



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

**“DEDUCCIÓN DE PATRONES DE ESTILO DE VIDA
USANDO ONTOLOGÍAS Y REGLAS SWRL”**

TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE:
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

ISMAEL MENA VIVEROS

ASESORA:

DRA. MARÍA JOSEFA SOMODEVILLA GARCÍA

H. PUEBLA DE ZARAGOZA, NOVIEMBRE DE 2015.

Agradecimientos

Agradezco...

... Agradezco antes que nada a Dios, por permitirme llegar hasta este punto con uno más de mis objetivos realizado y otorgarme el don de la salud.

... A mis hermanas, por su compañía, su cariño, su apoyo brindado, esas horas de largas charlas en las cuales se mezclan los consejos acerca de la vida universitaria y bromas.

...A mis amigos, amigos que se han convertido en hermanos, que han estado conmigo apoyando en las buenas y las malas, a veces más malas que buenas. Que con sus consejos, frases de aliento y ayuda he logrado una meta más.

... A mis padres, que como fruto de tantos sacrificios y esfuerzos hoy me acompañan al cumplir una meta más, que me han dado todo su amor, apoyo y cariño, que me dieron la vida y que siempre les estaré agradecido por todo lo que me han enseñado, y que se han tenido que privar de muchas cosas para que nosotros pudiésemos estudiar. A ustedes mis viejos, gracias.

... A la Dra. María Josefa Somodevilla García, mi asesora, por permitirme realizar este trabajo de tesis bajo su dirección, y con quien he trabajado estrechamente durante mucho tiempo; por haber confiado en mí, en mi trabajo y en mi capacidad y haberme guiado en muchos aspectos de mi vida profesional.

Este trabajo es por y para todos ustedes.

Resumen

El volumen de la información biomédica disponible en Internet crece constantemente día a día, con lo cual surge la necesidad de que esa información sea explotada, utilizada y compartida por los usuarios en las diferentes áreas del conocimiento. Las Enfermedades No Transmisibles (ENT o NCD por sus siglas en inglés), causan la muerte de más de 38 millones de personas cada año en todo el mundo, de acuerdo con la OMS. Derivado de esto, y considerando la magnitud y seriedad de este problema a nivel global, los gobiernos de todos los países han implementado diversas estrategias para alcanzar las metas de la reducción de factores de niveles de riesgo, así como la detección temprana y el efectivo tratamiento de las ENT.

En el proyecto se ilustra la utilización de reglas SWRL (Lenguaje de Reglas de la Web Semántica, por sus siglas en inglés) para la obtención de Patrones de Estilo de Vida. Se utilizará un sistema consistente de ontologías distintas, las cuales conceptualizan el conocimiento de áreas como personas, ENT, actividad física, nutrición, regiones geográficas y síntomas de enfermedades. Posteriormente, el sistema facilitará la integración de datos de dominios diferentes usando reglas SWRL para la toma de decisiones acerca de cambios en el estilo de vida.

Índice General

Agradecimientos	2
Resumen	3
Índice de figuras	7
Índice de tablas	9
Capítulo 1	10
Introducción	10
1.2 Objetivos Generales y Específicos.....	11
1.2.1 Objetivos Generales.....	11
1.2.2 Objetivos Específicos	11
1.3 Justificación	12
1.4 Organización del documento de tesis.....	13
Capítulo 2	14
Estado del Arte	14
2.1 Mezcla de Ontologías	14
2.2 Ontology Mapping.....	15
2.3 Integración de Ontologías por Puentes Semánticos.....	16
2.4 Lenguajes de ontologías.....	17
2.5 OWL (Lenguaje de Ontologías Web)	18
2.6 SWRL	19
2.7 Metodología general de desarrollo del sistema de ontologías	20
2.7.1 Metodología de Lenat y Guha (1990).....	20
2.7.2 Uschold & King (1995), Uschold & Grüninger (1996), Uschold (1996).....	20
2.7.3 Grüninger & Fox (1995).....	21
2.7.4 METHONTOLOGY (1996)	21
2.7.5 Metodología On-To-Knowledge (2001).....	22
2.7.6 Metodología NeON (2010).....	22
2.8 Metodología de Bravo, M. <i>et. al.</i>	23
2.8.1 Elicitación de términos	25
2.8.2 Integración (Axiomatización).....	26
2.8.3 Evaluación	27
Capítulo 3	29

Marco teórico	29
3.1 Web Semántica	29
3.1.1 Componentes de la Web Semántica	30
3.1.2 Estándares de la Web Semántica	31
3.2 Lenguajes de Ontologías Web	32
3.2.1 Antecedentes	32
3.3 OWL	33
3.3.1 Sublenguajes OWL	34
3.4 OWL2	37
3.5 ¿Qué es una ontología OWL? ¿Por qué desarrollarla?	38
3.5.1 Definición	38
3.6 Componentes de las ontologías OWL	39
3.6.1 Individuals - Individuos	39
3.6.2 Properties - Propiedades	40
3.6.3 Classes - Clases	41
3.7 Clases OWL	42
3.7.1 <i>Named Classes</i>	42
3.7.2 Intersection Classes (\cap)	42
3.7.3 Union Classes (\cup)	43
3.7.4 Clases de Complemento	44
3.7.5 Restrictions	44
3.7.6 <i>Restricciones Existenciales</i>	45
3.7.7 <i>Universal Restrictions</i>	45
3.7.8 <i>hasValue Restriction</i>	46
3.7.9 Enumeration Classes	47
3.8 Propiedades OWL	47
3.8.1 <i>Object Properties</i>	47
3.8.2 <i>Propiedades de tipo de dato</i>	48
3.8.3 <i>Propiedades de anotación</i>	48
3.8.4 Características de Propiedades	48
3.4 SWRL	51
3.5.1 Reglas	51
3.5.2 Tipos de átomos	51
3.5.3 Sintaxis legible por humanos	54

Capítulo 4	55
Desarrollo de la ontología propuesta para la deducción de patrones de estilo de vida ...	55
4.1 Análisis	55
4.1.1 Planteamiento del problema	55
4.1.2 Preguntas de investigación	55
4.1.3 Representación de la información	56
4.2 Diseño del sistema de ontologías (Nivel Macro)	56
4.3 Etapa de Diseño de las ontologías (Nivel Invidual)	57
4.4.1 Axiomatización (Subetapa de Integración de las ontologías).....	59
4.4.2 Subetapa de Evaluación	62
Capítulo 5	77
Conclusiones	77
5.1 Conclusiones finales	77
5.2 Trabajo futuro	78
Bibliografía	80

Índice de figuras

Fig. 2.1 Escenarios de construcción de ontologías.....	23
Fig. 2.2 Metodología general de construcción de ontologías.....	24
Fig. 2.3 Modelo base de construcción del sistema de ontologías.....	25
Fig. 3.1 World Wide Web Consortium	30
Fig. 3.2 Arquitectura de la Web Semántica (Fuente: https://en.wikipedia.org/wiki/Semantic_Web_Stack)	31
Fig. 3.3 Estructura de los lenguajes de ontologías para la web.....	33
Fig. 3.4 Estándar OWL2.....	37
Fig. 3.5 Representación de individuos	40
Fig. 3.6 Representación de propiedades entre individuos	40
Fig. 3.7 Representación de clases que contienen individuos.....	41
Fig. 3.8 Clase con nombre.....	42
Fig. 3.9 Clases de Intersección.....	43
Fig. 3.10 Clases de Unión.....	43
Fig. 3.11 Clases de Complemento.....	44
Fig. 3.12 Restricciones Existenciales	45
Fig. 3.13 Restricciones Universales	46
Fig. 3.14 Restricciones tieneValor	46
Fig. 3.15 Clases de Enumeración	47
Fig. 3.16 Propiedad transitiva.....	49
Fig. 3.17 Propiedad simétrica.....	50
Fig. 3.18 Propiedad reflexiva	50
Fig. 4.1 Relaciones semánticas del sistema de ontologías	57
Fig. 4.2 Herramienta Import Ontologies en Protégé 4.3.	58
Fig. 4.3 Estructura de clases del sistema de ontologías.....	58
Fig. 4.4 Grafo de ontologías del sistema	59
Fig. 4.5 Reglas SWRL del sistema.....	60
Fig. 4.6 Object properties incorporadas al sistema.....	61
Fig. 4.7 Clase <i>PersonWithNutritionHighInFat</i>	63
Fig. 4.8 Comparación entre Necessary conditions y Necessary & Sufficient Conditions	64
Fig. 4.9 Condiciones necesarias y suficientes del sistema	65
Fig. 4.10 Deduciendo miembros de clases por medio de reglas SWRL.	66
Fig. 4.11 Property Assertions para el individuo	67
Fig. 4.12 Descripción de las clases a las que pertenece el individuo	67
Fig. 4.13 Dedución de estilo de vida medio a través de reglas SWRL.....	68
Fig. 4.14 Dedución del patrón <i>PersonWithHighRiskOfSufferDisease</i>	69
Fig. 4.15 Dedución del patrón <i>RiskToSufferDiabetes</i>	70
Fig. 4.16 Ejecución del razonador.....	71
Fig. 4.17 Descripción del individuo Arthur.....	72

Fig. 4.18 Grafo del sistema.....	73
Fig. 4.19 Grafo de los patrones de estilo de vida encontrados	74
Fig. 4.20 Inferred Model del sistema.....	75
Fig. 4.21 Resultado de aplicar las particiones de valor aplicadas a la clase <i>PersonWithHighRiskOfSufferDisease</i>	76

Índice de tablas

Tabla 3.1 Compación entre los sublenguajes OWL	36
---	----

Capítulo 1

Introducción

El volumen de la información biomédica disponible en Internet crece constantemente día a día, con lo cual surge la necesidad de que esa información sea explotada, utilizada y compartida por los usuarios en las diferentes áreas del conocimiento. Las Enfermedades No Transmisibles (ENT¹ o NCD por sus siglas en inglés), causan la muerte de más de 38 millones de personas cada año en todo el mundo, de acuerdo con la OMS². Derivado de esto, y considerando la magnitud y seriedad de este problema a nivel global, los gobiernos de todos los países han implementado diversas estrategias para alcanzar las metas de la reducción de factores de niveles de riesgo, así como la detección temprana y el efectivo tratamiento de las ENT.

En el proyecto se ilustra la utilización de reglas SWRL³ para la obtención de Patrones de Estilo de Vida. Se utilizará un sistema consistente de ontologías distintas, las cuales conceptualizan el conocimiento de áreas como personas, ENT, actividad física, nutrición, regiones geográficas y síntomas de enfermedades. Posteriormente, el sistema facilitará la integración de datos de dominios diferentes usando reglas SWRL para la toma de decisiones acerca de cambios en el estilo de vida.

La necesidad de realizar este trabajo surge a partir del auge de enfermedades derivadas de la adquisición de ciertos hábitos dañinos para la salud, como el consumo de tabaco y de la falta de actividad física de las personas, dando como resultado el desarrollo de las Enfermedades No transmisibles: enfermedades que aquejan a gran parte de la población mundial y que tienen año con año consecuencias devastadoras en toda la población, i.e., la muerte de más de 38 millones de personas anualmente en todo el mundo. Las ENT afectan a personas de todos los grupos étnicos. Los factores de riesgo asociados a las ENT son el consumo de alimentos no sanos, dietas no balanceadas y la falta de actividad física.

¹ Enfermedades No Transmisibles

² <http://www.who.int/nmh/en/>

³ Semantic Web Rule Language – Lenguaje de Reglas de la Web Semántica, por sus siglas en inglés

Las ontologías han sido desarrolladas en un amplio rango de áreas de la ciencia, utilizadas para cubrir diversas áreas y tópicos: desde lingüística hasta el entorno biomédico. El uso de ontologías provee de mayor conocimiento acerca de un dominio o tema en específico, pues las ontologías son una conceptualización del dominio involucrado dentro del proceso.

Se presenta un sistema compuesto de seis ontologías distintas: *Person*, *OntoMex*, *Nutrition*, *Symptom*, *NCD* y *Physical Activity*, sobre el cual el lenguaje SWRL (Lenguaje de Reglas de la Web Semántica, por sus siglas en inglés) se utilizará para definir y aplicar reglas para manejar conceptos de las ontologías antes citadas. Esto permitirá la deducción de patrones relacionados y/o basados en estilos de vida.

Las ontologías se crearán de forma individual para posteriormente ser importadas desde la ontología que fungirá como sistema de ontologías. Este sistema contendrá a cada una de estas ontologías individuales, generando una nueva forma de lograr la integración de ontologías, como se propuso en [1] y [2]. Tales ontologías tienen la capacidad de ser escalables y pueden estar en constante actualización.

1.2 Objetivos Generales y Específicos

1.2.1 Objetivos Generales

- Deducir patrones relacionados y/o basados en estilos de vida.
- Generar conocimiento para la toma de decisiones acerca de cambios en el estilo de vida.

1.2.2 Objetivos Específicos

- Desarrollar ontologías individuales para su posterior importación en el sistema.
- Aplicar reglas SWRL en la deducción de patrones de estilos de vida por medio de la relación entre diversas clases y propiedades de objetos.

- Descubrir patrones de estilo de vida agregando condiciones necesarias y suficientes, para restringir los individuos para cumplir con propiedades específicas.

1.3 Justificación

Derivado de los altos índices de mortandad a nivel mundial causados por las ENT, surge este trabajo de tesis, añadiéndosele el gran impacto socioeconómico y los principales beneficios que aportara, los cuales se citan a continuación:

- En la población, al determinar los patrones de estilo de vida que siguen las personas, pudiendo posteriormente complementar con recomendaciones de cambio de sus hábitos para mejorar su estilo de vida.
- En el campo bioinformático, proporcionar información conceptualmente adecuada y de toma de decisiones acerca de las relaciones encontradas (patrones de estilos de vida) entre las personas y sus padecimientos y la influencia que tienen sobre éstos la alimentación y el tipo de actividad física.

Entre las principales aportaciones a la ciencia de este documento se pueden citar entre otras las siguientes:

- Aplicación de métodos de deducción de la Lógica Descriptiva, para poder predecir y describir información que se traduce en descubrimiento de hábitos dañinos, factores de riesgo y cambios en el estilo de vida de las personas.
- A partir de la especialización de las ontologías por medio de actualizaciones sucesivas revisiones, lograr una mejor integración de las citadas ontologías y mejorar el rendimiento del sistema, a partir de aumentar la exactitud de los conceptos e incorporar deducciones más adecuadas (patrones de estilo de vida). Esto producirá una mejor base de conocimiento para que un posterior sistema de recomendación logre determinar adecuadamente cambios en los patrones de estilo de vida.

1.4 Organización del documento de tesis

La tesis consta de cuatro capítulos comenzando con la introducción en la que se presenta un breve panorama del proyecto, objetivos y justificación del mismo, el marco teórico, en el cual se describen las áreas consideradas dentro de este trabajo, posteriormente en el estado del arte se detalla la investigación en la que este trabajo encaja y se da paso a una metodología de solución, donde se abordará la metodología de construcción del sistema de ontologías propuesta y el diseño e implementación de reglas SWRL en la deducción de patrones de estilo de vida, finalizando con las conclusiones obtenidas del desarrollo de esta tesis, así como una reseña de trabajo a futuro.

Capítulo 2

Estado del Arte

En la literatura, existen diversas definiciones para la integración de ontologías, entre ellas se puede citar la de (Pinto y Martins, 2001): “La integración de ontologías es el proceso mediante el cual se construye una ontología que trata o describe cierto dominio en particular mediante la reutilización de una o varias ontologías que describen distintos dominios”. Sin embargo, según (Sowa, 2000), la integración de ontologías es “El proceso de encontrar semejanzas entre dos ontologías distintas A y B, y derivando de tal proceso una nueva ontología C, proceso que facilita la interoperabilidad entre sistemas computacionales que son la base de las ontologías A y B. La nueva ontología C quizá pueda reemplazar a A o a B, o quizá pueda ser utilizada como un intermediario entre un sistema basado en la ontología A y otro sistema basado en la ontología B”.

La integración de ontologías puede ser lograda mediante:

- Mezcla de ontologías
- Mapeando ontologías locales a una ontología global
- Integración de ontologías locales por medio de puentes semánticos

2.1 Mezcla de Ontologías

Cuando el uso de ontologías comenzó a extenderse rápidamente y el número y complejidad de dichas ontologías incrementase, se llega a la conclusión que las ontologías deben integrarse.

La primera aproximación fue, el llamado ontology merging o mezcla de ontologías, esto es, mezclar todas las ontologías participantes para obtener una sola que contendría todo el conocimiento acumulado por las ontologías que se hubiesen mezclado [3]

Como ya se había revisado anteriormente la definición de (Sowa, 2000), se añade que se definen 3 niveles dependiendo del nivel de cambio necesario para derivar una nueva ontología de dos anteriores distintas, tales niveles son:

- 1) **Alineación de ontologías.** Es la forma más débil de integración, requiere cambios mínimos y soporta formas de interoperabilidad limitadas. Es útil para la clasificación y la recuperación de información, pero no soporta inferencias.
- 2) **Compatibilidad parcial de ontologías.** Requiere cambios para soportar una interoperabilidad más extendida, incluso pueda haber algunos conceptos o relaciones en un sistema o el otro que dificulten la interoperabilidad.
- 3) **Unificación de ontologías o Compatibilidad Total.** Quizá requiera cambios o reorganizaciones mayores de dos ontologías A y B, pero este hecho puede desembocar en una interoperabilidad completa y verificable en toda la ontología.

Aceptándose el tercer nivel como *Ontology Merge*, i.e. en donde la mezcla de ontologías se verifica y se cumple la interoperabilidad entre las mismas.

El uso de la técnica de la mezcla de ontologías es adecuado en sistemas de ontologías tradicionales, los cuales son generalmente pequeños o moderados en tamaño, estáticos y en los cuales la escalabilidad no es un requerimiento forzoso. Algunas herramientas que soportan la mezcla de ontologías son: SMART⁴, PROMPT/Anchor-PROMPT⁵, OntoMorph⁶, CMS⁷, FCA-Merge⁸ y Chimaera⁹.

2.2 Ontology Mapping

El mapeo de ontologías es el proceso de encontrar correspondencias entre los conceptos de dos ontologías específicas a través de reglas, este proceso es necesario para el intercambio de

⁴ Semantic Web information Management with automated Reasoning Tool (en inglés) Herramienta de Administración de Información de la Web Semántica

⁵ Ontología para producir automáticamente un conjunto de conceptos semánticamente relacionados de ontologías usando anclas

⁶ Sistema de Ontologías de Traducción de Conocimiento Simbólico

⁷ Ontología para representar Sistemas de Manejo de Contenidos

⁸ Mezcla de Ontologías utilizando una aproximación tipo bottom-up

⁹ Sistema para crear y mantener ontologías distribuidas en la Web

información y servicios dentro de una plataforma de agentes. Las ontologías pueden ser derivadas de ontologías globales o de “referencia”. En este caso, el mapeo de ontologías es mucho más fácil debido a que los conceptos provenientes de distintas ontologías que necesitan ser mapeados, son obtenidos de la misma ontología.

Algunos de los logros alcanzados en esta dirección son los reportados en los trabajos de Zhang, Yi, Wamberto Vasconcelos y Derek Sleeman: “Ontosearch: An ontology search engine” y también se pueden citar los trabajos de Kalfoglou y Schorlemmer con IF-MAP¹⁰.

2.3 Integración de Ontologías por Puentes Semánticos

En el tercer enfoque, el cual incluye a la integración de ontologías, las ontologías locales son mantenidas sin cambio alguno, sin embargo, se ligan a través de puentes semánticos, i.e., axiomas puente en lógica de primer orden. Los cuales definen los mapeos entre las ontologías. Generalmente, en este enfoque se tienen distintas etapas:

- ✓ Levantamiento y normalización
- ✓ Similitud
- ✓ Puenteo Semántico
- ✓ Ejecución
- ✓ Post-procesamiento

En la etapa de Puenteo Semántico, es donde se establece la aproximación, por medio de axiomas de lógica de primer orden. Los puentes semánticos describen relaciones entre entidades y están basados en similitudes que se establecen entre la ontología base y la ontología objetivo,

Esta aproximación es la más apropiada para sistemas altamente dinámicos y en constante crecimiento, por ejemplo, la Web Semántica. En este enfoque existen herramientas como CTXMatch, GLUE, MAFRA, LOM, QOMm ONION, OKMS, OMEN y P2P ontology

¹⁰ Método basado en el mapeo de ontologías enfocado en la Teoría del Flujo de Información

mapping, herramientas revisadas en [4], y en los trabajos de Barnickel, Weinand y Fluegge [5].

2.4 Lenguajes de ontologías

Una de las actividades que se están realizando en el campo de la Web Semántica es el desarrollo de motores de inferencia que soporten los lenguajes que se están definiendo para la Web Semántica. Podemos citar por ejemplo a Jena¹¹, un framework open source Java para el desarrollo de aplicaciones para Web Semántica y Linked Data, que incluye razonadores para RDF Schema¹² y OWL Lite (un subconjunto de OWL). Otro entorno de desarrollo para RDF que incluye un motor de inferencia es SESAME¹³.

OWL-DL (OWL Description Logic por sus siglas en inglés) es una parte importante de OWL, con menores capacidades de razonamiento que la lógica de primer orden, sin embargo, es en OWL-DL en la que es posible realizar un mejor razonamiento. Un ejemplo de razonador para OWL-DL es *Pellet*¹⁴. Pellet es un razonador implementado en Java que es especialmente útil en lo que a trabajar con múltiples ontologías se refiere.

Los lenguajes de ontologías RDF Schema y OWL ofrecen unas capacidades limitadas para razonamiento lógico. Por este motivo, se han desarrollado varios trabajos en los que se proponen lenguajes de reglas (es decir, permiten definir axiomas como parte de una base de conocimiento) para la Web Semántica. Uno de los primeros trabajos fue TRIPLE¹⁵. TRIPLE es un lenguaje de reglas basado en cláusulas de Horn que se define sobre RDF junto con un motor de inferencia capaz de razonar sobre modelos definidos en TRIPLE. Otra iniciativa muy importante es RuleML¹⁶. RuleML contiene aspectos de programación lógica, programación funcional y orientación a objetos.

¹¹ <https://jena.apache.org/>

¹² Estándar que provee un vocabulario para modelar datos RDF

¹³ Base de datos RDF open source con soporte para inferencias y consultas sobre RDFS

¹⁴ Razonador de la Web Semántica

¹⁵ Lenguaje de reglas RDF con Casos de Uso

¹⁶ RuleML es un lenguaje XML para la definición de reglas y orientado a su aplicación en la Web Semántica

2.5 OWL (Lenguaje de Ontologías Web)

OWL¹⁷ fue desarrollado como un lenguaje de ontologías para construir ontologías que provee descripciones de alto nivel del contenido de la Web. Dichas ontologías fueron construidas por medio de una jerarquía de clases o taxonomía que describe los conceptos de un dominio y relacionándose entre sí por medio de propiedades.

Otra de las funciones clave de OWL es que puede representar los datos como instancias de clases OWL también conocidos como individuos, que son los miembros de una clase y ofrece mecanismos para el razonamiento de los datos y su manipulación, así como también de un poderoso lenguaje axiomático para definir correctamente como interpretar los conceptos asentados en una ontología.

Una ontología consta de las siguientes partes:

1. **Clases e instancias:** Cualquier modelo que se construya contiene un conjunto de clases que representan conceptos relevantes al modelo. Por ejemplo, en una ontología que representase la organización de una escuela, "*Alumno*" y "*Profesor*" podrían ser clases de la ontología. Las instancias son entidades que pertenecen a una determinada clase. Por ejemplo, "*104C*" es una instancia de la clase "*Edificio*". Las clases se suelen organizar en una jerarquía, donde las instancias de una subclase pertenecen a la clase. Por ejemplo, podríamos tener en nuestra ontología la clase "*Espacio Físico*" de la que sería subclase "*Edificio*".
2. **Propiedades:** Establecen relaciones entre conceptos de la ontología. Por ejemplo, la propiedad "*tieneEdad*" relaciona una persona con su edad.
3. **Reglas:** Modelan sentencias lógicas que se verifican siempre. Se utilizan habitualmente para modelar conocimiento que no puede ser representado utilizando los otros 3 componentes. Se pueden utilizar reglas para 3 tipos de propósitos:
 - Generación de conocimiento: sirven para obtener nuevas sentencias lógicas a partir de la información almacenada en la base de conocimiento.

¹⁷ Web Ontology Language, por sus siglas en inglés

- Restricciones: nos indican propiedades que el modelo debe cumplir. Sirven para detectar inconsistencias.
- Reglas reactivas: determinan acciones a tomar por un sistema basado en conocimiento como consecuencia de que se cumplan ciertas condiciones. A las ontologías que no tienen reglas se les llama ontologías ligeras. Al tipo más sencillo de ontologías ligeras se les llama taxonomías y están formadas por una jerarquía de clases que representan los conceptos relevantes en el dominio, de forma que los conceptos más generales se encuentren en los niveles superiores de la jerarquía y los más particulares en los inferiores.

La decisión de utilizar o no reglas deben tomarse cuidadosamente, ya que las reglas nos proporcionan una gran capacidad expresiva pero también una mayor complejidad al razonar con la ontología, lo que puede ocasionar problemas de escalabilidad.

2.6 SWRL

SWRL¹⁸ es un lenguaje de expresión de reglas basado en OWL, que permite escribir reglas expresadas como conceptos OWL proporcionando capacidades de razonamiento.

Una regla SWRL es de la forma:

$$atomo \wedge atomo \dots \wedge atomo \rightarrow atomo \wedge atomo \quad (2.1)$$

Tanto el cuerpo como la cabeza de la regla son conjunciones positivas de los átomos; SWRL puede entender como significado que si todos los elementos en el antecedente son Verdaderos, entonces el consecuente debe ser también cierto. Un ejemplo de regla SWRL es el mostrado en la regla (2.2):

$$hasParent(?x1,?x2) \wedge hasBrother(?x2,?x3) \Rightarrow hasUncle(?x1,?x3) \quad (2.2)$$

¹⁸ Semantic Web Rule Language por sus siglas en inglés. Lenguaje de definición de reglas para la Web Semántica

La cual expresa lo siguiente: si una persona tiene como padre a otra, y ésta última es hermano de una tercera, entonces ésta última es tío de x2. Cualesquiera individuos que cumplan con la regla anterior, podrán ser identificados con el consecuente de la regla.

Los trabajos recientes se han concentrado en la adición de reglas SWRL a una ontología en OWL, para agregarle más expresividad a ésta. Permite a los usuarios escribir reglas que pueden expresarse en términos de clases OWL que pueden clasificar individuos OWL.

2.7 Metodología general de desarrollo del sistema de ontologías

Primeramente, se hará una revisión de las metodologías para el diseño de ontologías, las cuales se detallan a continuación:

2.7.1 Metodología de Lenat y Guha (1990)

Esta metodología es un método para construir Cyc, una base para representar conocimiento de sentido común. Los autores definieron una metodología de 3 pasos para la construcción de ontologías:

1. Extraer manualmente el conocimiento que está implícito en distintas fuentes (artículos, revistas, etc.)
2. Adquisición y codificación de nuevo conocimiento adquirido mediante herramientas.
3. Codificación automatizada mediante herramientas para extracción de conocimiento.

2.7.2 Uschold & King (1995), Uschold & Grüninger (1996), Uschold (1996)

Posteriormente, con base en la experiencia adquirida en el desarrollo de una ontología, surge la metodología de Uschold & King. De acuerdo con esta metodología, los procesos involucrados en la creación de ontologías son los siguientes:

1. Identificar el propósito y el ámbito de la ontología
2. Extraer los conceptos y las relaciones que involucradas entre tales conceptos y os términos que se utilizan para referirse a dichos conceptos y relaciones
3. Construir la ontología

2.7.3 Grüninger & Fox (1995)

Incorpora los conceptos de escenario y cuestiones de competencia, las respuestas de las cuestiones de competencia formuladas por los autores de la metodología son la base para extraer los conceptos principales de la ontología, propiedades relaciones y axiomas formales. En esta metodología se proponen los siguientes pasos:

- Definición de los escenarios motivadores, es decir, identificación de las posibles aplicaciones en las que la ontología será usada.
- Formulación de preguntas en lenguaje natural, a las que se les denomina *preguntas de competencia*, esto con la finalidad de determinar el ámbito de la ontología.
- Especificación de la terminología, es decir en base a las preguntas realizadas en el paso anterior, se define conceptos principales, relaciones, propiedades, etc.
- Formalización de las interrogantes.
- Especificación de axiomas formales.
- Verificación de la ontología.
- Dentro de esta metodología las cuestiones de competencia hacen referencias a consultas a las cuales la ontología debería responder.

2.7.4 METHONTOLOGY (1996)

Es una metodología creada por el Grupo de Ingeniería Ontológica de la Universidad Politécnica de Madrid [6]. La creación de la ontología puede empezar desde cero o en base a la reutilización de otras existentes. Methontology incluye la identificación del proceso de desarrollo de la ontología (calendario, control, aseguramiento de calidad, adquisición de conocimiento), un ciclo de vida basado en la evolución de prototipos, para la cual sigue los pasos definidos en el estándar IEEE 1074 de desarrollo de software. Sus pasos principales son:

- **Especificación.** Definir el alcance y granularidad de la ontología.
- **Conceptualización.** Permite organizar y estructurar el conocimiento adquirido mediante tablas, lenguaje UML, jerarquías etc.

- **Implementación.** Representa la formalización de la ontología, es decir pasar la conceptualización de la ontología a un lenguaje como RDF, OWL, etc.
- **Evaluación.** Comprobar el funcionamiento de la ontología.

2.7.5 Metodología On-To-Knowledge (2001)

Esta metodología [7], surgió como un proyecto del AIFB (Instituto de Informática Aplicada y Métodos Formales, por sus siglas en alemán) de la Facultad de Economía de la Universidad de Karlsruhe, Alemania. Mediante este proyecto se han desarrollado herramientas y métodos que soporten la administración de conocimiento, apoyado en una ontología compartible y usable.

Propone la construcción de la ontología teniendo en cuenta la forma en que la ontología será utilizada en las aplicaciones, consecuentemente, las ontologías desarrolladas bajo esta metodología son muy dependientes de la aplicación. Esta metodología aplica ontologías a la información electrónica con la finalidad de mejorar la administración de conocimiento e incluye los siguientes aspectos:

- Identificación de metas, las cuales deberán ser cumplidas por herramientas de gestión de conocimiento.
- Evaluación de la ontología a partir de casos de estudio

2.7.6 Metodología NeON (2010)

Propuesta en (Suarez-Figueroa, 2010) para el desarrollo de redes de ontologías con el objetivo de cubrir las carencias de las metodologías METHONTOLOGY y On-To-Knowledge. Dichas carencias son que ninguna de ellas trata simultáneamente el concepto de red de ontologías y dimensiones de colaboración, contexto y dinamismo. Está basada en la metodología de desarrollo de software XP (eXtreme Programming).

En la Figura 2.1 se muestran los escenarios de construcción de ontologías definidos por (Suarez-Figueroa, 2010).

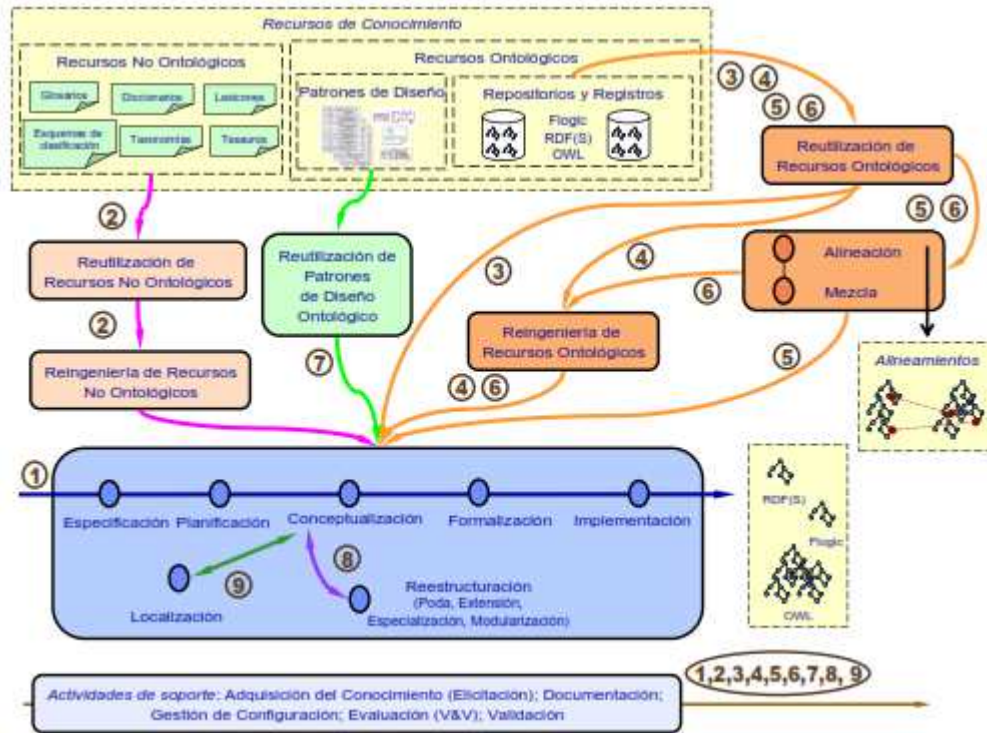


Fig. 2.1 Escenarios de construcción de ontologías

2.8 Metodología de Bravo, M. et. al.

Siguiendo para el desarrollo del sistema de ontologías se ha utilizado la metodología general de construcción que a continuación se detalla y fue presentada por (Bravo, M., et.al.) En la Figura 2.2, se muestra la metodología, teniéndose tres grandes etapas de construcción:

- Diseño del sistema de ontologías
- Diseño de las ontologías
- Integración y Evaluación de las ontologías



Fig. 2.2 Metodología general de construcción de ontologías

Y cumpliéndose con una serie de características, para agregar funcionalidad al sistema, entre las características se pueden citar las siguientes:

- Diseño orientado por dominios.
- Orientada a la reutilización.
- Modular.
- Incremental.
- Iterativa.
- Evaluación guiada por principios de diseño.
- Incorpora patrones de diseño de ontologías.

Con base en las características anteriores, es posible armar el modelo base de la metodología de general de (Bravo, M., *et. al.*), presentado en la Figura 2.3.

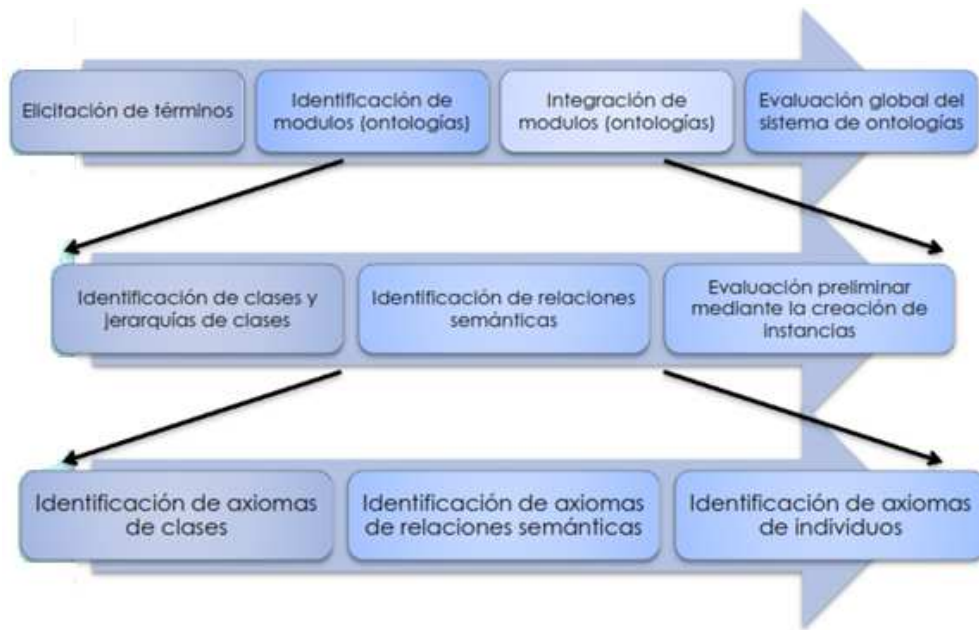


Fig. 2.3 Modelo base de construcción del sistema de ontologías

2.8.1 Elicitación de términos

La siguiente sub-etapa en el diseño de la metodología de construcción del sistema de ontologías es la elicitación de términos. Antes de ello se definirá primero lo que el termino elicitación significa.

Elicitación, proviene del griego *elicitus*, "inducido" y *elicere*, "atrapar", y en la ciencia de la computación se refiere al proceso de obtener información de un software a otro, de persona a persona o de una persona a una computadora.

En la etapa de elicitación de términos, se tendrá pues, que obtener información acerca de los conceptos relevantes de los dominios que se van a abstraer y de los que, posteriormente, se creará la ontología.

Como técnicas principales se tienen:

- Preguntas de competencia
- Planteamiento del caso a resolver

Por medio de elaborar preguntas de competencia se delimita el problema en el caso, y se define mejor la investigación, puesto que formulando una pregunta específica se sabe con certeza que técnica o que elemento utilizar para responder tal pregunta.

2.8.2 Integración (Axiomatización)

Finalmente, la última fase de la metodología que aquí se presenta y mediante la cual se diseñará y construirá el sistema propuesto en el Capítulo 1, será la de Integración y Evaluación de las Ontologías.

La etapa de integración y evaluación se ha dividido en dos sub etapas para una mejor comprensión y desarrollo de los componentes del sistema. En esta sub etapa, se integrarán las ontologías en el sistema y se realizará la axiomatización.

Identificación de relaciones semánticas

En esta fase se prosigue con la identificación de las relaciones semánticas, esto es, identificar las *object properties*, *datatype properties*, y relaciones entre las clases y ontologías. Dentro de ésta fase se realizarán los siguientes pasos:

Identificación de relaciones jerárquicas.

Se verificarán las relaciones entre clases, por ejemplo, la relación *Person followsA Nutrition*, es una relación jerárquica entre las ontologías *Person* y *Nutrition*, que, al ser importadas al sistema pasan a convertirse en clases, así como también verificar las demás relaciones que tengan parte en el diseño conceptual del sistema en sí. Para el caso de esta tesis, este paso se verifica en el diseño de las relaciones semánticas del sistema, mostrado con más detalle en la Figura 4.1, Sección 4.2.

Identificación de relaciones tipo datatype properties.

En este paso cabe realizar la pregunta ¿Cuáles son las características necesarias para definir un concepto? Determinar las características de los conceptos que formarán parte,

inicialmente de una ontología en particular y posteriormente, que se involucrarán con el sistema y que han de tener parte en él.

Como *datatype properties* principales inicialmente se puede tener para el caso de una ontología de una persona los siguientes: edad, altura, peso, etc.

Identificación de relaciones tipo object properties

En este paso es necesario analizar nuevamente las preguntas de competencia formuladas e identificar aquellas relaciones que pudieran haberse omitido. Por ejemplo, después de identificar y extraer los conceptos principales, que serán mapeados como clases, identificar aquellas declaraciones que se establezcan entre tales conceptos.

Identificación de características de las relaciones

Determinar a cuales tipos de características de las propiedades OWL (*functional, inverse functional, symmetric* o *transitive*) pertenecen los axiomas (*object properties*) que están siendo utilizados en el sistema.

Por ejemplo, para el ejemplo de *object property* utilizada anteriormente, **followsA** y que interactúa con las clases *Person* y *Nutrition*, le corresponde la característica *functional*, puesto que para cada uno de los miembros de ambas clases tenemos una relación uno a uno, i.e., para una persona determinada, esa persona sigue un tipo de nutrición determinado.

2.8.3 Evaluación

En la segunda sub etapa de Integración y Evaluación, se procederá a probar la consistencia del sistema de ontologías por medio de la ejecución del razonador *Pellet*, así como también la consistencia de las reglas SWRL generadas y que éstas generen los patrones de estilo de vida correctamente.

Posteriormente, se analizan los individuos miembro de las clases inferidas, axiomas de condiciones necesarias y suficientes, particiones de valor y la arquitectura de clases del sistema.

En el siguiente capítulo se introduce el marco teórico que es el contexto en el que está inmersa el área de las ontologías, la web semántica y el lenguaje SWRL.

Capítulo 3

Marco teórico

En este capítulo se presentan los fundamentos teóricos que sustentan el desarrollo del documento de tesis, se iniciará exponiendo los fundamentos de la Web Semántica, los lenguajes de ontologías para la web, la lógica descriptiva y el estándar OWL, seguidos del estándar SWRL. Cabe añadir que este capítulo al incluir fundamentos teórico-técnicos, resulta ser de los más extensos, si no el que más.

3.1 Web Semántica

La Web actual (la WWW¹⁹) es una gran biblioteca de documentos interrelacionados que son transferidos y/o compartidos por computadoras alrededor de todo el mundo y presentados a las personas. Se ha tenido un gran desarrollo, partiendo desde los sistemas con enlaces, en la que cualquiera puede contribuir. Esto significa que, generalmente, la calidad de la información o incluso su persistencia puede no estar garantizada. La WWW contiene demasiada información y conocimiento, sin embargo, los dispositivos usualmente presentan esos documentos describiendo el conocimiento. Las personas deben disponer de los dispositivos e interpretar la información por ellas mismas.

La Web Semántica es un esfuerzo del W3C²⁰ (World Wide Web Consortium) para crear tecnologías que interactúen con la información contenida en la WWW, la interpreten y reutilicen para ayudar a las personas a encontrar el conocimiento requerido y que dicho conocimiento sea compartido y enriquecido, mediante distintas formas, aplicaciones informáticas a través de la comunidad y empresas.

¹⁹ World Wide Web

²⁰ <http://www.w3.org>



Fig. 3.1 World Wide Web Consortium

El objetivo principal de la Web Semántica es proveer de una extensión a la Web que contenga metadatos (datos que “hablan” acerca de los datos) semánticos y ontológicos a la Web para permitir la creación de un conjunto común de herramientas para facilitar la compartición de la información.

Dadas las limitaciones de HTML para los propósitos descritos anteriormente, los grupos de trabajo encargados y que participan en la Web Semántica, han propuesto diversas soluciones para , como ejemplos se pueden citar: RDF, RDFS, OWL, SPARQL, etc., que se han convertido en estándares y que son utilizados en tareas como consulta (SPARQL), inferencia (OWL), linked data, etc.

3.1.1 Componentes de la Web Semántica

Como ya ha sido explicado, la Web Semántica es un conjunto de tecnologías y estándares que la habilitan. Estas tecnologías están especificadas como Estándares W3C y se mencionan a continuación:

- Resource Description Framework (RDF)
- RDF Schema (RDFS)
- Simple Knowledge Organization System (SKOS)
- SPARQL
- Notation3
- N-Triples
- Turtle
- Web Ontology Language (OWL)
- Rule Interchange Format (RIF)

La Pila de la Web Semántica²¹ (Semantic Web Stack, Semantic Web Cake, Semantic Web Layer Cake) ilustra la arquitectura de la Web Semántica y se muestra en la Figura 3.2.

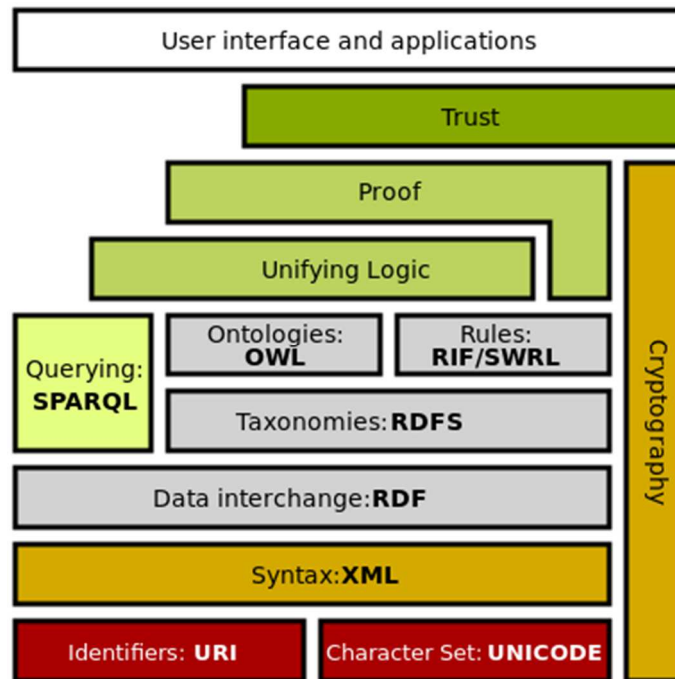


Fig. 3.2 Arquitectura de la Web Semántica (Fuente: https://en.wikipedia.org/wiki/Semantic_Web_Stack)

3.1.2 Estándares de la Web Semántica

RDF Schema

Esquema de Marco de trabajo de Descripción de Recursos, (RDFS por sus siglas en inglés) comúnmente abreviado RDFS, RDF(S), RDF-S o RDF/S es un conjunto de clases con ciertas propiedades que utiliza el estándar RDF y su modelo de datos de KR, provee elementos básicos para la representación de ontologías, a veces llamados vocabularios RDF, pensados para desarrollar estructuras RDF. Estos datos pueden ser salvados en Triples y es posible interactuar con ellos por medio de **SPARQL**²². En 2004, como parte de una revisión mayor de RDF, RDFS se convirtió en una Recomendación W3C.

²¹ https://en.wikipedia.org/wiki/Semantic_Web_Stack

²² SPARQL Protocol and RDF Query Language, lenguaje de consultas de RDF

Aunque RDFS provee de soporte parcial a especificaciones de ontologías, la necesidad de una mayor expresividad de la ontología se había hecho cada vez mayor.

Web-Ontology Working Group

El W3C creó el Web-Ontology Working Group, como parte de la Actividad de Web Semántica (SWA por sus siglas en inglés). Comenzó sus actividades en Noviembre de 2001, teniendo a James Hendler y Guus Schreiber como co-líderes. Los trabajos de este grupo dieron paso a la primera versión del estándar OWL, cuyo primer borrador fue publicado en 2002 y que se convirtió en Recomendación (también conocido como Estándar) W3C en 2004.

OWL Working Group

En 2005, en el marco del workshop Experiencias OWL, hubo consenso en que debido a los avances recientes en lógica descriptiva podría permitir una mejor revisión para satisfacer las necesidades de los usuarios de forma más completa y manteniendo una buena calidad en lo que a funciones computacionales se refiere. Los trabajos para mejorar OWL iniciaron con la versión OWL1.1 y finalizaron con el lanzamiento de **OWL2**²³ y que más tarde se convertiría en una Recomendación W3C en Octubre de 2009.

3.2 Lenguajes de Ontologías Web

3.2.1 Antecedentes

Desde la década de los 90's han existido numerosos esfuerzos en lo que se define como Representación del conocimiento (KR²⁴ por sus siglas en inglés) que actualmente se denomina Representación del conocimiento y razonamiento (Knowledge Representation and Reasoning) y que se define como un campo de la Inteligencia Artificial dedicado a representar información acerca del mundo de forma que una computadora pueda utilizar dicha información para solucionar tareas complejas y como esa idea podría ser útil en la WWW.

²³ <http://www.w3.org/TR/2012/REC-owl2-primer-20121211/>

²⁴ <http://groups.csail.mit.edu/medg/ftp/psz/k-rep.html>

En el año 2000, DARPA²⁵, inicio el desarrollo de DAML²⁶, por un equipo liderado por James Hendler. En marzo de 2001, el Joint EU/US Committee on Agent Markup Languages, Comité Unido UE/USA Sobre Lenguajes de Agentes de Marcado, decidió que DAML sería mezclado con OIL²⁷ (un lenguaje de KR basado en XML). Posteriormente el Grupo de Trabajo Unido UE/USA Sobre Lenguajes de Agentes de Marcado convino en desarrollar a **DAML+OIL** como un lenguaje de ontologías web. DAML+OIL fue planeado como una capa sobre **RDFS** con semántica formal basada en Lógica Descriptiva.

DAML+OIL es una gran influencia en el desarrollo de OWL. En la Figura 3.3 se muestra la relación entre los lenguajes de ontologías para web y los estándares de Web Semántica.

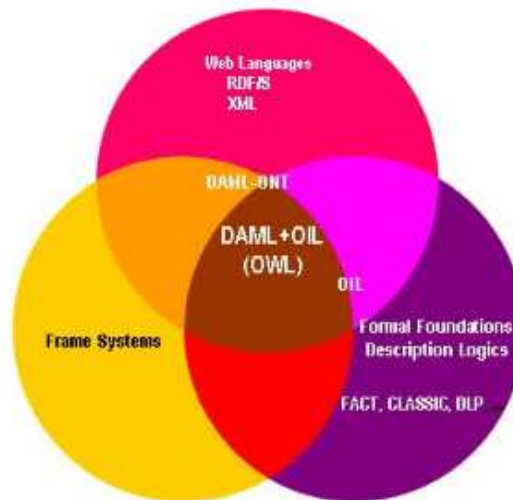


Fig. 3.3 Estructura de los lenguajes de ontologías para la web [19]

3.3 OWL

El estándar **OWL** (Web Ontology Language) del cual ya se había esbozado una noción de lo que significa en la Sección 2.5, es la base de las ontologías, puesto que es su lenguaje de creación. OWL fue desarrollado por el Web-Ontology Working Group y lanzado como Recomendación W3C en 2009. Como se detalla a continuación, tiene tres variantes o “especies”.

²⁵ Agencia de Proyectos de Investigación Avanzados de Defensa

²⁶ DARPA Agent Markup Language,

²⁷ Ontology Inference Layer or Ontology Interchange Language

3.3.1 Sublenguajes OWL

El lenguaje OWL provee de tres variantes, con expresividad en aumento que están pensados para distintas comunidades de usuarios y desarrolladores.

OWL Lite

Fue pensado y desarrollado originalmente para ofrecer soporte a aquellos usuarios que necesiten de una jerarquía de clasificación y restricciones simples. Por ejemplo, mientras se permite la cardinalidad, solo es posible elegir valores entre 0 y 1. Se esperaba que fuese más simple el proveer soporte a OWL Lite por medio de una herramienta que en sus parientes más expresivos, permitiendo una ruta de migración rápida para sistemas que utilizaran tesauros y otras taxonomías.

En la práctica, sin embargo, muchas de las restricciones de expresividad de OWL Lite representan poco más que “inconvenientes sintácticos” puesto que muchas de las construcciones de OWL DL, se pueden crear mediante la combinación de características complejas de OWL Lite. Probando con lo anterior, que el desarrollo de herramientas para OWL Lite es más complicado que en OWL DL y que su uso no esté muy extendido.

OWL DL

Diseñado para proveer la máxima expresividad mientras se cumple con completitud computacional (esto es, que todas las deducciones o implicaciones sean computables), decidibilidad (que el cómputo de un problema finalice en un tiempo finito) y la disponibilidad de algoritmos de razonamiento. OWL DL incluye todas las construcciones de OWL, con la limitante que algunas solo pueden ser usadas bajo ciertas condiciones. Ofrece una sintaxis amigable y las inferencias realizadas son siempre decidibles. Cuenta con ciertas restricciones para las construcciones OWL entre las cuales se pueden citar las siguientes:

- Se requiere de separación entre clases, *datatype properties*, *object properties*, *annotation properties*, propiedades de la ontología,(i.e., versión, imports, etc), individuos, valores de datos y los *built-in* o restricciones precargadas que se encuentran en el core de OWL.

- En OWL los conjuntos de *datatype properties* y *object properties* son disjuntos. Esto implica que las siguientes características de propiedades:
 - *Inverse Of*,
 - *Inverse Functional*,
 - *Symmetric* y
 - *Transitive*

No pueden ser especificadas para *datatype properties*.

- Los axiomas sobre la diferencia o equivalencia de individuos deben ser aplicados a individuos con nombre.

OWL DL debe su nombre a la correspondencia con la Lógica Descriptiva (DL, Description Logic, por sus siglas en inglés).

OWL Full

Está orientado a usuarios que requieren de la máxima expresividad y la libertad sintáctica de RDF/XML, i.e., no hay límite en su representación, con la excepción de que no provee garantías computacionales (pueden existir inferencias no decidibles).

OWL Full no podría ser considerado como sublenguaje OWL, puesto que contiene en si a todo el lenguaje OWL y como ya se ha explicado antes todas las construcciones de RDF sin ninguna restricción. Por ejemplo, en OWL Full el recurso `owl:Class` es equivalente a `rdfs:Class`. Esto es diferente a OWL Lite y OWL DL, puesto que en ambos `owl:Class` es una subclase propia de `rdfs:Class`, lo que significa que, no todas las clases RDF son clases OWL en OWL Lite y OWL DL. Por ejemplo, en OWL es perfectamente válido tener un identificador “*Fokker-100*” el cual trabaja tanto como una clase nombrada (denotando el conjunto de aviones Fokker 100 que es un modelo de avión de pasajeros bimotor) como un individuo nombrado (una instancia de la clase *TipoDeAvion*).

En la Tabla 3.1 se observa la diferencia entre cada sublenguaje OWL de acuerdo a distintos componentes tales como: Compatibilidad con RDFS, Restricciones en las definiciones de las clases, Descripciones de las clases, etc.

Tabla 3.1 Compación entre los sublenguajes OWL

	Lite	DL	Full
Compatibilidad con RDF	Teóricamente, ningún documento RDF es compatible con OWL Lite	Teóricamente, ningún documento RDF es compatible con OWL DL	Todos los documentos rdf validos son compatibles con OWL Full
Restricciones con la definición de clases	Requiere separación entre clases, instancias, propiedades y valores de datos	Requiere separación entre clases, instancias, propiedades y valores de datos	Las clases pueden ser instancias o propiedades al mismo tiempo, por ejemplo tener una clase Fokker-100 y un identificador con el mismo nombre
Mezcla RDF	Restringe la mezcla de RDF o entre RDF y OWL	Restringe la mezcla de RDF o entre RDF y OWL	Permite la mezcla de RDF o entre RDF y OWL
Descripciones de clases	La única descripción de clase disponible es IntersectionOf	Las clases pueden estar descritas como UnionOf , ComplementOf, IntersectionOf, y enumeration Eg: las clases pueden estar descritas de forma exhaustiva por sus instancias	Las clases pueden estar descritas como UnionOf , ComplementOf, IntersectionOf, y enumeration Eg: las clases pueden estar descritas de forma exhaustiva por sus instancias
Restricciones de Cardinalidad	Cardinality: 0/1 MinCardinality: 0/1 MaxCardinality: 0/1	Cardinality >= 0 MaxCardinality >= 0 MinCardinality >= 0	Cardinality >= 0 MaxCardinality >= 0 MinCardinality >= 0
Restricciones de valor	owl:allValuesFrom Owl:someValuesFrom Tipo de objeto para owl:valueFrom debería ser un nombre de clase o un identificador de clase	Owl:allValuesFrom Owl:someValueFrom Owl:hasValue	Owl:allValuesFrom Owl:someValueFrom Owl:hasValue
Metamodelado	No permite metamodelado	No permite metamodelado	Permite metamodelado: las construcciones en RDF y OWL pueden ser aumentadas o redefinidas
Clase	La clase OWL:class es subclase de RDFS:class	La clase OWL:class es subclase de RDFS:class	RDFS:class y OWL:class son equivalentes

3.4 OWL2

Existe una nueva versión del estándar **OWL**, la cual se ha denominado **OWL2**, la cual se va a utilizar en este trabajo de tesis, y en la cual ciertos elementos han cambiado con respecto del estándar **OWL 1**²⁸. En la Figura 3.4 se muestra la estructura de OWL 2.

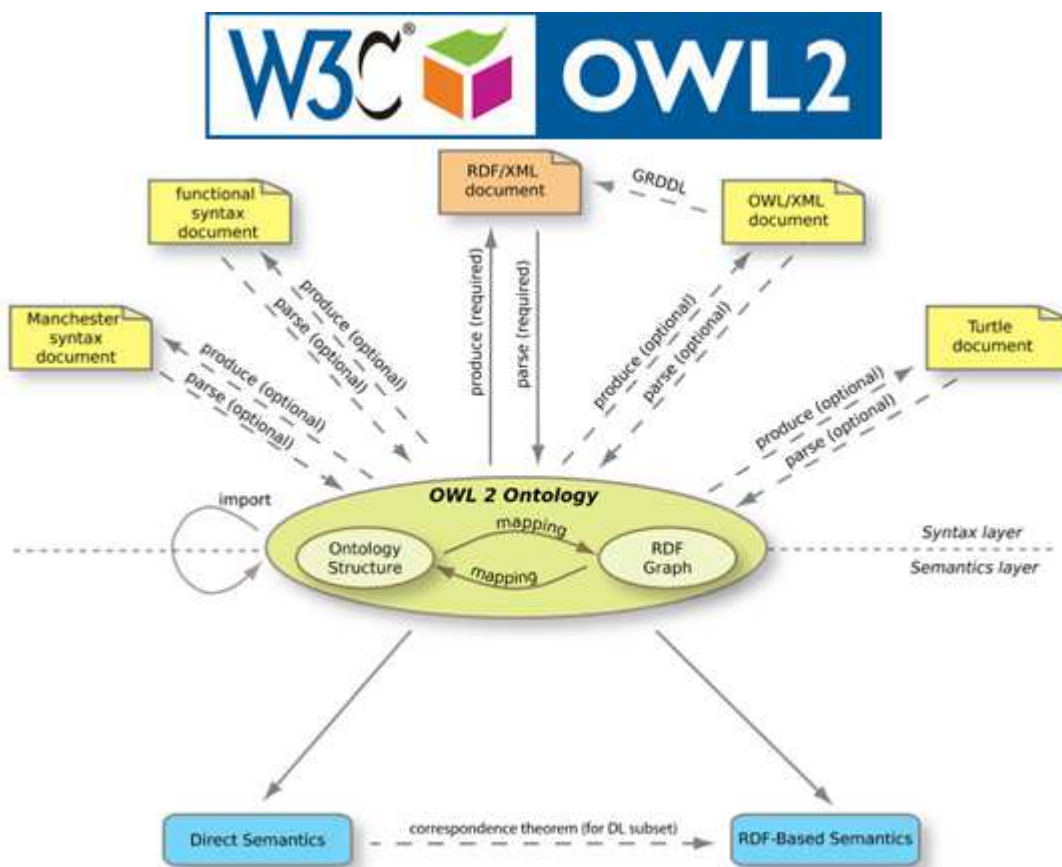


Fig. 3.4 Estándar OWL2

Entre las nuevas funcionalidades agregadas destacan:

1. Azúcar sintáctico para que algunas declaraciones comunes sean más fáciles de expresar (en clases disjuntas, por medio del axioma **DisjointClasses** es posible declarar que dos individuos no pertenecen a una clase al mismo tiempo, *NegativeObjectPropertyAssertion*, etc)
2. Nuevas construcciones que incrementan la expresividad

²⁸ <http://www.w3.org/TR/2004/REC-owl-guide-20040210>

3. Soporte extendido para *datatypes*
4. Capacidades de metamodelado más simples
5. Capacidades de anotación extendidas
6. Otras innovaciones

3.5 ¿Qué es una ontología OWL? ¿Por qué desarrollarla?

3.5.1 Definición

Existen diversas definiciones para el término ontología, de acuerdo al campo del conocimiento sobre el cual se estudie dicho término. Por ejemplo, en filosofía, una ontología se refiere a suposiciones sobre lo que constituye la realidad.

En el contexto de la ciencia de computación, el término ontología se refiere a una representación formal y explícita de los conceptos y relaciones que se establecen entre ellos de un dominio dado [8,9]

Noy y McGuinness [9], definen una ontología como: “una descripción formal y explícita de los conceptos de un dominio de discurso”.

Las ontologías un dominio concreto, son usadas entre otras cosas para almacenar conocimiento acerca de algún dominio en particular. Una ontología describe los conceptos que intervienen en el dominio o que forman parte de él y las relaciones que se establecen entre dichos conceptos [8].

¿Por qué se desarrollan las ontologías? Algunas de las razones principales son:

- *Compartir el conocimiento* común de la estructura de la información que se intercambia entre las personas o entre agentes de software.
- Reutilización del conocimiento de un dominio
- Hacer suposiciones claras y detalladas de un dominio
- Analizar el conocimiento de un dominio.

Encontrándose que es “*Compartir el conocimiento común de la estructura de la información que se intercambia entre las personas o entre agentes de software*” la principal razón para la creación de ontologías (Musen 1992; Gruber 1993).

En ocasiones, la ontología en si no es el principal objetivo. El desarrollo de una ontología es similar a definir un conjunto de datos y su estructura, para que otros programas puedan utilizarlos.

3.6 Componentes de las ontologías OWL

Las ontologías OWL tienen componentes similares a las ontologías basadas en los marcos de trabajo de Protégé. En OWL existen `Individuals` (Individuos), `Properties` (Propiedades) y `Classes` (Clases), que corresponden con los elementos `Individuals` (Individuos), `Properties` (Propiedades) y `Classes` (Clases) de Protégé.

3.6.1 Individuals - Individuos

Los individuos representan objetos del dominio en el cual estamos interesados²⁹. Una diferencia importante entre Protégé y OWL es que OWL no utiliza UNA (Suposición de Nombres Únicos, por sus siglas en inglés). Esto significa que dos nombres distintos, podrían referirse al mismo individuo. Por ejemplo, “Reina Isabel”, “La Reina” e “Isabel de Windsor” *podrían* referirse todos al mismo individuo.

En OWL debe declararse explícitamente que algún individuo es igual a otro o distinto a otro, en otro caso un individuo *podría* ser igual a otro o *podría* ser distinto a otro. La Figura 3.5 muestra una representación de algunos individuos en algún dominio.

²⁹ También conocido como el dominio de discurso



Fig. 3.5 Representación de individuos

Los individuos son también conocidos como *instancias* y pueden ser referidos como ‘instancias de clases’.

3.6.2 Properties - Propiedades

Las propiedades son relaciones binarias³⁰ sobre individuos, i.e. las propiedades vinculan a dos individuos. Por ejemplo, la propiedad **livesIn** vincula a los individuos Matthew y England, i.e. Matthew **livesIn** England, Matthew vive en Inglaterra, o la propiedad **hasSibling** entre los individuos Matthew y Gemma. Las propiedades pueden tener inversas como en el caso de la propiedad **hasOwner**, tiene una inversa llamada **isOwnedBy**.

Las propiedades también pueden estar limitadas a tener un solo valor, es decir, puede tratarse de *propiedades funcionales*; o también puede haber propiedades *simétricas* o *transitivas*. Estas ‘Características de propiedades’ serán revisadas con más detalle en la Subsección 3.7.4. En la Figura 3.6 muestra la representación de propiedades entre individuos.

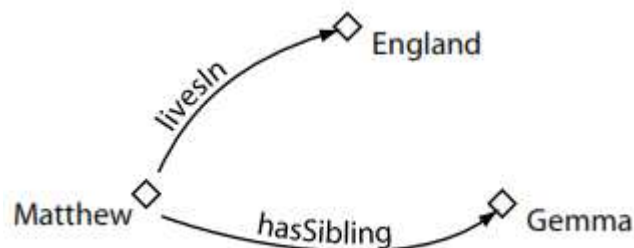


Fig. 3.6 Representación de propiedades entre individuos

³⁰ Una relación binaria es una relación entre *dos* objetos

3.6.3 Classes - Clases

Las clases OWL son interpretadas como conjuntos que contienen individuos. Son descritas utilizando descripciones formales que declaran expresamente los requerimientos para que un individuo cualquiera sea miembro de las clases. Por ejemplo, la clase *Gato*, contendrá a todos los individuos que son gatos en nuestro dominio de interés³¹. Las clases pueden ser organizadas en una jerarquía de superclase-subclase, jerarquía que es también conocida como *taxonomía*. Las subclases especializan a las superclases. Por ejemplo, supongamos que se tiene una clase *Animal* aparte de la clase *Gato*, *Gato* puede ser subclase de *Animal*, entonces *Animal* es superclase de *Gato*.

Una de las características clave de OWL-DL es que la relación superclase-subclase, de la que ya se ha hablado, puede ser computada automáticamente por un *razonador*, el cual se detallará más adelante. En la Figura 3.7 es mostrada una representación de algunas clases, las cuales contienen individuos.

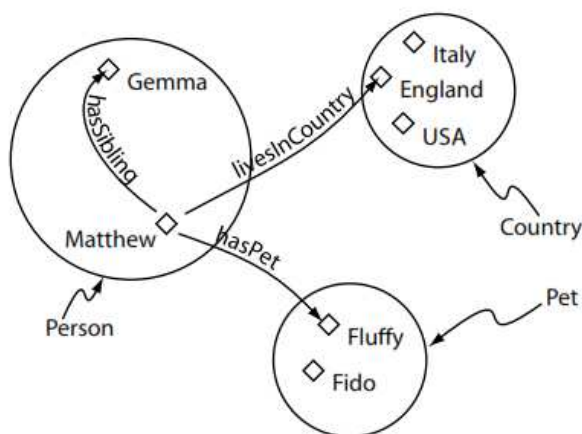


Fig. 3.7 Representación de clases que contienen individuos

En OWL las clases son construidas a partir de descripciones que especifican las condiciones que debe cumplir un individuo para ser miembro o pertenecer a dichas clases. A continuación se detallarán las maneras de describir una clase que ofrece OWL.

³¹ Un individuo puede pertenecer a más de una clase

3.7 Clases OWL

OWL soporta 6 maneras de describir una clase (*class description*), la más simple es la *Named Class* o Clase con nombre. En OWL, todos los objetos derivan de **Thing**. En OWL DL y OWL Lite un individuo no puede ser al mismo tiempo clase e individuo.

3.7.1 Named Classes

Las clases nombradas, también conocidas como clases con nombre son el tipo de descripción de clases más simple y es la que describe individuos que pertenecen a una clase. En la Figura 3.8 se muestra un ejemplo de una *Named Class*.

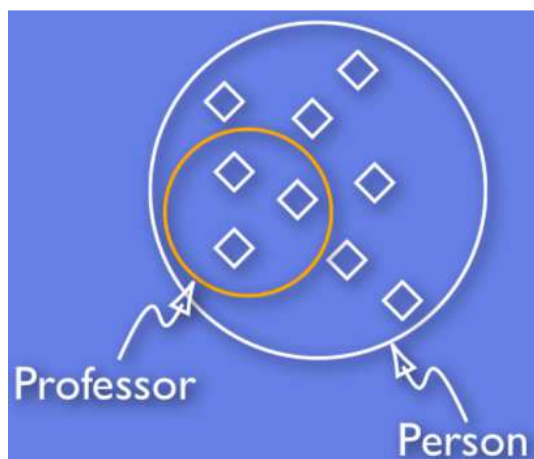


Fig. 3.8 Clase con nombre

3.7.2 Intersection Classes (\cap)

Las **clases de intersección** son el resultado de combinar dos o más clases por medio del operador de intersección (**AND**). En Protégé se utiliza el símbolo de intersección **and** para simbolizar estas clases llamadas anónimas. Por ejemplo, supóngase que existen dos clases llamadas *Human* y *Female*, al realizar la intersección de ambas clases se obtiene una clase *anónima* que contiene a los individuos que son miembros de la clase *Human* y de la clase *Female*.

La clase de intersección anónima utilizada anteriormente puede también utilizarse para otra descripción de clase. Por ejemplo, se requiere construir la descripción de la clase *Woman*, es posible especificar que *Woman* es subclase de la clase anónima descrita por la intersección

entre *Human* y *Female*, i.e., *Woman* es subclase de *Human* y *Female*. En la Figura 3.9 aparece la representación gráfica de una clase de intersección anónima.

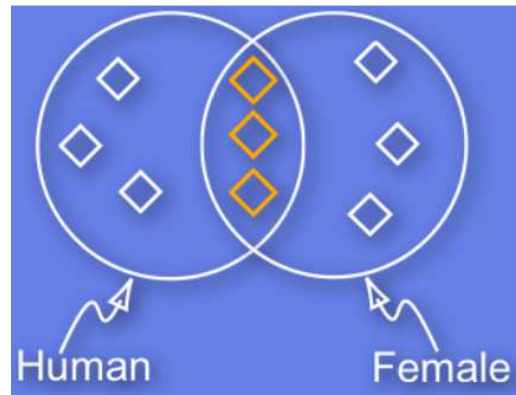


Fig. 3.9 Clases de Intersección

3.7.3 Union Classes (U)

Las **clases de unión** están formadas por la combinación de dos o más clases por medio del operador de unión (OR). En Protégé se utiliza el símbolo de intersección **or**. Por ejemplo supóngase que existen dos clases: *Student* y *Professors*, al realizarse la unión de las clases *Student* y *Professors*, una clase anónima contendrá tanto a los miembros de la clase *Student*, como a los de la clase *Professors*. En la Figura 3.10 se muestra este tipo de clase.

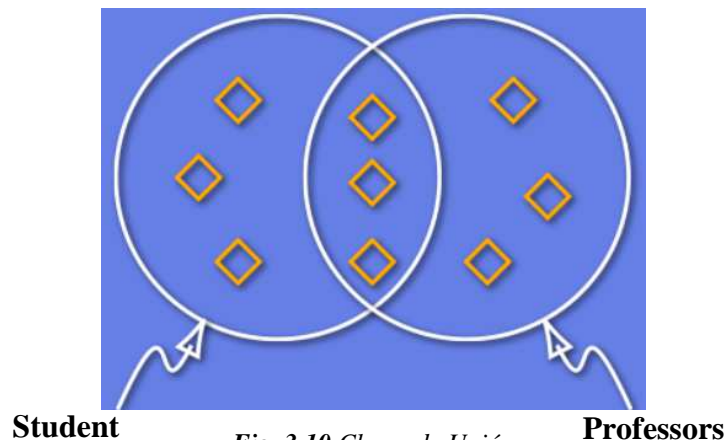


Fig. 3.10 Clases de Unión

3.7.4 Clases de Complemento

Las clases de complemento contienen a todos los individuos que no están contenidos en las clases de las cuales son complemento. Por ejemplo, existen dos clases, *Student* y *Worker*, la clase anónima que describe el complemento de la *Worker* serán los estudiantes que NO sean trabajadores y en la Figura 3.11 aparece la representación gráfica de este ejemplo.

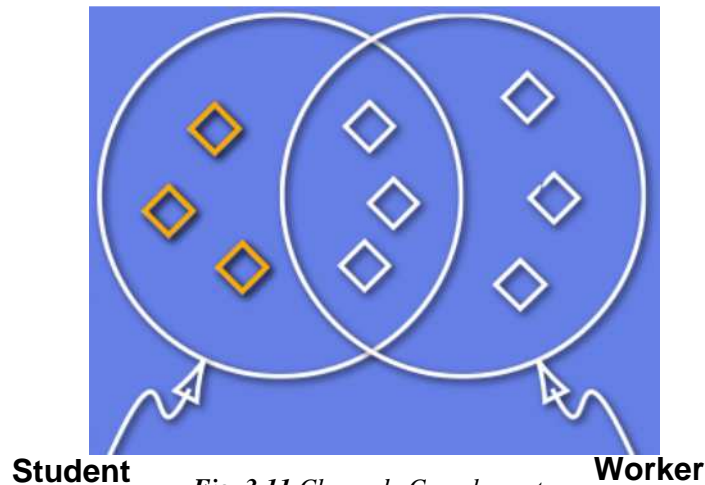


Fig. 3.11 Clases de Complemento

3.7.5 Restrictions

Una restricción describe una clase de individuos basados en el tipo y posiblemente en el número de relaciones que participan en ella. Al igual que las propiedades, las restricciones describen relaciones binarias, y las clases por ellas descritas hablan sobre los individuos miembros de las clases que en esas relaciones participan. Las restricciones son también un tipo de clases al igual que las clases con nombre.

En OWL es posible existen tres tipos principales de restricciones:

- a) Restricciones de Cuantificación
- b) Restricciones de Cardinalidad
- c) Restricciones tieneValor

A continuación se detallarán esas restricciones.

3.7.6 Restricciones Existenciales

Dentro de las restricciones de cuantificación, hay dos subtipos: existenciales y universales. Las restricciones existenciales describen clases de individuos que participen en *al menos una* relación junto a una propiedad específica para un individuo que es miembro de una clase específica. Por ejemplo, “la clase de individuos que tiene al menos (*some*) un colega”. Las restricciones existenciales son el tipo más común de restricciones en una ontología OWL, también son conocidas como *someValuesFrom restrictions*. En lógica descriptiva se utiliza el símbolo \exists (que se lee existe).

En Protégé se usa el símbolo **some** para denotar las restricciones existenciales, su sintaxis es: **hasColleague some Person**. En sintaxis DL el ejemplo anterior sería: \exists **hasColleague Person**, como aparece en la Figura 3.12.

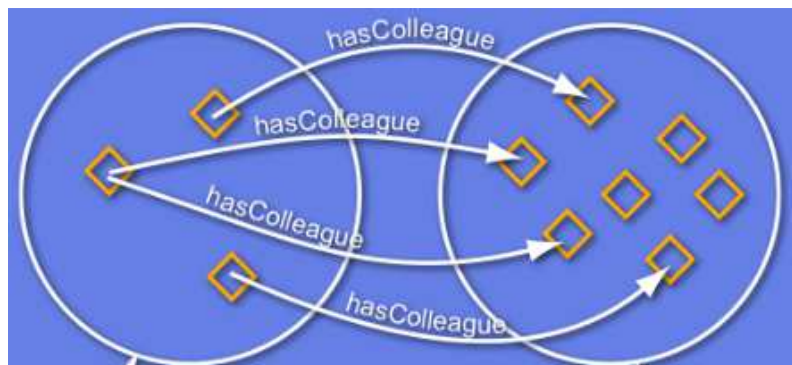


Fig. 3.12 Restricciones Existenciales

3.7.7 Universal Restrictions

Las restricciones existenciales describen clases de individuos que para una propiedad dada todos los individuos deben ser miembros de tales clases especificadas. Por ejemplo, “la clase de individuos que sólo tienen (*only*) un colega que pertenezca a la clase *Persona*”, las restricciones universales son también conocidas como *AllValuesFrom restrictions*. En lógica descriptiva se utiliza el símbolo \forall (que se lee para todo).

En Protégé se usa el símbolo **only** para denotar las restricciones universales, su sintaxis es **hasColleague only Person**. En sintaxis DL el ejemplo anterior sería: \forall **hasColleague Person**. Un ejemplo de esta restricción es mostrado en la Figura 3.13.

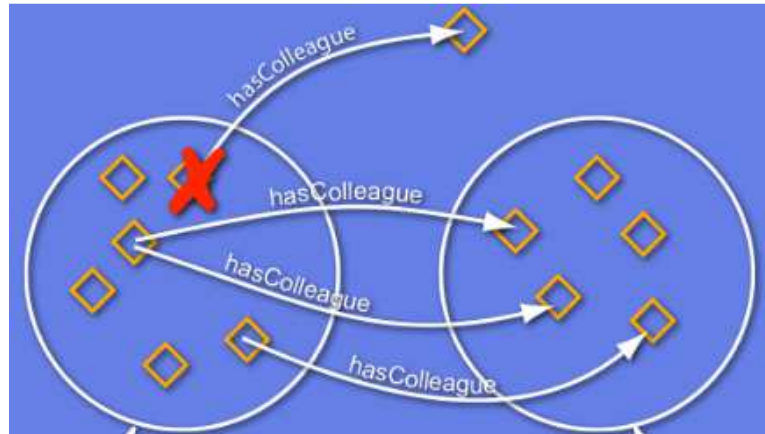


Fig. 3.13 Restricciones Universales

3.7.8 hasValue Restriction

Las restricciones *tieneValor* se representan por el símbolo \exists y describen los conjuntos de individuos (clases anónimas generadas por esta restricción) que están ligados a un valor V , dicho valor puede ser un individuo o un valor de datos. En otras palabras, describe una clase de individuos para los cuales la propiedad tiene al menos un valor *semánticamente* igual a V . Por ejemplo, la restricción **tienePaísDeOrigen** \exists **Italia** (donde Italia es un individuo de la clase *País*), describe el conjunto de individuos que su país de origen es Italia. En la Figura 3.14 se muestra otro ejemplo de una restricción **tieneValor**.

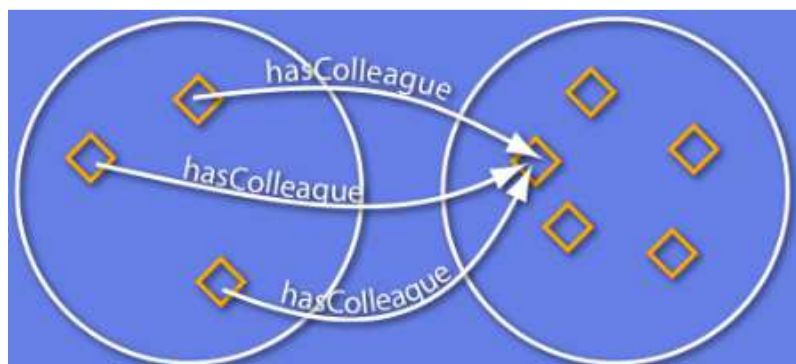


Fig. 3.14 Restricciones tieneValor

3.7.9 Enumeration Classes

OWL permite que las clases sean definidas por enlistar o enumerar de forma exhaustiva a todos los individuos que son miembros de dichas clases. Estas clases también son llamadas *Enumerated Classes* o Clases de enumeraciones y se denotan por medio de llaves ({}) y dentro de las llaves se enlistan todos sus miembros, por ejemplo {España, Alemania, Francia,...}, como se muestra en la Figura 3.15. Las clases de enumeración no forman parte del estándar OWL Lite.



Fig. 3.15 Clases de Enumeración

3.8 Propiedades OWL

Las propiedades OWL representan relaciones binarias entre dos objetos, clases o individuos, a estas relaciones también es posible denominarlas relaciones taxonómicas. Existen dos tipos principales de propiedades:

- a) Propiedades de Objeto (*Object properties*) y
- b) Propiedades de tipo de dato (*Datatype properties*)

3.8.1 Object Properties

Las *object properties* son relaciones entre dos individuos. Para determinar una *object property* es posible iniciar relacionando dos clases, por ejemplo:

Alumno **estaInscritoEn** Materia

Ahora, si se identifica a dos individuos pertenecientes (dos individuos nombrados cualesquiera) cada uno a una de las dos clases del ejemplo anterior, i.e., *Alumno* y *Materia*, se obtendrá la siguiente propiedad:

Javier_Lopez **estaInscritoEn** Base_de_Datos

Con lo cual, se obtiene el resultado correcto, relacionando dos individuos.

3.8.2 *Propiedades de tipo de dato*

Las *datatype properties* son relaciones que se establecen entre un individuo y valores de tipo de datos (*datatype values*), por ejemplo:

Matthew **tieneEdad** "25"^^integer

3.8.3 *Propiedades de anotación*

Existe un tercer tipo de propiedad: las *Annotation properties* (Propiedades de anotación), que son utilizadas para agregar información o metadatos, esto es, datos acerca de los datos, a las clases, individuos y *object/datatype properties*.

3.8.4 **Características de Propiedades**

OWL permite que el significado de las propiedades (únicamente para las *object properties*) sea enriquecido mediante el uso de “características” para tales propiedades. En las siguientes subsecciones se discutirán tales características.

3.8.4.1 *Funcional*

Si una propiedad es funcional, entonces para un individuo dado existe a lo más un individuo relacionado a él mediante la propiedad funcional, i.e., que para un individuo dado, la propiedad toma a lo más un solo valor. Por ejemplo, supónganse la propiedad funcional **tieneMadreBiológica** y dos individuos *Jean* y *Margaret*, dichos individuos están relacionados por medio de la propiedad de la siguiente manera: Jean

tieneMadreBiológica Margaret, lo que nos indica que el individuo Jean solamente una madre biológica.

3.8.4.2 *Funcional Inversa*

Si una propiedad es funcional inversa, esto significa que la propiedad *inversa es funcional*. Para un individuo dado existe a lo más un individuo relacionado a él mediante la propiedad funcional inversa. El ejemplo de la propiedad anterior se utilizará nuevamente, si existe una propiedad funcional Jean **tieneMadreBiológica** Margaret, entonces existe su propiedad inversa llamada **esMadreBiológicaDe**, relacionaría en este caso a los individuos *Margaret* y *Jean* de la siguiente manera: Margaret **tieneMadreBiológica** Jean, i.e., que el individuo Margaret es madre biológica de Jean.

3.8.4.3 *Transitiva*

Si una propiedad es transitiva y la propiedad relaciona dos individuos a y b, y un individuo b relacionado a un individuo c, entonces, es posible inferir que el individuo a está relacionado al individuo c mediante la propiedad transitiva. En la Figura 3.16 se muestra una representación gráfica de la propiedad transitiva utilizando la propiedad **hasAncestor**.

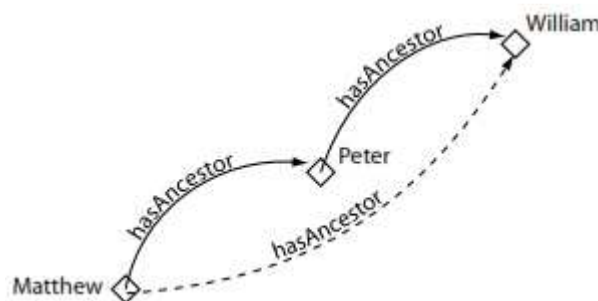


Fig. 3.16 Propiedad transitiva

Simétrica

Si una propiedad es simétrica y la propiedad relaciona a dos individuos a y b, entonces el individuo b está relacionado al individuo b mediante la propiedad simétrica. Por ejemplo, existe la propiedad simétrica **hasSibling** que relaciona a los individuos *Matthew* y *Gemma* de la siguiente manera Matthew **hasSibling** Gemma, entonces su propiedad simétrica sería

la siguiente: Gemma **hasSibling** Matthew y su representación gráfica es presentada en la Figura 3.17.



Fig. 3.17 Propiedad simétrica

Antisimétrica

Si una propiedad es antisimétrica, y la propiedad relaciona a dos individuos b y c , entonces el individuo c no puede estar relacionado al individuo b mediante la propiedad antisimétrica.

Reflexiva

Si una propiedad es reflexiva, entonces para un individuo dado a , existe una relación reflexiva que relaciona al individuo a consigo mismo. Por ejemplo, existe la propiedad simétrica **knows** que relaciona a los individuos *George* y *Simon* de la siguiente manera *George knows Simon*, y además cada uno de los individuos relaciona consigo mismo i.e., *George conoce a Simon* y al mismo tiempo *George se conoce a sí mismo*. Su representación gráfica se muestra en la Figura 3.18.

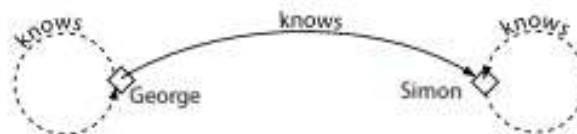


Fig. 3.18 Propiedad reflexiva

Irreflexiva

Si una propiedad es irreflexiva, entonces para un individuo dado existe a lo más un individuo relacionado a él mediante la propiedad funcional, i.e., que para un individuo dado, la propiedad toma a lo más un solo valor.

3.4 SWRL

Para el caso del estándar SWRL, igualmente ya se había definido una breve noción en el Capítulo 2, Sección 2.6. Sin embargo, en esta sección, se abundará en el estándar base de las reglas que serán utilizadas posteriormente en la implementación del sistema de ontologías.

Fue enviado al W3C en 2004, por varios autores, entre ellos Ian Horrocks y Peter F. Patel-Schneider. Los autores propusieron un lenguaje basado a su vez en una combinación de los sublenguajes OWL Lite, OWL DL y el lenguaje RuleML de Datalog.

Como ya se había esbozado en el capítulo 2, sección 2.6, SWRL es un lenguaje de expresión de reglas basado en OWL. Utiliza principalmente una sintaxis abstracta de OWL, llamada EBNF o notación de Backus-Naur Extendida por sus siglas en inglés.

3.5.1 Reglas

Una regla es una secuencia de axiomas complejos y axiomas simples. En sintaxis abstracta su notación es como sigue:

$$\mathbf{axioma ::= regla} \quad (3.1)$$

Cada está constituida por un antecedente o cuerpo (*body*) y un consecuente (*head*), el antecedente podrá estar constituido por varios átomos y el consecuente solo podrá ser uno. En la ecuación (3.2) se presenta este esquema:

$$\begin{aligned} \mathbf{regla} & ::= \text{' Implica (' [URlreference] \{ annotation \} antecedente } \\ & \mathbf{consecuente} \text{ ')' } \\ \mathbf{antecedente} & ::= \text{'Antecedente(' átomo \} ')' } \\ \mathbf{consecuente} & ::= \text{'Consecuente(' átomo \} ')' } \end{aligned} \quad (3.2)$$

3.5.2 Tipos de átomos

Los tipos de átomos soportados por SWRL son los siguientes:

- Átomos de clases
- Átomos de propiedades de individuos

- Átomos de propiedades de datos valuados
- Átomos de individuos diferentes
- Átomos de individuos iguales
- Átomos *built-in*
- Átomos de rango de datos

Y, a continuación, se describirán los tipos de átomos.

Átomos de clases

Un átomo de clase consiste de una clase nombrada (también llamada clase con nombre) o una expresión de clase y un argumento simple representando un individuo OWL, por ejemplo: **Persona(?p)**, **Mujer(Andrea)**.

En estos ejemplos, *Persona* y *Mujer* son dos clases OWL nombradas, ?p es la variable designada para representar un individuo OWL y Andrea es el nombre del individuo. Una regla simple podría ser la presentada en (3.3):

$$\mathbf{Mujer(?p) \rightarrow Persona(?p)} \quad (3.3)$$

Átomos de propiedades de individuos

Un átomo de propiedades de individuos consiste de una *object property* y dos argumentos representando individuos OWL, por ejemplo: **tieneHermano(?p, ?x)**, **tieneHermanos(Andrea, ?x)**.

Ahora, supóngase que se requiere escribir una regla que exprese que una persona con hermanos tiene una hermana mujer, entonces, es necesario extraer los conceptos persona, mujer y hermanos en OWL. Se pueden utilizar las clases utilizadas en (3.3), y los ejemplos anteriores. Entonces la regla tiene la forma (3.4):

$$\mathbf{Persona(?p) \wedge tieneHermanos(?p, ?s) \wedge Mujer(?s) \rightarrow tieneHermana(?p)} \quad (3.4)$$

Átomos de propiedades de datos valuados

Un átomo de propiedades de datos valuados consiste de una *data property* y dos argumentos, el primero representa a un individuo OWL y el segundo a un valor de datos, por ejemplo: **tieneEdad**(?x, ?edad), **tieneAltura**(Javier, ?a), **tieneNombre**(?x, "Javier").

Utilizando una propiedad de datos valuados booleana, una regla que afirme que todas las personas que son dueñas de un automóvil, son conductores es (3.5):

$$\text{Persona}(?p) \wedge \text{tieneAutomovil}(?p, \text{true}) \rightarrow \text{Conductor}(?p) \quad (3.5)$$

Átomos de individuos diferentes

Un átomo de individuos diferentes consiste del símbolo **differentFrom** y dos argumentos que representan individuos OWL, por ejemplo: **differentFrom**(?x, ?y), **differentFrom**(Javier, Carlos).

Átomos de individuos iguales

Un átomo de individuos iguales símbolo **sameAs** y dos argumentos que representan individuos OWL, por ejemplo: **sameAs**(?x, ?y), **sameAs**(Javier, Javi).

Átomos *built-in*

Una de las características más poderosas de SWRL es el soporte de *built-ins* definidas por el usuario. Un *built-in* es un predicado que toma uno o más argumentos y evalúa si los argumentos satisfacen el predicado. Por ejemplo, un *built-in equal*, puede ser definido con dos argumentos, y determinar si ambos argumentos son iguales. Existen diversos *built-ins* definidos para operaciones matemáticas y de cadena, como suma aritmética, sustracción, multiplicación, división. En (3.6) se muestra una regla que utiliza un *built-in* para determinar que si una persona es mayor de 17 años, entonces es adulto

$$\text{Persona}(?p) \wedge \text{tieneEdad}(?p, ?edad) \wedge \text{swrlb:greaterThan}(?edad, 17) \rightarrow \text{Adulto}(?p) \quad (3.6)$$

Átomos de rango de datos

Un átomo de rango de datos consiste de un nombre de tipo de datos y un argumento representando un valor de datos, por ejemplo: `xsd:int(?x), [3, 4, 5](?y)`

3.5.3 Sintaxis legible por humanos

Mientras la sintaxis abstracta es consistente con OWL, no es particularmente fácil de leer. Se utiliza una sintaxis legible por humanos para leer de manera más fácil las reglas SWRL. En esta sintaxis una regla tiene la forma:

$$\text{antecedente} \Rightarrow \text{consecuente} \quad (3.7)$$

A continuación, en el siguiente capítulo se abordará el análisis del problema y su posterior implementación.

Capítulo 4

Desarrollo de la ontología propuesta para la deducción de patrones de estilo de vida

En este capítulo se presenta el análisis y diseño de la propuesta del sistema de ontologías para la deducción de patrones de estilo de vida, la creación de los axiomas y relaciones semánticas necesarias y la aplicación de las reglas SWRL.

4.1 Análisis

4.1.1 Planteamiento del problema

El auge de enfermedades derivadas de la adquisición de ciertos hábitos dañinos para la salud, como el consumo de tabaco y de la falta de actividad física de las personas, dando como resultado el incremento de las Enfermedades No transmisibles. Las ENT aquejan a gran parte de la población mundial y reportan consecuencias devastadoras en toda la población, i.e., la muerte de más de 38 millones de personas anualmente de todos los grupos étnicos en todo el mundo.

4.1.2 Preguntas de investigación

El sistema de ontologías deberá ser capaz de responder preguntas de competencia tales como:

- ¿Cuáles son las características de una persona sedentaria?
- ¿Cómo es la nutrición de una persona que padece de una ENT?
- ¿Cuáles son las características de una persona con un estilo de vida saludable?
- ¿Cuáles son los principales factores de riesgo para sufrir de diabetes?
- ¿Cómo se define un patrón de estilo de vida nivel medio?
- ¿Cuáles son las características de una persona con nutrición alta en grasas?
- ¿Está relacionado el sufrir de diabetes con la nutrición alta en grasas?

Dichas preguntas se han desarrollado con el objetivo de lograr un cambio en el estilo de vida de una persona.

4.1.3 Representación de la información

La representación de la información es a través de ontologías, representando los dominios de aplicación: Personas, Regiones geográficas, Nutrición, Síntomas de enfermedades, ENT y Actividades físicas.

4.2 Diseño del sistema de ontologías (Nivel Macro)

Una vez definida la metodología analizada en el Capítulo 2, y con base en el problema planteado en la Sección 4.1 y las preguntas de competencia implementadas en la Sección 4.1.2, se procede a iniciar con la etapa de Diseño del Sistema de Ontologías.

Esta etapa consistirá en el diseño general del sistema de ontologías propuesto, y sus relaciones semánticas involucradas. Posteriormente, en la etapa de integración y evaluación, se realiza la aplicación de reglas SWRL para deducir patrones de estilo de vida.

El software utilizado para crear el sistema fue Protégé ver4.3, e inicialmente, las ontologías correspondientes a los dominios representados en el sistema se crearon de forma individual, teniéndose las siguientes ontologías:

- *Person* - Personas,
- *OntoMex* - Regiones geográficas,
- *Nutrition* - Nutrición,
- *Symptom* - Síntomas de enfermedades,
- *NCD* - ENT y
- *Physical Activity* - Actividades físicas

El diseño conceptual del sistema se muestra en la Figura 4.1. Son observables las relaciones semánticas que fueron establecidas entre clases.

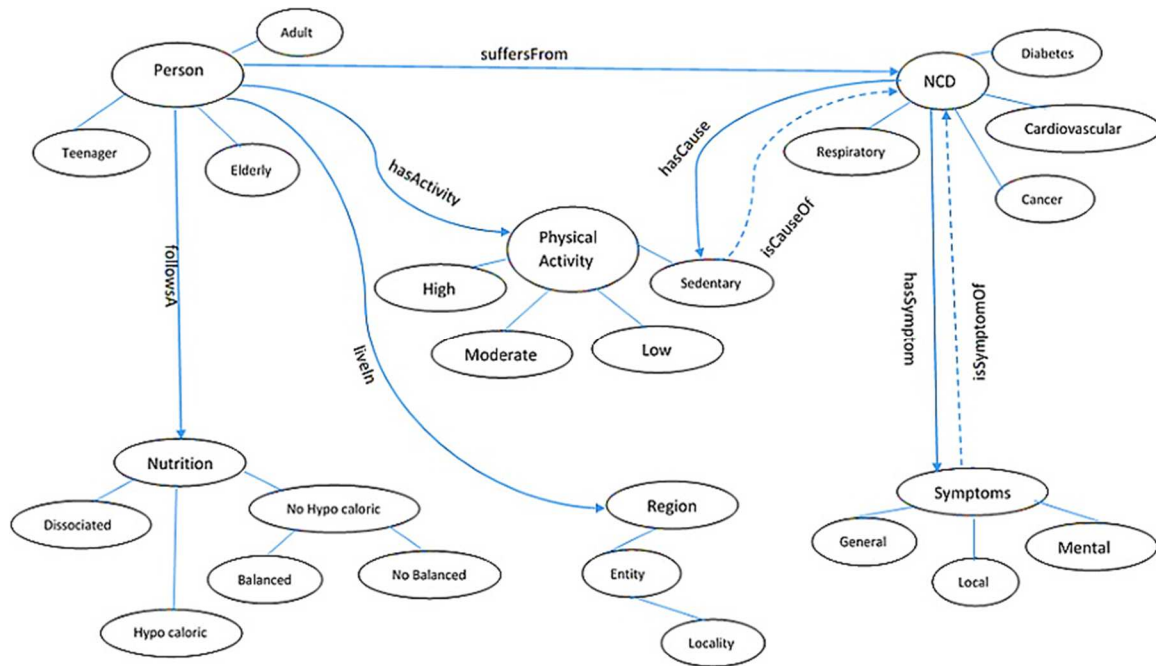


Fig. 4.1 Relaciones semánticas del sistema de ontologías

4.3 Etapa de Diseño de las ontologías (Nivel Invidual)

Una vez terminada la fase de Diseño del Sistema de Ontologías, se prosigue con la fase de diseño y construcción de las ontologías.

Posteriormente, se creó una nueva ontología en la cual se volcaron las ontologías citadas anteriormente, por medio de la importación de las seis ontologías que compondrán el sistema por medio de la herramienta *Import Ontologies* de la herramienta de software Protégé 4.3, conforme se muestra en la Figura 4.2.

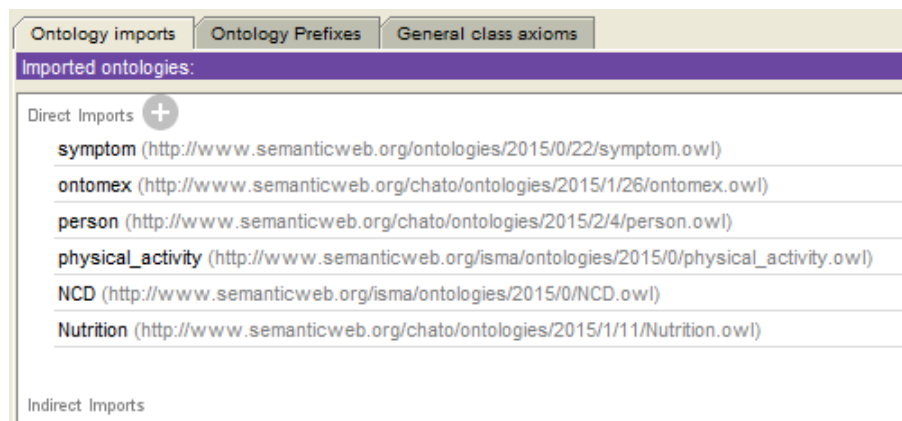


Fig. 4.2 Herramienta Import Ontologies en Protégé 4.3.

Una vez que han sido realizadas las respectivas acciones de importación de cada una de las seis distintas ontologías individuales, se tiene la siguiente estructura de clases, mostrada en la Figura 4.3.

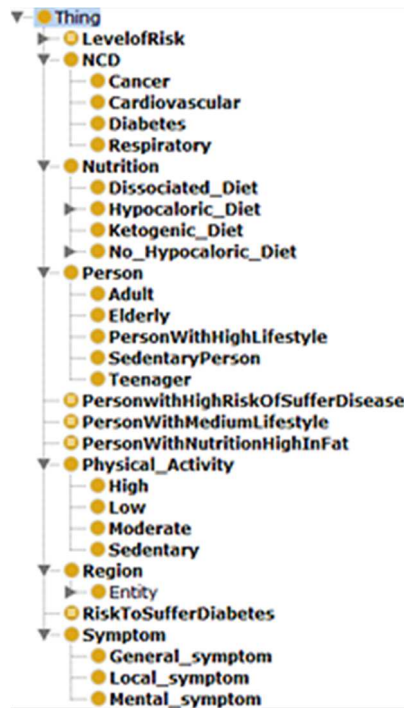


Fig. 4.3 Estructura de clases del sistema de ontologías.

En la Figura 4.4 se muestra la representación gráfica de las ontologías que conforman el sistema.

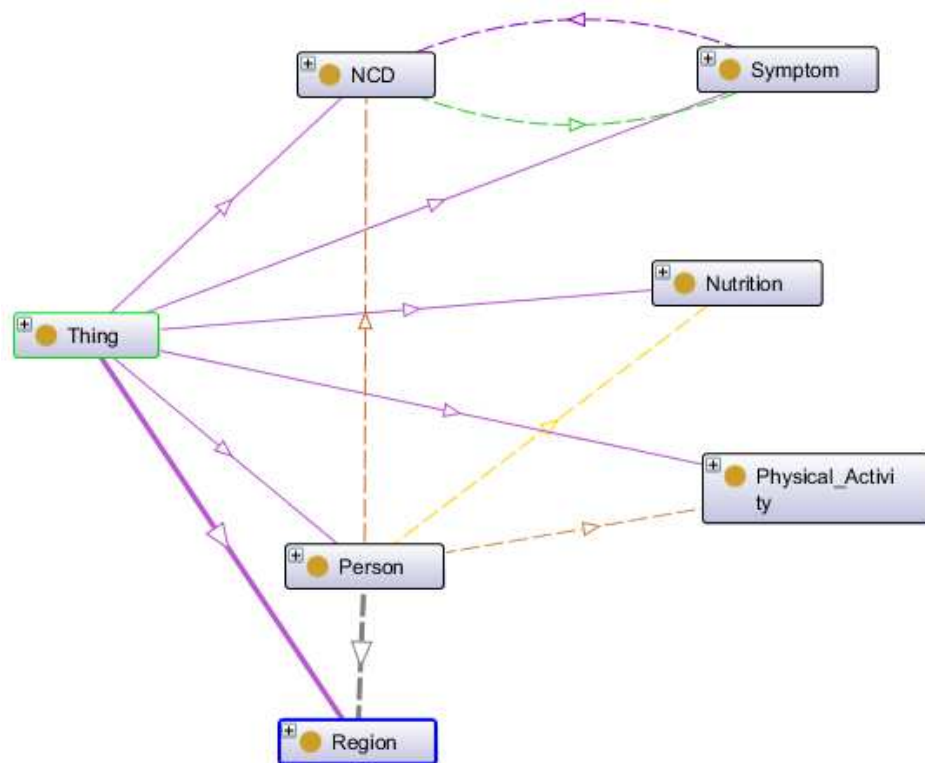


Fig. 4.4 Grafo de ontologías del sistema

4.4.1 Axiomatización (Subetapa de Integración de las ontologías)

Después de concluir con las dos fases anteriores de la metodología de desarrollo de del sistema de ontologías, únicamente resta la última fase, la cual es Integración y Evaluación del sistema de ontologías. La Etapa de Integración y Evaluación se ha dividido en dos sub-fases para una mejor aplicación de la metodología como ya se ha explicado en el Capítulo 2, sección 2.8.2.

Se definirán los siguientes axiomas: Propiedades de tipo de datos, Propiedades de objetos, clases inferidas, restricciones y particiones de valor. En la Figura 4.5, se observan las reglas SWRL que forman parte del sistema de ontologías.

Rules:
Entity(?en), Nutrition(?n), Adult(?p), Sedentary(?pa), integer[>= 40 , <= 50](?age), followsA(?p, ?n), hasActivity(?p, ?pa), livesIn(?p, ?en), hasAge(?p, ?age) -> RiskToSufferDiabetes(?p)
Teenager(?x), High(?y), hasActivity(?x, ?y) -> PersonWithHighLifestyle(?x)
Non-Balanced_diet(?n), Elderly(?a), Diabetes(?e), followsA(?a, ?n), suffersFrom(?a, ?e) -> PersonWithNutritionHighInFat(?a)
High_Level(?hl), Person(?p), Sedentary(?pa), hasActivity(?p, ?pa), hasLevelOfRisk(?s, ?lr) -> PersonwithHighRiskOfSufferDisease(?p)
Person(?p), Diabetes(?n), High(?h), hasActivity(?p, ?h), suffersFrom(?p, ?n) -> PersonWithMediumLifestyle(?p)
Adult(?x), Sedentary(?y), hasActivity(?x, ?y) -> SedentaryPerson(?x)

Fig. 4.5 Reglas SWRL del sistema

Propiedades de Tipo de datos.

Creación de reglas SWRL utilizando únicamente *datatype properties*. Inicialmente, estas reglas conformaron la base de pruebas inicial del sistema. Se definió la siguiente regla SWRL (4.1), que determina la pertenencia a la clase *Teenager* de un individuo, y se expresa de la siguiente manera en lenguaje natural: “Si una persona tiene una edad entre un rango de 12 a 17 años, entonces es un adolescente”.

$$Person(?p), integer[>=12, <=17](?age), has Age(?p, ?age) -> Teenager(?p) \quad (4.1)$$

Como se observa, la regla únicamente determinaba individuos de una sola clase y una sola ontología, al ejecutar el razonador, se clasificaba correctamente a los individuos que cumpliesen la condición de la regla (4.1), utilizando únicamente una *data property*.

Igualmente se definió otra *data property* y se diseñó una nueva regla, la cual determinaría personas sedentarias, con base en el tipo de actividad realizada por el individuo.

La regla (4.2) expresa que: “Si la actividad física realizada por una persona solo se reduce a estar sentada, es decir, no realiza ninguna actividad física, entonces es una persona sedentaria”.

$$Person(?p), activityType(?p, "to sit") -> SedentaryPerson(?p) \quad (4.2)$$

En este punto, solo eran utilizadas *data properties*, y aunque semánticamente eran correctas y el razonador no arrojaba ninguna inconsistencia, al no hacer uso de todas las demás clases no se obtenía ningún avance significativo.

Es vital añadir que, al realizar la clasificación bajo las anteriores condiciones, las clases que contienen a los individuos que cumplen las condiciones citadas, deben ser subclases de la clase *Person*, ya que si esto no se realiza, el sistema podría tener una inconsistencia.

Propiedades de Objetos

Creación de *object properties* que establecen relaciones entre instancias de clases y de clases a clases, para agregar mayor expresividad a las reglas SWRL, además de que sea posible que las clases que contienen a los miembros que cumplan las condiciones definidas para las reglas SWRL no deban de estar sujetas a ser subclases de otra clase en particular.

Las *object properties* que han sido incorporadas al sistema se muestran en la Figura 4.6.

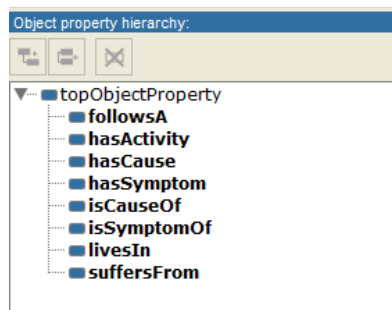


Fig. 4.6 *Object properties incorporadas al sistema*

Y por medio de la *object property* **hasActivity** se crea una regla (4.3), que declara lo siguiente: “Si una persona tiene una actividad física nula, entonces es una persona sedentaria”, regla que es similar a la regla (4.2), sin embargo, con la funcionalidad de las *object properties* que muestra la Figura 4.6 incorporadas al sistema.

$$\begin{aligned}
 & Person(?x), Physical_Activity(?act), hasActivity(?x, ?act) \\
 & \quad \rightarrow SedentaryPerson(?x)
 \end{aligned}
 \tag{4.3}$$

Como consecuencia del uso de las *object properties*, se incorporaron *Inferred Classes* o clases inferidas, esto es, nuevas clases las cuales inicialmente estarán sin individuos. Cuando se ejecute el razonador con diversos axiomas incorporados se realiza la clasificación de la ontología, las clases inferidas contengan a los miembros que cumplan con las condiciones axiomáticas definidas anteriormente. De esta forma será posible responder a las preguntas de investigación, que representan uno de los objetivos de esta tesis.

4.4.2 Subetapa de Evaluación

Esta subetapa está conformada a su vez por las siguientes sub-fases:

- Diseño e implementación de Clases Inferidas
- Diseño e implementación de Axiomas o Restricciones
- Diseño e implementación de Condiciones Necesarias y Suficientes
- Deducción de miembros de clases
- Descripción de miembros de clases
- Deducción de Patrones de Estilos de Vida

Clases inferidas

Creación de clases sin elementos, para que posteriormente, al realizar la inferencia con el razonador, éste clasifique miembros dentro de estas clases que cumplan con las o la regla SWRL, los axiomas de condiciones necesarias y suficientes o en su defecto restricciones existenciales.

Iniciamos con la clase *PersonWithNutritionHighInFat*, la cual contendrá a los individuos con una nutrición alta en grasas. Esta clase utiliza las *object properties* **followsA** y **suffersFrom**, las cuales establecen relaciones entre las clases *Person* y *Nutrition* y, *Person* y *NCD*, respectivamente. Estas relaciones relacionan a individuos que llevan un tipo de nutrición específico e igualmente personas que sufren de un tipo de *NCD* específica.

Como detalle, la clase *PersonWithNutritionHighInFat* se creó como subclase de **Thing**, en vez de *Person*, para no sujetar a la nueva clase al mero uso de *data properties* únicamente y elevar las capacidades semánticas y de razonamiento de la ontología. En la Figura 4.7 se muestra la citada clase seleccionada en color azul.

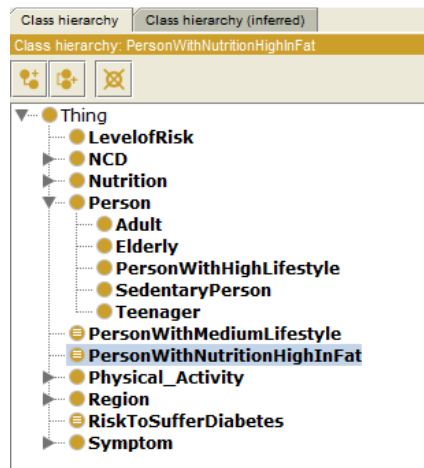


Fig. 4.7 Clase *PersonWithNutritionHighInFat*.

Axiomas existenciales

Inclusión de axiomas o restricciones existenciales y axiomas de condiciones necesarias y suficientes al sistema de ontologías. Las restricciones existenciales describen clases de individuos que cumplen con al menos una relación en la propiedad, en este caso en el axioma.

Las condiciones necesarias y suficientes, además de lo anterior, agregan una restricción para delimitar los individuos que cumplen exactamente con las dos condiciones, agregando mayor expresividad a la ontología al especificar miembros de clases de una manera más restrictiva y consistente.

Continuando con el caso base de la clase *PersonWithNutritionHighInFat*, se ha creado el axioma **followsA some Non-Balanced_diet**. Cuando es ejecutado el razonador, éste arroja una inconsistencia, por lo cual es necesario convertir a la clase *PersonWithNutritionHighInFat* en una clase definida.

Condiciones Necesarias y suficientes

Clases primitivas

Por defecto, todas las clases creadas en Protégé se crean como clases primitivas, esto es, que usan únicamente condiciones necesarias para describirlas, como el axioma creado anteriormente. Las condiciones necesarias pueden leerse de la siguiente manera: “Si algo es miembro de esta clase, entonces es *necesario* cumplir con éstas otras condiciones”. Si únicamente son utilizadas condiciones necesarias, no es posible decir: “Si algo cumple con tales condiciones, entonces debe ser miembro de ésta clase”.

Clases definidas

Las clases definidas utilizan, además de las condiciones necesarias, un tipo de condiciones llamado condiciones *necesarias y suficientes*, en las cuales, aparte de cumplir con las condiciones necesarias, se incluyen las condiciones suficientes que delimitan a los individuos que pertenecen a una clase definida. Con las condiciones necesarias y suficientes es posible decir que si un individuo es miembro de una clase A, debe satisfacer las condiciones previamente definidas y que si cualquier individuo satisface esas condiciones, entonces debe ser miembro de la clase A. En la Figura 4.8 se muestra una comparación entre las condiciones necesarias y las condiciones necesarias y suficientes.

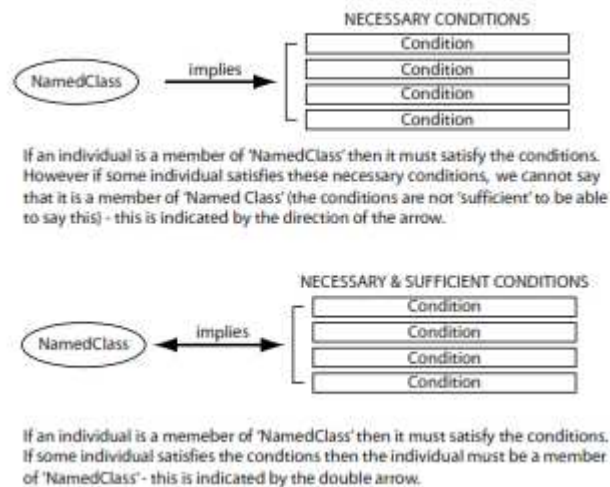


Fig. 4.8 Comparación entre Necessary conditions y Necessary & Sufficient Conditions

En Protégé las clases primitivas son llamadas **Superclasses** y las clases definidas son llamadas **Equivalent Classes**.

Para lograr que la clase *PersonWithNutritionHighInFat* se convirtiera en una clase definida y evitar la generación de inconsistencias debidas a los solapamientos en la ontología al realizar la clasificación, se utiliza la opción “**Convert selected rows to defined class**” que se activa al seleccionar el axioma con el botón derecho del mouse. Una vez teniéndose la clase como definida se ejecuta el razonador y se tiene un resultado similar al mostrado en la Figura 4.9.

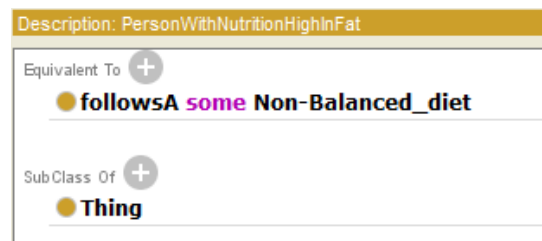


Fig. 4.9 Condiciones necesarias y suficientes del sistema

Deduciendo miembros de clases

Por medio de las reglas SWRL implementadas en el sistema de ontologías, de la aplicación de restricciones, de la implementación y definición de condiciones necesarias y suficientes o en su defecto de axiomas existenciales, deducir los individuos pertenecientes a determinadas clases, como resultado de clasificar la ontología y no contener el sistema ninguna inconsistencia.

Ahora, el siguiente paso en la metodología es que, una vez se han definido los axiomas adecuados para la clase de prueba, y con base en los axiomas mostrados en las Figuras 4.5, 4.6 y 4.8 se crea la regla (4.4), que se expresa de la siguiente forma en lenguaje natural: “Si un Adulto Mayor, lleva una nutrición no balanceada y sufre de diabetes, entonces es una persona con una nutrición alta en grasas”, y cuya expresión en SWRL es como sigue:

$$\begin{aligned}
 & Non_Balanced(?n), Elderly_Person(?a), Diabetes(?e), followsA(?a, ?n), \\
 & suffersFrom(?a, ?e) \rightarrow PersonWithNutritionHighInFat(?a)
 \end{aligned}
 \tag{4.4}$$

La regla (4.4) usa la *object property* **suffersFrom**, que relaciona a individuos de las clases *Person* y *NCD*, así como la propiedad **followsA**, que relaciona a individuos de las clases *Person* y *Nutrition*.

Entonces, la clase *PersonWithNutritionHighInFat* contendrá a los individuos que sufren de diabetes, son personas de la tercera edad y llevan una nutrición no balanceada. Esta es la primera aproximación a deducir Patrones de Estilo de Vida por medio de las reglas SWRL creadas. En la Figura 4.10 se muestra el resultado de ejecutar el razonador sobre la ontología una vez que la regla ha sido creada.

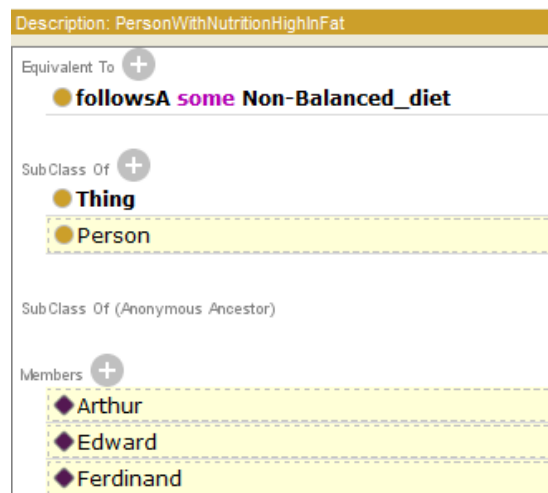


Fig. 4.10 Deduciendo miembros de clases por medio de reglas SWRL.

Descripción de miembros

Descripción, por medio de las aserciones de Propiedades de Objetos (*Object Property Assertions*), basadas a su vez en las Propiedades de Objeto de las clases, descritas en el paso 2, de las propiedades inherentes a los individuos pertenecientes a una clase. Por ejemplo, en el caso de un individuo de la clase *Persona*, describir que tipo de actividad realiza (**hasActivity**), o que tipo de NCD padece (**suffersFrom**), toda vez que previamente haya sido clasificado el sistema de ontologías y que éste no presente inconsistencias.

En este apartado el objetivo será describir los individuos que son miembros de la clase inicial de prueba. Se abre la ficha *Individuals*, y después en el apartado *Property assertions* para el individuo seleccionado y se obtendrá el resultado mostrado en la Figura 4.11.

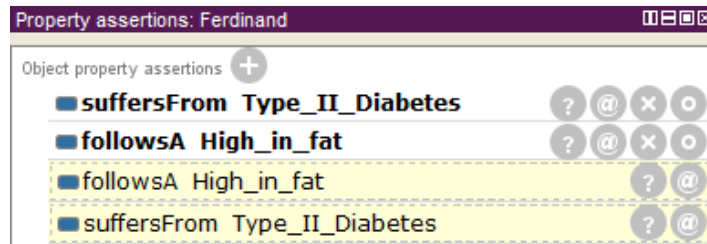


Fig. 4.11 Property Assertions para el individuo

Deducción de patrones de estilos de vida

Una vez teniendo las reglas SWRL definidas e implementadas en el sistema de ontologías, determinar cuáles son los estilos de vida adecuados basados en las clases creadas y de las cuales se han inferido sus miembros.

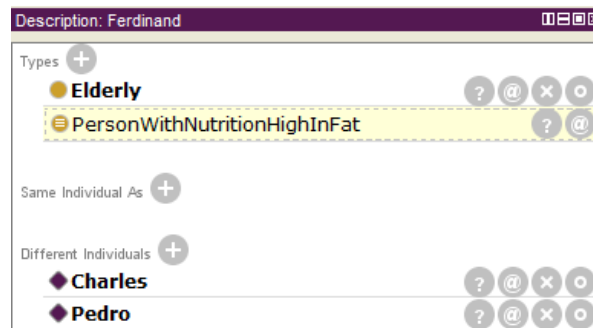


Fig. 4.12 Descripción de las clases a las que pertenece el individuo

Posteriormente, se crea una nueva clase para, en este caso, determinar personas con estilo de vida alto, con un cierto detalle: esas personas serán adolescentes y creamos la regla (4.5) para obtener a esos individuos. La regla se expresa así: “Si una persona adolescente tiene una actividad física alta, entonces su estilo de vida es también alto”.

$$Teenager(?x), High(?y), hasActivity(?x, ?y) \rightarrow PersonWithHighLifestyle(?x) \quad (4.5)$$

Determinar personas con estilos de vida alto y de cualquier rango de edad, es necesario entonces cambiar en la regla que se trata de la clase *Person* para englobar a cualquier individuo.

Los individuos con estilo de vida alto son clasificados por medio de la Regla (4.6) y que expresa lo siguiente: “Si una persona sufre de diabetes, pero tiene una actividad física alta, entonces su estilo de vida es medio”.

$$\begin{aligned}
 & Person(?p), Diabetes(?n), High(?h), hasActivity(?p, ?h), \\
 & \quad suffersFrom(?p, ?n) \rightarrow \\
 & \quad PersonWithMediumLifestyle(?p)
 \end{aligned}
 \tag{4.6}$$

En la Figura 4.13 se muestran los axiomas que describen a la clase *PersonWithMediumLifestyle* y los miembros que pertenecen a ella al ejecutar el razonador y clasificar la ontología. Es de notar que dicha clase es una clase con condiciones necesarias y suficientes

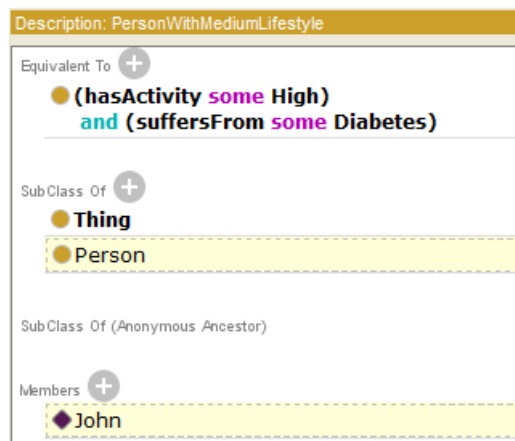


Fig. 4.13 Deducción de estilo de vida medio a través de reglas SWRL.

Posteriormente se crea la regla (4.7), que en lenguaje natural declara lo siguiente: “Si una persona que tiene un nivel alto en su factor de riesgo y tiene una actividad física sedentaria, entonces es una persona con un riesgo alto de sufrir una enfermedad”.

$$\begin{aligned}
& \text{High_Level}(?hl), \text{Person}(?p), \text{Sedentary}(?pa), \text{hasActivity}(?p, ?pa), \\
& \text{hasLevelOfRisk}(?s, ?lr) \\
& \rightarrow \text{PersonwithHighRiskOfSufferDisease}(?p)
\end{aligned} \tag{4.7}$$

Producto de la aplicación de la regla 4.7 se obtiene el resultado mostrado en la Figura 4.14.

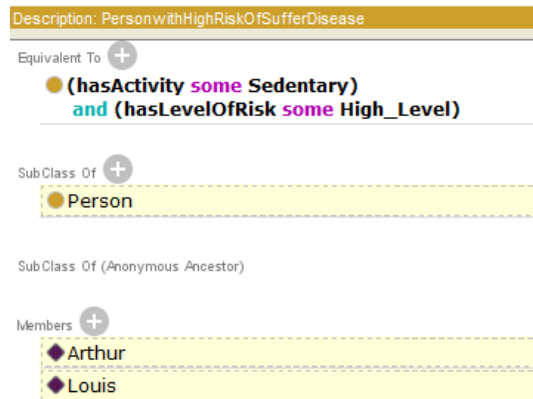


Fig. 4.14 Deducción del patrón PersonWithHighRiskOfSufferDisease

Ahora es necesaria una regla que deduzca un patrón para determinar el riesgo de padecer diabetes de acuerdo a los siguientes antecedentes: entidad federativa donde vive una persona (algún estado del norte del país), rango de edad entre 40 y 50 años, actividad física sedentaria, a elegir una de las dos actividades de prueba con que cuenta el sistema: sitting o T.V. y nutrición alta en grasas.

Para lograr tal objetivo, se crea la regla (4.8), que en lenguaje natural expresa: “Si una persona adulta entre 40 y 50 años de edad con una nutrición determinada, vive en cierto Estado, y es sedentaria, entonces es una persona con riesgo de padecer diabetes”.

$$\begin{aligned}
& \text{Entity}(?en), \text{Adult}(?p), \\
& \text{hasAge}(?p, ?age), \text{integer}[>= 40, <= 50](?age), \\
& \text{Nutrition}(?n), \text{Sedentary}(?pa), \text{followsA}(?p, ?n), \text{hasActivity}(?p, ?pa), \\
& \text{livesIn}(?p, ?en) \\
& \rightarrow \text{RiskToSufferDiabetes}(?p)
\end{aligned} \tag{4.8}$$

Una vez creada la regla (4.8), se ejecuta el razonador y se obtienen los resultados mostrados en la Figura 4.15. Es de notar que el razonador infiere que esta nueva clase es subclase de la clase *PersonWithNutritionHighInFat*, como puede observarse en la Figura 4.16 y tiene como miembro al individuo *Arthur*, el cual es miembro también de la clase *PersonWithNutritionHighInFat* y que tiene 40 años de edad.

Al analizar las características del individuo antes mencionado observamos en la Figura 4.17 que su actividad física es permanecer sentado, vive en Nuevo León y lleva una nutrición alta en grasas. También se observa que el individuo resulta ser miembro de otras clases como *SedentaryPerson*, y *RiskToSufferDisease*.

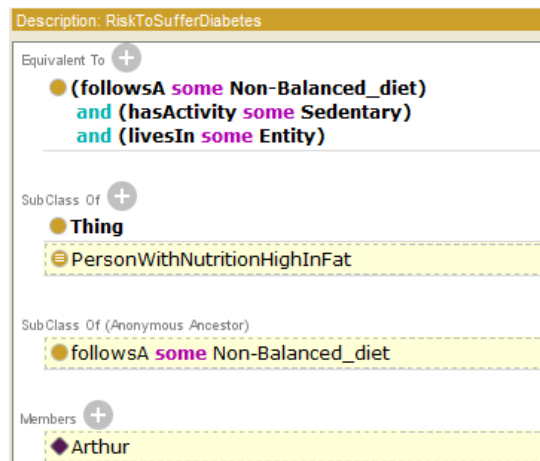


Fig. 4.15 Deducción del patrón *RiskToSufferDiabetes*.

El razonamiento en OWL (por medio de OWL DL) está basado en el **OWA**³² (Suposición de Mundo Abierto, por sus siglas en inglés), también llamado **OWR**³³, al contrario que en muchos aspectos del razonamiento humano que trabaja con **CWA**³⁴. La suposición de mundo abierto quiere decir que no es posible asumir que una declaración es falsa hasta que esté explícitamente declarado que tal declaración es falsa. Esto es, si algo no está dicho que sea verdadero, no es posible asumir que sea falso.

³² Open World Assumption

³³ Open World Reasoning

³⁴ Closed World Assumption

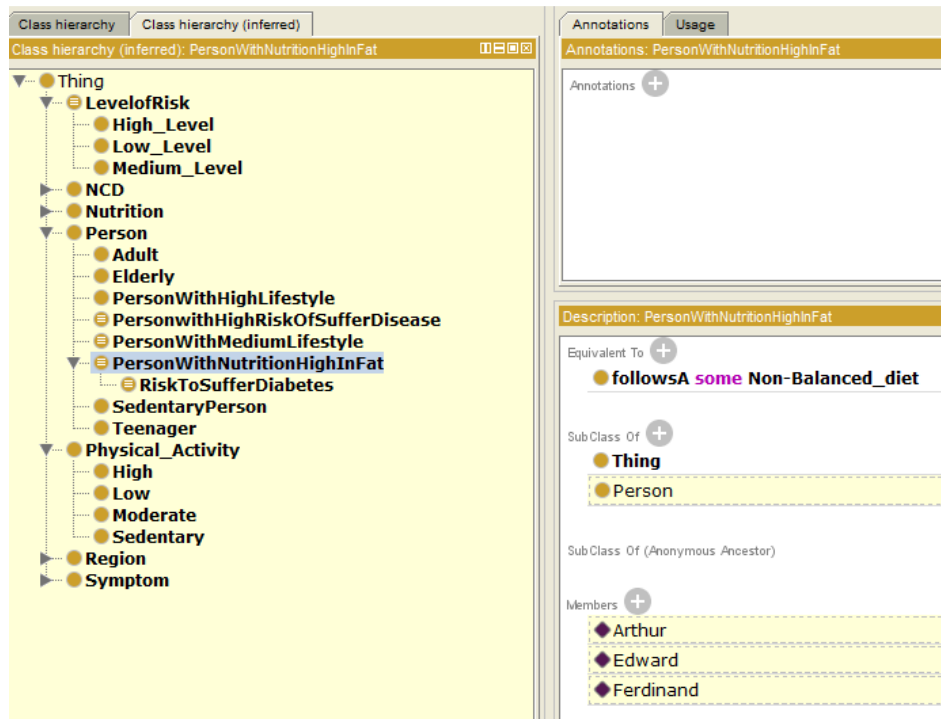


Fig. 4.16 Ejecución del razonador

En el caso del sistema de ontologías, una vez que se tienen agregados los axiomas de condiciones necesarias y suficientes, i.e., una vez que se ha expresado que la declaración de los patrones de estilo de vida *RiskToSufferDiabetes*, es posible clasificar a dicha clase como subclase de *PersonWithNutritionHighInFat*, como lo muestra la clasificación del sistema en la Figura 4.16.

Seguidamente, es posible hacer una descripción de los individuos, y se puede observar que, de acuerdo a las restricciones definidas para cada individuo (*object* y *datatypes properties*), los axiomas y el razonamiento basado en DL, tal individuo puede ser miembro de distintas clases y, tales clases ayudan a inferir correctamente que debido a una actividad sedentaria, nutrición alta en grasas, la edad y la región donde se vive, el riesgo de sufrir diabetes o de sufrir alguna enfermedad es alto, como se observa en la Figura 4.17.

The screenshot displays a software interface for describing an individual named Arthur. It is divided into two main sections:

- Description: Arthur (Left Panel):**
 - Types:** A list of classes that Arthur belongs to: **Adult** (selected), **PersonwithHighRiskOfSufferDisease**, **RiskToSufferDiabetes**, and **SedentaryPerson**. Each class has control icons (question mark, at-sign, X, O).
 - Same Individual As:** A section with a plus sign, currently empty.
 - Different Individuals:** A section with a plus sign containing two individuals: **Edward** and **Louis**, each with control icons.
- Property assertions: Arthur (Right Panel):**
 - Object property assertions:** A list of assertions with control icons:
 - hasActivity Sitting** (selected)
 - followsA High_in_fat** (selected)
 - livesIn Nuevo_Leon** (selected)
 - hasActivity Sitting** (dashed border)
 - followsA High_in_fat** (dashed border)
 - livesIn Nuevo_Leon** (dashed border)
 - Data property assertions:** A list of assertions with control icons:
 - hasAge 40** (selected)
 - hasAge 40** (dashed border)

Fig. 4.17 Descripción del individuo Arthur.

En la Figura 4.19, se muestra el grafo de los patrones de estilo de vida encontrados.

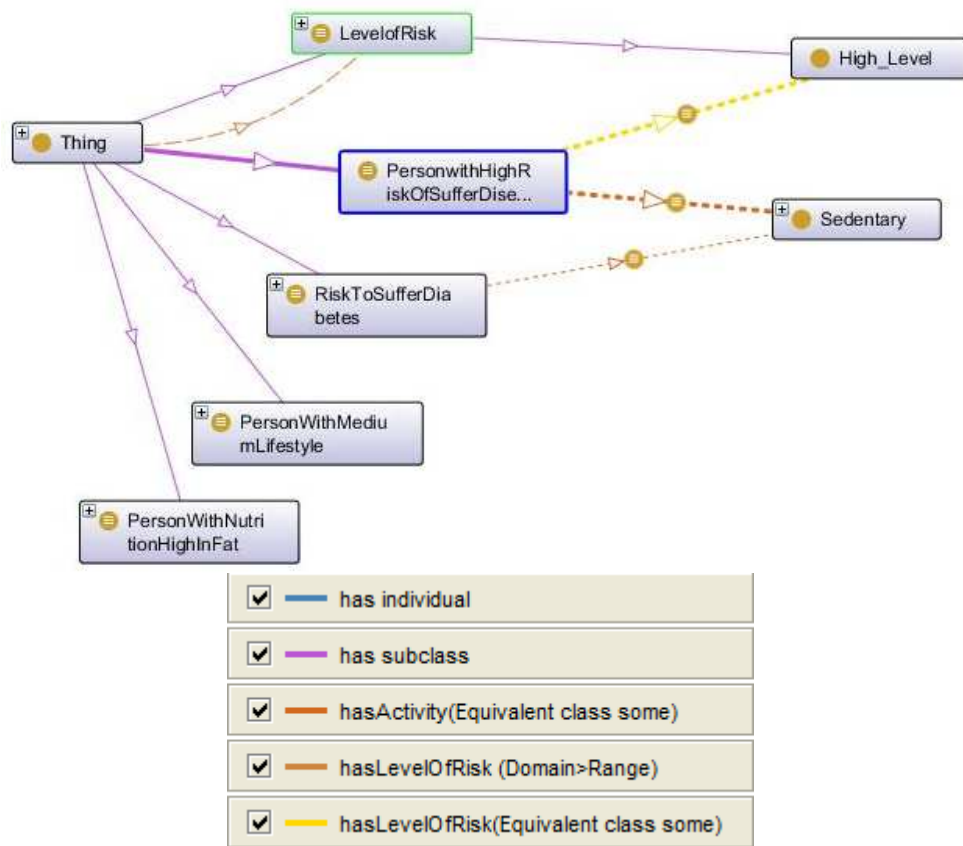


Fig. 4.19 Grafo de los patrones de estilo de vida encontrados

En la Figura 4.20 se tiene el *Inferred Model* del sistema generado por el plugin ToolViz de Protégé, el cual es el diagrama de la jerarquía de clases del sistema una vez se ejecuta el razonador.

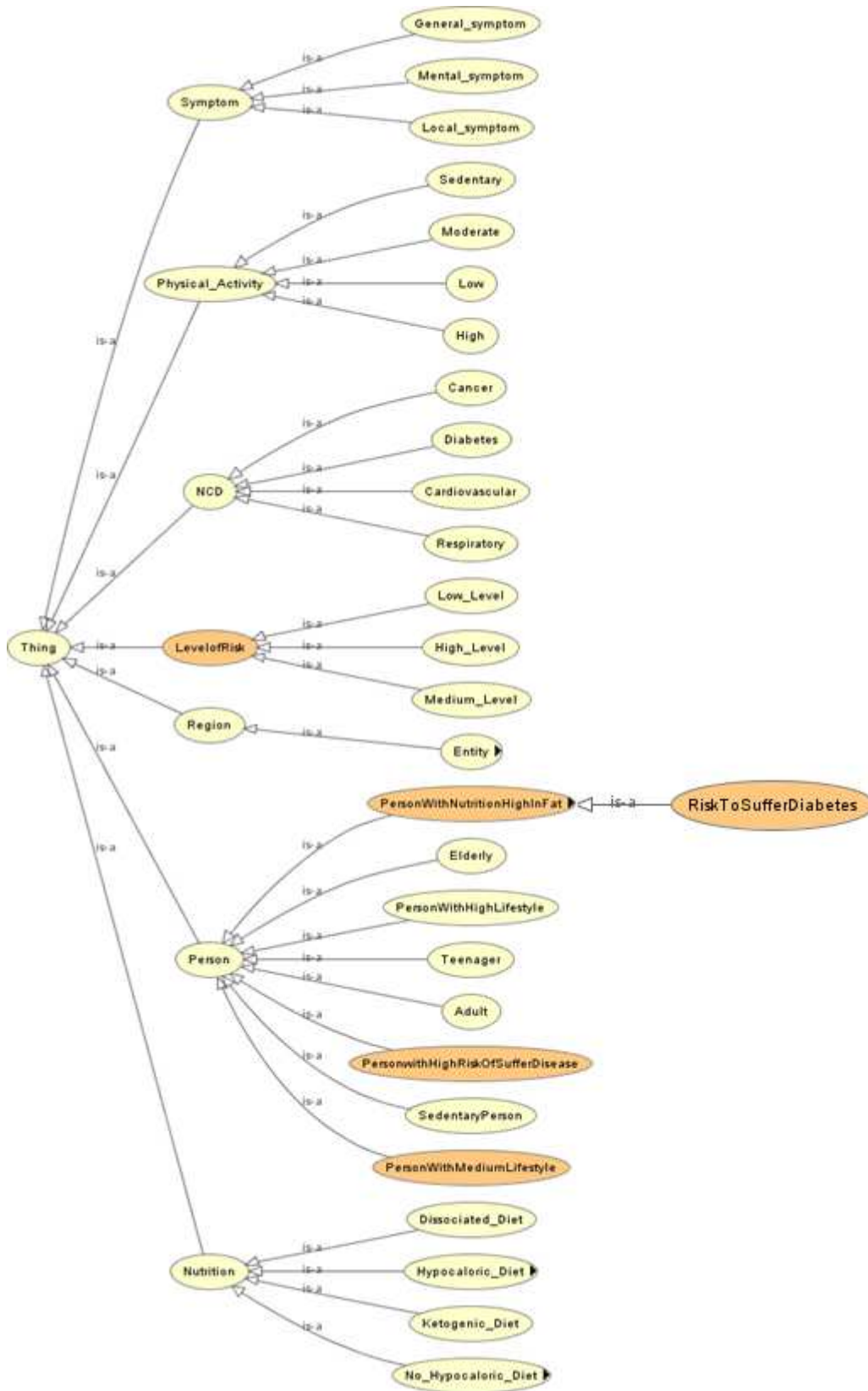


Fig. 4.20 Inferred Model del sistema

Como último paso se crearon Particiones de valor, i.e. clases que fueron utilizadas principalmente para mostrar los niveles de factores de riesgo. Para ello, se creó la clase *LevelOfRisk*, para indicar el nivel de riesgo de sufrir una enfermedad, asignado a la clase *PersonWithHighRiskOfSufferDisease*. Y que agregado a una regla SWRL, será acorde a los patrones encontrados, mostrado en la Figura 4.21.

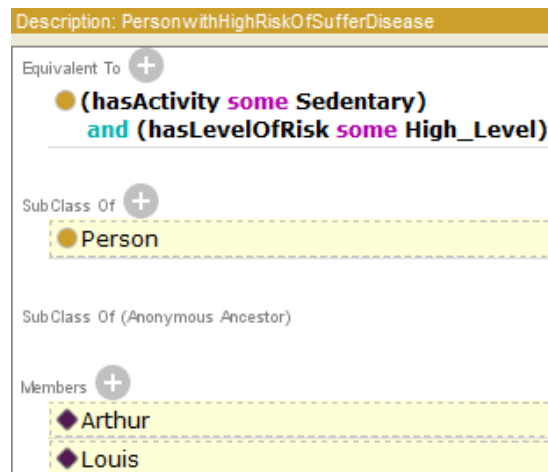


Fig. 4.21 Resultado de aplicar las particiones de valor aplicadas a la clase *PersonWithHighRiskOfSufferDisease*

Finalmente, se presentan las conclusiones finales derivadas del desarrollo observado en el presente capítulo.

Capítulo 5

Conclusiones

En este capítulo se presentan las conclusiones y algunos comentarios acerca del trabajo futuro y subsecuentes consideraciones en el área de desarrollo de esta tesis, que podrán elevar la calidad del sistema así como también su complejidad.

Durante el desarrollo de este proyecto los diversos retos a los que me he enfrentado comprenden desde el planteamiento del sistema hasta el análisis de los patrones obtenidos.

5.1 Conclusiones finales

Los patrones de estilo de vida encontrados por medio de la aplicación de las reglas SWRL aquí presentadas, han sido muy útiles en el descubrimiento de las relaciones que se establecen entre múltiples dimensiones de datos de diversos dominios, incluidos entre ellos los siguientes:

- Personas,
- ENT,
- Síntomas,
- Actividad Física
- Nutrición y
- Regiones geográficas

Producto de lo anterior se cuenta con mayor información acerca de las diferentes formas en las cuales se relacionan las personas, Enfermedades No Transmisibles, síntomas, actividad física, nutrición y la región donde viven.

El sistema de ontologías propuesto, resulta eficiente para tomar decisiones al analizar múltiples datos provenientes de diversos dominios, datos que a su vez, se encuentran interrelacionados entre sí, y contenidos en distintas ontologías. Igualmente, muestra que la lógica descriptiva utilizada en el razonamiento OWL DL en la herramienta Protégé, es

eficaz para responder las preguntas de competencia que se formulen y que tengan que ver con el análisis realizado en el enfoque del problema en cuestión. Es importante señalar que este sistema implementa una herramienta de soporte computacional para la toma de decisiones, en particular una base de conocimiento.

De este trabajo de tesis se han derivado dos artículos de investigación; uno de ellos fue presentado en la conferencia IWBBIO 2015 [1], celebrada en la ciudad de Granada, España, y el segundo fue presentado en la conferencia DEXA 2015 celebrada en Valencia, España [2].

Para diseñar las reglas se utilizó parte de la definición de los estilos de vida y recomendaciones de la Biblioteca Nacional de Medicina de Estados Unidos³⁵ y los Institutos Nacionales de Salud de Estados Unidos³⁶. Por otra parte los valores para una dieta sana y otras recomendaciones acerca de la obesidad y sobrepeso fueron los que especifica la OMS en [20] y [21]. Fueron añadidas también, otras indicaciones suministradas por parte de especialistas en nutrición del Instituto Nacional de Salud Pública³⁷.

Todo lo anteriormente expuesto lleva a concluir que la metodología presentada para crear un sistema de ontologías por medio de la importación de ontologías individuales, es también una buena solución al problema de la integración de ontologías.

5.2 Trabajo futuro

Finalmente, se presentan una serie de comentarios acerca del trabajo futuro en el área de desarrollo de la presente tesis, que podrán elevar la calidad del sistema así como su complejidad.

³⁵ <https://www.nlm.nih.gov/medlineplus/spanish/healthyliving.html>

³⁶ <http://www.nih.gov/>

³⁷ <http://www.insp.mx/>

Entre las consideraciones de trabajo futuro se han definido las siguientes:

- En cuanto al sistema se prevé desarrollar una aplicación Web que permita incluir consultas en *SPARQL*, para dar una mayor robustez al sistema. Esta aplicación podría ser utilizada por la comunidad universitaria y científica la región, del país y hasta internacionalmente.
- Migrar a un entorno de servidor, donde el proceso de carga de las ontologías sea mucho más rápido y se evite la restricción de rendimiento, lo cual permitirá que el razonador pueda inferir un mayor número de individuos.
- Especializar las ontologías de *Nutrition* y *Symptom* para generar patrones de estilo de vida más adecuados.
- Como parte del proyecto de posgrado, me encuentro en el proceso de investigación del estado del arte respecto al poblado automático de las ontologías.

Bibliografía

- [1]. Somodevilla, M.J., Pérez de Celis, M.C., Pineda, I.H., Colmenares L.E., Mena, I. System Development Ontology to Discover Lifestyle Patterns Associated with NCD. In: Ortuño F., and Rojas, I. (eds). IWBBIO 2015, LNCS, Part II. vol. 9044, pp. 47--56. Springer International Publishing Switzerland (2015)
- [2]. Somodevilla, M.J., Mena, I., Pineda I.H. and Pérez de Celis, M.C. Deducting Lifestyle Patterns by Ontologies' SWRL Rules. In: Spies, M., Wagner, R.W. and Tjoa, A.M. (eds), 26th International Conference on Databases and Expert Systems 2015, IEEE CPS. (2015)
- [3]. Ontology Integration and Merging, Recuperado de: http://semanticweb.org/wiki/Ontology_Integration_and_Merging (2015)
- [4]. Choi, N., Song, I., Han, H. A survey on Ontological Mapping. SIGMOD Record, ACM. vol. 35. (3), pp. 34--41. (2006)
- [5]. Barnickel, N., Weinand, R. and Fluegge, M. Semantic System Integration—Incorporating Rulebased Semantic Bridges into BPEL Processes, Proceedings of the 6th International Workshop on Evaluation of Ontology-based Tools and the Semantic Web Service Challenge, Tenerife, España, (2008)
- [6]. Fernández-López, M., Gómez-Pérez, A., Juristo, M. METHONTOLOGY: From ontological art towards ontological engineering. Proceedings of the Ontological Engineering AAAI-97, Spring Symposium Series, pp. 33--40. (1997)
- [7]. Staab, S., Schnurr, H.P., Studer, R. and Sure, Y. Knowledge Processes and Ontologies. IEEE Intelligent Systems, 16 (1), pp. 26--34. (2001)
- [8]. Semantic Sea Ice Interoperability Initiative. Recuperado de <http://nsidc.org/ssiii/ontology-definition.html>, (2015)
- [9]. Noy, N.F. and McGuinness, D.L., Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.
- [10]. Musen, M.A. Dimensions of knowledge sharing and reuse. Computers and Biomedical Research 25. pp. 435--467. (1992).

- [11]. Horridge, M., Brandt, S. "A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3". The University Of Manchester (2011)
- [12]. Bravo, M., Rodriguez, J., Pascual, J. SDWS - Semantic Description of WebServices. International Journal of Web Services Research, Vol. 11 (2) (2014).
- [13]. Bravo, M. Similarity Measures for Web Service Composition Models. International Journal on Web Service Computing. Marzo 2014, Vol. 5 (1) (2014)
- [14]. Barry, Conboy, Lang, Wojtkowski, Wojtkowski: "Information Systems Development". Springer Editorial. ISBN 978-0-387-68772-8. (2009)
- [15]. Gruber, T.R. "A translation approach to portable ontology specifications." Knowledge acquisition, pp. 199--220. (1993)
- [16]. Zhang, Yi, Vasconcelos, W. and Sleeman, D.: "Ontosearch: An ontology search engine." Research and Development in Intelligent Systems XXI. Springer London, pp. 58--69. (2005)
- [17]. Pan, Jeff Z., Edward Thomas, and Sleeman, D. "Ontosearch2: Searching and querying Web ontologies." In: Baptista Nunes,M, Isaías, P. and Martínez, I.J. (eds.). Proceedings of IADIS International Conference WWW/Internet 2006. IADIS. pp. 211--218. (2006)
- [18]. d'Aquin, M., et al.: "WATSON: a gateway for the semantic Web". Poster session of the European Semantic Web Conference, ESWC 2007. (2007)
- [19]. McGuinness, D. The Semantic Web (Current State and Directions). NSF/NCAR October 30 (2002)
- [20]. OMS, Obesidad y sobrepeso. Recuperado de: <http://www.who.int/mediacentre/factsheets/fs311/es/>
- [21]. OMS, Dieta y actividad física. Recuperado de: <http://www.who.int/dietphysicalactivity/es/>