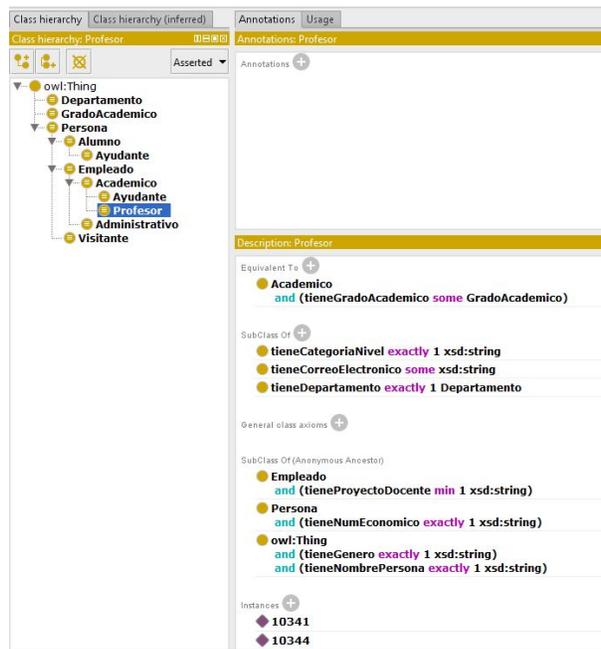


Figura 4 Descripción de la clase Profesor.

- Paso 4. En este paso se realizan las actividades de poblado y corrección de la ontología. El propósito de instanciar individuos en la ontología es para verificar el uso de los conceptos definidos, verificar si los tipos de datos especificados para las propiedades son correctos e intuitivos. En esta actividad se realiza la evaluación de la “usabilidad” de la ontología para la creación de nuevos individuos. Si alguna propiedad de datos o de clases no es clara ni útil, ésta se modifica o se elimina definitivamente.
- Paso 5. En este paso se ejecutan las tareas de chequeo de consistencia lógica del razonador para verificar que la ontología poblada sea lógicamente congruente. En la figura 5 se muestran las propiedades de objetos y de datos de un individuo instanciado en la clase de Profesores. La ontología fue evaluada verificando que fuera consistente lógicamente.

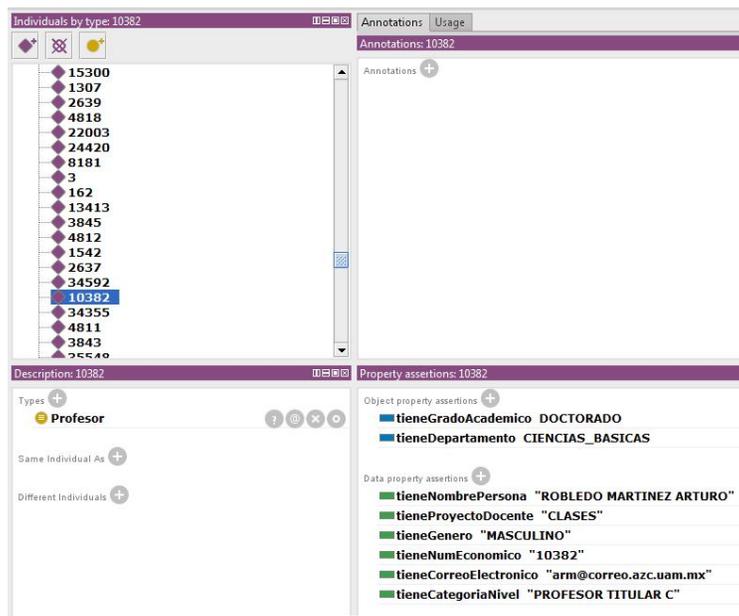


Figura 5 Propiedades instanciadas de un individuo de la clase Profesor.

Etapa 3. Integración del Sistema de Ontologías

Durante esta fase se realiza la importación de las ontologías individuales dentro de una ontología general para la gestión de horarios. La figura 6 muestra las ontologías importadas en la ontología Horario.

La figura 7 muestra los IRIs de las ontologías importadas en la ontología Horario.

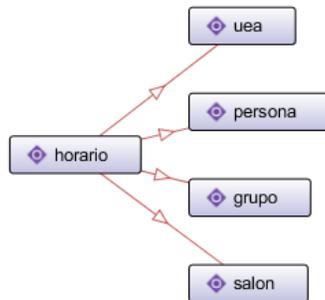


Figura 6 Ontologías importadas en el sistema de ontologías.

```
Ontology: <http://www.academia_uam.com/informacion_academica/ontologies/horario.owl>
Import: <http://www.academia_uam.com/informacion_academica/ontologies/grupo.owl>
Import: <http://www.academia_uam.com/informacion_academica/ontologies/persona.owl>
Import: <http://www.academia_uam.com/informacion_academica/ontologies/salon.owl>
Import: <http://www.academia_uam.com/informacion_academica/ontologies/uea.owl>
```

Figura 7 Ontologías importadas en la ontología **Horario**.

El sistema de ontologías Horario especifica que cada individuo de la clase Horario deberá tener exactamente un grupo, deberá tener exactamente una UEA, y que será identificado exactamente por un único Id. También hereda las restricciones de que los días de clases deberán ser de la clase Dia, que tendrá máximo un salón asignado y que tendrá máximo un profesor asignado, que debe definir un cupo del curso, y una hora de clase. Los axiomas de clase definidos son todos de tipo disyunción. La clase Horario es disjunta de las siguientes clases: Grupo, Dia, Departamento, GradoAcademico, Persona, Salon, Area, Exigencia, Tronco y Uea.

Etapas 4. Evaluación del Sistema de Ontologías

La evaluación del sistema de ontologías se realiza de dos formas: la primera se refiere a la verificación del cumplimiento de los principios de diseño y la segunda se refiere a la competencia del sistema de ontologías, es decir, si ésta es capaz de responder a cada una de las preguntas de competencia que se especificaron al principio de la metodología. La verificación de consistencia lógica del sistema de

ontologías se ejecuta para verificar que ninguna de las definiciones y axiomas de clase tienen contradicciones lógicas, o que algún individuo instanciado en la ontología contradice los axiomas definidos. Esta actividad final de razonamiento consiste en ejecutar las tareas de clasificación taxonómica, derivación de tipos y verificar consistencia.

3. Resultados

Como resultado de la ejecución de la metodología de diseño, en la figura 8 se muestran las métricas del sistema de ontologías para la gestión de horarios. Como se puede observar, el sistema consta de más de 20 clases o conceptos principales, un total de 12 propiedades entre objetos, 23 propiedades de datos y un total de 1069 individuos. Por el tipo de axiomas y restricciones empleado, el nivel de expresividad alcanzado es ALCOQ(D).

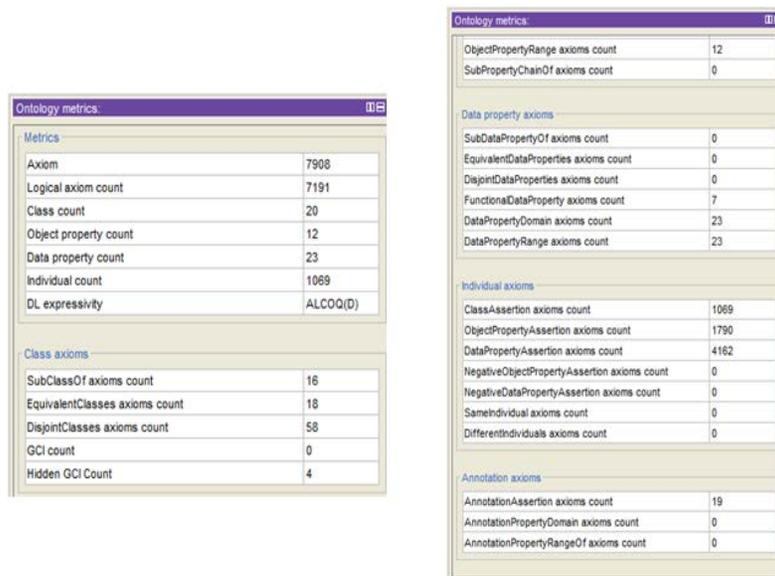


Figura 8 Métricas del sistema de ontologías.

En relación con la evaluación de los principios de diseño, la ontología cumple satisfactoriamente con los siguientes:

- Principio de diseño de claridad. Aseguramos que no hay una interpretación incorrecta al definir sólo las igualdades en las definiciones de las clases axiomáticas.

- Principio de diseño de coherencia. Después de la ejecución del Fact++, todos los individuos fueron clasificados correctamente.

Los principios de diseño más importantes se consideraron y verificaron a través de herramientas de Protege tales como los razonadores y el DLQuery.

Con respecto a la **evaluación de la competencia** del sistema de ontologías se tradujeron las preguntas de competencia al lenguaje de consulta SWRL. La tabla 1 muestra algunos ejemplos de preguntas de competencia traducidas al lenguaje de reglas.

Tabla 1 Traducción de las preguntas de competencia a reglas.

Pregunta	SWRL	Resultado
¿Qué UEAs imparte el profesor "Pérez" indicando salón, días y horas?	Profesor(?prof) ^ tieneNombrePersona(?prof, ?nombre) ^ swrlb:stringEqualsIgnoreCase(?nombre,"Pérez") ^ Horario(?hor) ^ tieneProfesor(?hor, ?prof) ^ tieneDia(?hor, ?dia) ^ tieneHora(?hor, ?horas) ^ tieneSalon(?hor, ?salon) -> sqwrl:select(?prof, ?nombre, ?salon, ?dia, ?horas)	Considerando que esta regla se ejecuta proporcionando el nombre del profesor, devuelve el horario que tiene actualmente el profesor.
¿Cuáles son los cursos que se imparten en el salón F305 los días martes?	Horario(?hor) ^ tieneSalon(?hor, ?salon) ^ swrlb:stringEqualsIgnoreCase(?salon,"F305") ^ tieneUea(?hor, ?uea) ^ tieneNombreUea(?uea, ?nombreUea) ^ tieneDia(?hor, ?dia) ^ tieneHora(?hor, ?horas) -> sqwrl:select(?salon, ?nombreUea, ?dia, ?horas)	Cuando esta regla se ejecuta devuelve la lista de Ueas que se imparten en el salón especificado en la consulta.
¿Cuáles son los cursos que tienen en su nombre la palabra "programación"?	Uea(?uea) ^ tieneNombreUea(?uea, ?nombreUea) ^ swrlb:contains(?nombreUea, "programación") -> sqwrl:select(?uea, ?nombreUea)	Esta regla devuelve las claves y nombres de las Ueas que contienen la palabra especificada en la consulta.
¿Cuáles son los cursos que se imparten los martes de 10:00 a 12:15?	Horario(?hor) ^ Dia(?dia) ^ tieneDia(?hor, ?dia) ^ swrlb:stringEqualsIgnoreCase(?dia,"MARTES") ^ tieneUea(?hor, ?uea) ^ tieneNombreUea(?uea, ?nombreUea) ^ tieneHora(?hor, ?horas) ^ swrlb:stringEqualsIgnoreCase(?horas,"10:00 - 12:15") -> sqwrl:select(?salon, ?nombreUea, ?dia, ?horas)	Esta regla devuelve el salón y los nombres de las Ueas que se imparten el día y la hora especificada en la consulta.

Adicionalmente se ejecutaron consultas utilizando el razonador Pellet y el lenguaje SPARQL en *Protege*. La siguiente pregunta es un ejemplo de la traducción de las pregunta: ¿Qué UEA imparte la Dra. Maricela Bravo, en qué horas y qué días? En la figura 9 se muestra el código de la pregunta en lenguaje SPARQL y la respuesta dada por el razonador Pellet.

Como se puede apreciar en la figura 9, la respuesta del razonador indica que la Dra. Bravo imparte dos UEA: Arquitectura e integración de aplicaciones empresariales los martes y jueves de las 14:30 a las 16:45 horas, y Programación orientada a servicios, también, los martes y los jueves, pero de las 16:45 a las 19:00 horas.

nomProf	nomUea	día	hora
"BRAVO CONTRERAS MARICELA CLAUDIA"	"ARQUITECTURA E INTEGRACION DE APLICACIONES EMPRESARIALES"	JUEVES	"14:30 - 16:45"
"BRAVO CONTRERAS MARICELA CLAUDIA"	"ARQUITECTURA E INTEGRACION DE APLICACIONES EMPRESARIALES"	MARTES	"14:30 - 16:45"
"BRAVO CONTRERAS MARICELA CLAUDIA"	"PROGRAMACION ORIENTADA A SERVICIOS"@	MARTES	"16:45 - 19:00"
"BRAVO CONTRERAS MARICELA CLAUDIA"	"PROGRAMACION ORIENTADA A SERVICIOS"@	JUEVES	"16:45 - 19:00"

Figura 9 Evaluación de la competencia del sistema de ontologías.

4. Discusión

El sistema de ontologías para la gestión de horarios se diseñó y construyó con una metodología que incorpora características de calidad desde la inepción de la ontología. En este artículo las ontologías son consideradas como módulos reutilizables, el objetivo de este enfoque de diseño es que los dueños y desarrolladores de las ontologías las puedan reutilizar de manera independiente

dentro de sus empresas para más aplicaciones. Otro de los requisitos relevantes que debe atender una metodología para el diseño y construcción de ontologías es que incorpore métodos y técnicas para abarcar todos los conceptos y relaciones semánticas necesarios para satisfacer las necesidades de los usuarios y de las aplicaciones de la ontología. Por ello, en esta metodología se hace énfasis en el uso de las preguntas de competencia para la identificación y construcción de relaciones. Durante el proceso de diseño se deben tomar decisiones que eventualmente facilitarán o afectarán la traducción de las preguntas a reglas, así que se debe incorporar un proceso de evaluación continua mediante el poblado y el uso de consultas ya sea con DL-Query, con SWRL o con SARQL.

5. Conclusiones

En este artículo se describió la metodología implementada para el diseño y construcción de un sistema de ontologías para la gestión de horarios. La metodología presentada es holística en el sentido de que abarca desde el diseño de clases hasta la integración y evaluación del sistema de ontologías. Esta metodología incorpora actividades para atender los principios de claridad, coherencia, modularidad y usabilidad. El sistema de ontologías implementado muestra que una metodología que incorpora características de calidad desde el principio logra producir ontologías consistentes, usables y reutilizables.

6. Bibliografía y Referencias

- [1] Fernández, M.; Gómez-Pérez, A., Juristo, N. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. Symposium on Ontological Engineering of AAAI. Stanford (California), March 1997.
- [2] Gómez-Pérez, A., Ontological engineering: A state of the art. Expert Update: Knowledge Based Systems and Applied Artificial Intelligence, 2(3), pp. 33-43, 1999.
- [3] Gruber, Thomas R., Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies 43.5, pp. 907-928, 1995.

- [4] Bernaras, A., Laresgoiti, I., & Corera, J., Building and Reusing Ontologies for Electrical Network Applications. In ECAI pp. 298-302, PITMAN, 1996.
- [5] Gómez Pérez, A. Knowledge Sharing and Reuse. In J. Liebowitz (Editor) Handbook of Expert Systems. CRC, 1998.
- [6] Gómez-Pérez, A., Fernández, M., & Vicente, A. D., Towards a method to conceptualize domain ontologies, 1996.
- [7] Grüninger, M., & Fox, M. S., Methodology for the Design and Evaluation of Ontologies, 1995.
- [8] Lenat, D. B., & Guha, R. V., Building large knowledge-based systems; representation and inference in the Cyc project. Addison-Wesley Longman Publishing Co., Inc, 1989.
- [9] M. Fernández-López, A. Gómez-Pérez, A. Pazos-Sierra, J. Pazos-Sierra, Building a chemical ontology using METHONTOLOGY and the ontology design environment, IEEE Intelligent Systems & their applications 4 (1), pp. 37–46, 1999.
- [10] Morbach, Jan, Andreas Wiesner, and Wolfgang Marquardt, OntoCAPE—A (re) usable ontology for computer-aided process engineering, Computers & Chemical Engineering 33.10, pp. 1546-1556, 2009.
- [11] Schreiber, G., Knowledge engineering and management: the CommonKADS methodology. MIT press, 2000.
- [12] Suárez-Figueroa, M. C., NeOn Methodology for building ontology networks: specification, scheduling and reuse, Doctoral dissertation, Informatica, 2010.
- [13] Uschold, M., & Gruninger, M., Ontologies: Principles, methods and applications. Knowledge engineering review, 11(2), pp. 93-136, 1996.
- [14] Uschold, M., & King, M., Towards a methodology for building ontologies, Edinburgh: Artificial Intelligence Applications Institute, University of Edinburgh, pp. 15-30, 1995.